

Title: DevOps Drives Innovation

Course Title: Management of Technology Innovation Course Number: ETM 549 Instructor: Dr. Charles Weber Term: Spring Year: 2017 Author(s): Nirupama Mantha

Report No.: Type: Student Project Note:

ETM OFFICE USE ONLY

Table of Contents

Abstract	3
Introduction	3
Research questions	4
Research methods	1 4 4
Thesis statement	5
Literature Review	5 5 7
Adoption of Agile and DevOps12	2
Relation between Agile and DevOps1	3
Increase in adoption	5
	,
Case Studies	7 7 7
Case: HISCOX	7
Expert interview results19	9
Conclusion	4
Implications and suggestions for further research	5
References	5

Abstract

DevOps is a set of cultural and technical practices that help organizations improve how they work so they can deliver better software faster. Companies who want to move to a DevOps model need to have an Agile mindset and need a cultural change as to how the organization functions. In the current market, there is a need for software to be delivered more frequently, reliably and at higher quality standards. Companies move to Agile software development methodologies to help support the needs, to move to crossfunctional teams that are self-organizing and can deliver software on a much more frequent cadence. This paper talks about the challenges for adopting DevOps and the changes companies need to make, also discussing how the innovation in Agile software development lead to DevOps and how that is changing the organizational culture.

Introduction

Software companies are globally distributed, so are the teams that work on features. Trying to affect process, people, technology and cultural changes across the entire portfolio, in a globally distributed team and with a lot of associated technical debt, is a challenge, but it's the reality for many companies.

This paper will research the challenges faced by organizations in moving to Agile and how they can adopt tools and process changes to move to a DevOps model.

Research questions

This report is trying to answer the following questions:

- How does DevOps relate to Agile?
- What are the challenges in adopting Agile and DevOps?

Research methods

Literature Review

The research for the paper will be done through Literature review of existing DevOps articles, white papers as well as existing case studies of companies that have gone through the shift. The "State of DevOps Report" published yearly will be used to analyze data and trends that are seen by practitioners and companies.

Expert Interviews

Interviews will be conducted with three experts who are in management positions in Software Organizations that have implemented DevOps and have worked with teams that followed Agile Software Development methods. Below are the questions that will be used for the interviews

Interview Questions:

- 1. Can you move to DevOps and follow a Waterfall approach to Software Development?
- 2. Can you move to DevOps without being Agile?
- 3. What were your challenges to move to Agile?
- 4. What were your challenges to move to DevOps?

- 5. What benefits do you see with Agile and DevOps?
- 6. What could you have done differently in implementing Agile or DevOps? (or) What would you recommend for a team/company adopting Agile or DevOps?

Thesis statement

Agile Software development lead to DevOps.

Literature Review

Innovation in Agile Software development



Figure1: Now every company is a software company – Forbes; Source: [8]

Post dot com boom, business environment continues to change at a dramatically increasing pace, to thrive in this turbulent environment, we must confront the business need for relentless innovation and forge the future work- force culture [17]. Agile software development methodologies like Extreme Programming, Scrum, Kanban etc.

embrace change unlike traditional software development methods and are based on the Lean and Agile principles, were individuals and interactions, working software, customer collaboration and responding to change are valued [18].

Traditional approaches to software development tried to reduce cost by eliminating change; While today, eliminating change early means ignoring customer needs, which means business failure [17].

Agile software development addresses two pressures that characterize today's business and technology world: the need for dynamic, innovative approaches and the desire to build workplaces that aren't described in Dilbert cartoons [17]

Agile has transformed software development, the term "agile innovation" means teams using agile get things done faster, make customers happy and enjoy their work. Jeff Sutherland et.al in their article "Secret History of Agile innovation" [19] mention that the starting point for agile might be in the 1930's when Walter Shewhart of Bell labs began applying the Plan-Do-Study-Act (PDSA) cycles to the improvement of products and processes. This was passed onto Edwards Deming who was hired by Toyota, where the famous Toyota production system was developed – which is the primary source of "Lean" thinking [19].

In 1986 Takeuchi and Nonaka studies manufacturers that were releasing successful innovations faster than competitors, where the companies were using that they called the "rugby" approach, where the team goes the whole distance as a unit, passing the ball back and forth, instead of the conventional "relay race" method where functional specialists hand off completed phase to the next functional stage [19].

6

Jeff Sutherland borrowed Takeuchi's and Nonaka's rugby approach and applied to software development adding in specific operational practices, and created a new way of developing software "Scrum". He collaborated with Ken Schwabber to codify the approach and in 1995 presented Scrum to the public [19]

In 2001, 17 developers met in Snowbird, Utah (including Sutherland). The group included advocates of Extreme Programming(XP), Feature-driven development(FDD) etc. The group named the movement "Agile" and created the "Agile Manifesto for software development" which had four key values everyone agreed on. They also developed 12 operating principles called "Principles behind the Agile Manifesto" [19].

The agile movement spread rapidly, formal lean and Kanban software development systems emerged during the 2000s. Once Agile methodologies have been established, there have been frameworks developed to scale agile like SAFe, DAD, Nexus and LeSS to name a few [12].

DevOps overview

There has been a lot of interest in DevOps lately but it's not always obvious what DevOps is and what the benefits are for those who are new to these ideas [1]. Nigel Kirsten, Chief Technical Strategist at Puppet feels that not having s strictly defined definition of what it means to do DevOps has allowed the DevOps movement to grow and respond to the changing landscape of infrastructure options [1].

DevOps is a new term emerging from applying newer Agile and Lean approaches to operations. It was first called "Agile Operations". It is understanding of the value of

collaboration between development and operations staff through all stages of the development lifecycle [3]. A Practical definition of DevOps would be "The practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support" [3]. It's operations staff applying the same principles as software development for systems work.



Figure 2: Terms referring to DevOps and their prevalence in literature review; Source: [21] Reducing the silos created between the "Dev" and "Ops" sides has been the core driver behind DevOps. DevOps can be interpreted as an outgrowth of Agile Software Development – which prescribes close collaboration of customers, Product management, Developers and QA. DevOps is simply extending Agile principles beyond the boundaries of "the code" to the entire delivered service [3]. DevOps practices can be crystalized into five dimensions that characterize

DevOps. Table below summarizes some of the DevOps practices in each of the

dimensions with patterns and examples [21].

DevOps Dimension	Patterns of DevOps prac- tices	Examples of practices found in ML
Collaboration	Rethinking and reorientation of roles and teams in develop- ment and operations activities	 Increasing the scope of responsibilities Developers paying closer attention to deployment scripts, configuration files, load and performance testing and other activities usually associated with operations groups Developers learning from operations about resilience, monitoring and diagnosing distributed systems Developers learning from operations about resilience, monitoring and diagnosing distributed systems Developers learning from operations about resilience, monitoring and diagnosing distributed systems Developers learning from operations work, e.g., Docker to remove the need for specifying environments specifications Development pipeline to do less operations work, e.g., Docker to remove the need for specifying environments specifications Development totates roles with operations teams, operations attend developer stand-ups and showcases Operations involved earlier in development to understand what project environments are required to support the application. Also, regular meetings, e.g. weekly to discuss cross-team priorities In circumstances in which production incidents occur, development and operations come together to troubleshoot and resolve problems as one team Setting-up shared (virtual and physical) workspace
Automation	Infrastructure and deployment process automation	 Infrastructure-as-code Automate and maintain infrastructure configurations and files using tools such as Chef, Puppet Developers are shielded from infrastructure issues and are able to create virtual development, test and production environments as well as deploy application using tools like Vagrant and Docker Scripts used to handle infrastructure are versioned, testable and repeatable Immutable infrastructure, i.e. artifacts, in their production environments are not updated, rather the infrastructure is always replaced Deployment process automation Production-like environments are used by development teams for development and testing Developers self-service environments and deployments Consistent, reliable and repeatable deployment mechanism across different environments Configuration changes across environments are automated
Culture	Empathy, support and good working environment between development and operations	 Both developers and operations wear pagers as responsible persons to handle incidents Integrating development into blameless production post-mortems Making communication between development and operations non-adversarial and less formal Mutual respect, support and willingness to work together and share responsibilities
Monitoring	Instrumenting application and aggregating monitored data into insights	 Developers and operations are both involved in determining and implementing monitoring parameters of a system Set-up monitoring on the production environment for development teams visible through radiators Development team, including QA, use small subset of high-priority test cases to be executed in production to actively monitor the environment Effective instrumentation of software by development in collaboration with operations to give information about its health and performance. Also, developers are able to quickly recover code failures in production using aids, such as feature flags
Measurement	Useful Metrics	 Both operations team and development team are incentivized and rewarded by the same metrics Both development and operations focus on business value as the essential unit of measurement Progress in development is measured in terms of a working system in production environment Developers use production feedback to drive decisions, improvements, and changes to the system

Table1: DevOps dimension, patterns and practices; Source: [21]

The word DevOps was introduced in 2009 by Patrick Debois, who organized the first DevOpsDays conference. The conference was to bring together practitioners to discuss and bridge the gap between Development and Operations. Since 2009 DevOps conferences have been held across the globe, driven by community with a goal to improve, people, process and tools [22].

Common myths about DevOps are that DevOps is all about tooling or that its only for startups. DevOps is a cultural shift, collaboration across teams and breaking silos that have been established so value can be delivered faster, more frequently and with ease. The DevOps Cycle is as follows



Figure 3: DevOps Cycle; Source: [23]

Some of the DevOps principles to support the DevOps cycle are

- Version Control
- Test Automation
- Continuous Integration
- Infrastructure as code
- Monitoring
- Continuous Delivery
- Orchestration Pipelines

There are many tools that are available to support DevOps needs, below is an example of tools used by Pokémon Go



Figure 4: DevOps Tools - Pokémon Go; Source: [23]

Adoption of Agile and DevOps

Although DevOps emphasizes people (and culture) over tools and processes, implementation utilizes technology. As a result, Gartner, Inc. expects strong growth opportunities for DevOps toolsets, with the total for DevOps tools reaching \$2.3 billion in 2015, up 21.1 percent from \$1.9 billion in 2014. By 2016, DevOps will evolve from a niche strategy employed by large cloud providers to a mainstream strategy employed by 25 percent of Global 2000 organizations [24]

The 2017 State of DevOps report shows the increase in the number of DevOps teams by department



Figure 5: DevOps teams by Department; Source:[25]

Over the past years, there has been a shift in awareness at senior levels of management that traditional software development is not meeting the needs of customers and end users. A survey conducted shows the measures companies are taking to deliver better software and the results show that 56% are moving to Agile and 23% to a DevOps structure [6].



Figure 6: Measures taken to deliver better software faster; Source: [6]

Relation between Agile and DevOps

Every business is using code to adapt to changes in the marketplace [6]. The term Agile refers to creating software using the iterative and incremental practices enabling companies to respond to rapidly changing requirements. DevOps extends Agile concepts to other IT departments such as infrastructure, networking, data center operations, and support [7].

A Multivocal literature (ML) review study on the Relationship between DevOps to Agile, Lean and Continuous Deployment revealed that DevOps phenomenon originated from continuous deployment as an evolution of agile software development [26]. The study also revealed that a successful adoption of DevOps requires agile software development [26].



Figure 7: Publication years and number of included ML and scientific articles; Source: [26] The observations were made as part of the research on the relationship between DevOps and Agile [26]

- *Origin and Background:* DevOps is described as an extension of agile software development, with the aim of applying the Agile values, ideas and practices to operations.
- Adoption: DevOps has been related to successful agile software development.
 Organizations applying agile methods will find DevOps familiar. It states that agile in itself does not lead to DevOps adoption, unless Operations are involved and collaborated on, it's important to have those communications going.

- *Implementation:* DevOps is related to agile software development, in implementing continuous integration and using agile practices to guide interactions between development and operations.
- Goals and Values: DevOps and agile software development are driven by similar goals and values, DevOps requires development teams to work with the operations team and understand their needs similar to agile were development teams work with customers to understand the real needs and priorities.

Increase in adoption

Today organization depends on software. Retail, logistics, government, scientific research, tech, education, financial services — every sector needs some sort of software to meet customer and user needs [1]. The expectations people have of software changed dramatically, they expect reliable and convenient services that are regularly improved [1].

The growing popularity of agile methodologies resulted in more releases, putting even more pressure on operations teams, and making it more urgent to improve how they managed infrastructure. [1]

In agile software development, the relationship between customers and suppliers remains a challenge [20], there is an increase in continuous value delivery. The figure below shows the increase in interest over Scrum and DevOps over the years.



Figure 8: Relative interest over time on themes "Scrum", "extreme programming", "DevOps" and "continuous integration" based on searches in Google Trends, showing results for category "computers and electronics/programming"; Source: [20]

Agile relies on inspect and adapt methods and using empirical data. The increase in Agile adoption has led to the idea of Continuous experimentation where potentially valuable features are delivered to different user groups and data is collected to understand the value of the delivered functionality [20]. In this approach, different versions of the software might be delivered to different user groups, to understand product usage and experienced customer value [20]. Specific tools will be needed to support the execution and analysis for this method.

Challenges in adoption of Agile and DevOps

Below are some of the challenges in moving to Agile and DevOps

• The hardest part of DevOps is the people aspect. DevOps is a cultural idea, so it's very people centric [28]

- Change in Organizations Culture
 - Changing to a culture of continuous learning and experimenting
 - Changing the mindset of people
- Process and tools needed to Move to Agile and DevOps

Results

Case Studies

Case: Smartbit

Smartbit founded in 2003 is the creator of Smart Schools, an Education software company, an online digital education platform used by 90 percent of Flemish-language schools in Belgium. Their challenge was managing a growing, complex cloud infrastructure with a small IT team [27]. As a result of moving to DevOps, employing Infrastructure as Code and implementing tools to support that, Smartbit is able to provide uninterrupted services to the majority of Belgian schools and scale efficiently as Smartschool is adopted in the rest of the country's education system.

Case: HISCOX

Hiscox is a publicly traded property and casualty insurance company based in the UK. Their challenge was to the need to increase the pace at which they released change and new products to market, that was highly competitive. The company decided to introduce DevOps practices and use automation tools to manage its infrastructure (Puppet Enterprise) [28]. Hiscox started their DevOps journey by educating their executives on the problems they were facing and the challenges with the current way of working. Once the executives were on board the created product focused teams were the members were cross functional team members including business analysts, developers, testers and others needed for the teams to be autonomous. They implemented the change in product team on a particular business unit. This helped them think about not only deploying the product but how they will maintain it post deployment [28].

Some challenges they faced starting their DevOps journey [28]

- Sheer scope and size of challenge with the change
- Effect people and process changes
- Change in technology and cultural changes
- Trying to move to Agile, managing deployments, testing, spinning new environments, instituting version control strategies all at once
- Test bottleneck

Hiscox implemented TDD (test –driven-development) and BDD (business-drivendevelopment) to address some of their test bottleneck issues. They invested time upfront in changing mindsets and getting buy-in.

Here are the results seen by Hiscox after implementing DevOps [28]

- Reduced cost per release on a major application by 97%.
- Release cycle improved from 10 weeks per deployment to 50 deployments per week.

- Reduced time per release by 89%
- Reduced staff required to release by 75%

Expert interview results

Experts who are in management positions in Engineering organizations and had implemented Agile and DevOps in their organizations were chosen. Expert 1 is a Director of Engineering at a Sportswear company and Expert 2 is a Software Development Manager at an e-commerce company. The experts were asked the same set of questions and the results from the interviews are presented below.

1. Can you move to DevOps and follow a Waterfall approach to Software Development?

Expert 1: Yes. DevOps for me is Developer (engineer) owning the code from the inception, testing, deployment to production and support. The entire lifecycle is owned by the Developer. One can do DevOps with waterfall too. But then the "Development" process is slow non-iterative but the "ops" process is owned by Engineer. What will happen in this case is the Developer is bogged down with defects and work arounds as they don't have flexibility to fix things until the next release. As waterfall typically comes with a long release cycles, engineers will then have to work more time on the "Ops" part rather than the "Dev" part.

Expert 2: Yes

2. Can you move to DevOps without being Agile?

Expert 1: Yes, but please read answer to my question above.

Expert 2: Yes, because you can have the ability to develop software more frequently.

3. What were your challenges to move to Agile?

Expert 1: It's culture and organization. Bringing the product and engineering together and be able to work iteratively is a big challenge. In the beginning, Product will be of the mindset that the only thing that comes beyond MVP (minimum viable product) is support no other features. Stakeholders had a difficult time funding "Products" on a continuous basis and the fact that funding needs to be in the increments of a team/squad and not one person at a time was also difficult within the organization.

For the Dev team, trying to create stories that fit in a two week sprint and that has clear acceptance criteria and can be demonstrated was also a challenge. Without automation creating products iteratively caused a lot of manual testing and increased cost of development.

Expert 2: It's all about people, more than the systems - changing people's attitude and mindset, to take on iterative development and deployment. The team to deliver in small iterations, if the upper management is still thinking in waterfall is a challenge. E.g. DBAs think of ER diagrams as a whole as opposed to iterative designs. One needs to make sure we get buy-in from all cross functional teams Reconciling the mismatch from dev teams and upper management there is a challenge.

4. What were your challenges to move to DevOps?

Expert 1:

- Culture: How can every engineer have access to production environment? Creating an auditable and safe practice was important.
- What do we do with current support Organization?
- Developers being afraid about the possibility of being on-call 24X7, causing a lot of apprehension
- Continuous delivery: When a problem happens in production, if an engineer cannot fix and deploy the fix for this as soon as possible (based on other priorities), DevOps will not be successful. Without CD the amount of time the Engineer has to spend on monitoring, incident management and problem management will eat into development time. Slowly more time is spent on "ops".

Expert 2: operations team is not involved in DevOps process, team is still struggling with DevOps

5. What benefits do you see with Agile and DevOps?

Expert 1:

- Agile lot of "why" on the internet. One of the most important thing is the team always works on the highest priority items at that time. Product owner gets to see the progress and change priorities, iterative development.
- DevOps Engineers own the solutions. Because they wrote the solution they can support it the best. Because they support, they will write good code and fix it as soon as defects are found (Eat your own dog food). Things are more stable and built better here.

Expert 2:

- Agile Is the only way to develop software right now. You don't want to be disconnected with the customers. Deliver working software, identify what's out of sync and adjust, instead of deviating for very long
- Cost of developing software is high if team is inflexible in adapting to chaining requirements. Agile is the only way to adjust to chaining requirements and addressing customer needs.
- If DevOps becomes a bottleneck then the efficiency gained with Agile will be lost, they both need to go hand in hand.

- 6. What could you have done differently in implementing Agile or DevOps? (or) What would you recommend for a team/company adopting Agile or DevOps? Expert1:
 - Automation, Automation Every part of the build pipeline needs to be automated. Things should include deploy, validations, Canary analysis, roll forward, roll back etc.
 - Testing: Unit testing, integration testing, Synthetic testing in production, everything needs to exist.
 - Having some experienced engineers who have seen real DevOps in production will be great.
 - The team needs to recognize that support activities are going to come when it's un-expected. How they organize to account for that and continue to adjust is very important.
 - Don't do DevOps without a good Continuous Delivery process, or the team will be stuck with release and deploy management processes.

Expert2:

Had an agile coach who educated the team on agile practices.
 Management was also on the same page, which helps. It gets difficult to get everyone on the same page in a fast-growing environment, in such cases teams tend to be more autonomous and choose their own tools and practices.

- Agile as a process is not a problem constant change and adopt, with retrospectives, things that are in the team's control are not an issue, anything beyond the team's control-just have to live with it
- DevOps integration tests were not built and learnt it the hard way
- How mature the regression test suites are will lead to success
- Roll back is important if the changes are not reversible, then it's a point of no return - having a good rollback strategy - that's automated or one click will be good
- One box testing features need to go into Production and you are not sure, you can deploy it to one box and check behavior, then let this be deployed to the rest of the machines.

Conclusion

The literature and articles reviewed for this paper show that organizations need to move to agile software development methodologies to keep up with the changing markets and customer demands. The growing popularity of agile methodologies resulted in more releases, putting even more pressure on operations teams, and making it more urgent to improve how they managed infrastructure [1].

"DevOps is a culture, not a role," DevOps transformation lead at Sainsbury's Mike Dilworth said "The whole company needs to be doing DevOps for it to work." [6]. DevOps is about much more than simply applying agile principles to infrastructure management. One important reason for this is the strong cultural and organizational dimension to resolving the conflict between incentives for development and operations

24

teams. DevOps places a strong focus on cross-functional teams working together across the divide [1]. As the line between app dev and infrastructure deployment becomes more blurred, your ability to deliver software quickly, and with fewer errors, improves dramatically [1].

DevOps is an evolution or extension of agile software development, which was based on lean principles. The findings from the expert interviews revealed that to gain the full benefits of DevOps organizations need to embrace Agile, where they are iterating frequently and have tighter feedback loops to respond to changing requirements. The interviews also revealed that for both Agile and DevOps, the teams need to have the right set of tools to be successful. It's also important to walk with Agile before running and it's about embracing continuous learning [2].

The research questions that were asked were

- How does DevOps relate to Agile?
- What are the challenges in adopting Agile and DevOps?

Agile and DevOps go hand in hand where DevOps picks up after agile. The challenges that were identified are mainly Changing culture, mindset and addressing changes needed to the current organizational structures along with tooling needs.

Implications and suggestions for further research

The research for this paper was mainly based on literature review of scholarly articles, technology blog posts and case studies. The interviews from the experts provided great insights, but also highlighted that how organizations perceive DevOps is different,

specially in large enterprises. It would be good to interview a larger sample and compare results. DevOps introduced new roles in organizations like DevOps engineer, DevOps teams etc. There seems to be some literature around Site Reliability Engineer(SRE) role and how that might integrate or change the DevOps roles currently identified. It would be beneficial to do a comparison and understand the differences. It will also be beneficial to come up with a list of capabilities organizations will need to have to be successful with Agile or DevOps implementations, I was not able to do that currently due to time limitations.

References

[1] Kirsten, Nigel. "What Is DevOps?" *Puppet.com*. Puppet, 30 Jan. 2017. Web. 1 May 2017.

[2] "Gartner Highlights 10 Things CIOs Need to Know About Agile Development." *Gartner*. Gartner, 1 June 2015. Web. 06 May 2017.

[3] Mueller, Ernest. "What Is DevOps?" *The Agile Admin*. The Agile Admin, 7 Dec. 2016. Web. 3 May 2017.

[4] Kim, Gene. "The Three Ways: The Principles Underpinning DevOps." *IT Revolution*. IT Revolution, 25 Apr. 2017. Web. 06 May 2017.

[5] Kim, Gene. "Starting DevOps Value Stream." *IT Revolution*. IT Revolution, 25 Apr. 2017. Web. 06 May 2017.

[6]_"DevOps and the need for speed " *Puppet.com*. Puppet, 30 Jan. 2016. Web. 1 May 2017.

[7] Mayner, Stephen W. *Transformational leadership and organizational change during agile and devops initiatives*. Diss. Capella University, 2017.

[8] Forbes, Techonomy. "Now Every Company Is A Software Company." *Forbes*. Forbes Magazine, 09 Jan. 2012. Web. 06 May 2017.

[10] Riggins, Jennifer. "How Shell Oil Is Taking DevOps and Agile to the Cutting Edge." *The New Stack.* Shell, 1 Jan. 2017. Web. 15 Apr. 2017.

[11] N, Ravinshanker. "DevOps and Agile." *Scrum Alliance*. N.p., 18 Apr. 2014. Web. 06 June 2017.

[12] Marschall, Matthias. "How Are Lean, Agile, and Devops Related to Each Other?" *AGILE WEB DEVELOPMENT & OPERATIONS*. N.p., 12 July 2012. Web. 4 May 2017.

[13] Perrow, Mike. "How to Scale Agile and DevOps Together." *Tech Beacon*. N.p., 2017. Web. 17 Apr. 2017.

[14] Overby, Stephanie. "7 Key DevOps Roles Organizations Need for a Successful Transition." *TechBeacon*. TechBeacon, 07 Apr. 2017. Web. 06 May 2017.

[15] Simon, Phil. "DevOps: 4 Benefits of Using Agile Methods." *TechBeacon*. TechBeacon, 07 Apr. 2017. Web. 06 Apr. 2017.

[16] Hill, Patrick. "Love DevOps? Wait until You Meet SRE." *Atlassian*. N.p., 2017. Web. 06 May 2017.

[17] Kirsten, Nigel. "What Is DevOps?" *Puppet.com*. Puppet, 30 Jan. 2017. Web. 1 May 2017.

[18] Highsmith, Jim, and Alistair Cockburn. "Agile software development: The business of innovation." *Computer* 34.9 (2001): 120-127.

[19] "Manifesto for Agile Software Development." *Manifesto for Agile Software Development*. N.p., 2001. Web. 06 June 2017.

[20] Takeuchi, Darrell K. RigbyJeff SutherlandHirotaka. "The Secret History of Agile Innovation." *Harvard Business Review*. N.p., 20 Apr. 2016. Web. 06 May 2017.

[21] Dingsøyr, Torgeir, and Casper Lassenius. "Emerging themes in agile software development: Introduction to the special section on continuous value delivery." *Information and Software Technology* 77 (2016): 56-60.

[22] Lwakatare, Lucy Ellen, Pasi Kuvaja, and Markku Oivo. "An Exploratory Study of DevOps Extending the Dimensions of DevOps with Practices." *ICSEA 2016* (2016): 104.

[23] "DevOpsDays." DevOpsDays. N.p., n.d. Web. 06 June 2017.

[24] Anand, Vishnu. "Pokemon Go – a Perfect Use-case of DevOps Principles." *Edureka!* N.p., 19 Aug. 2016. Web. 1 June 2017.

[25] "Gartner Says By 2016, DevOps Will Evolve From a Niche to a MainstreamStrategy Employed by 25 Percent of Global 2000 Organizations." *Gartner*. N.p., 5 Mar. 2015. Web. 6 May 2017.

[26] Kirsten, Nigel, Alanna Brown, Jene Kim, Jez Humble, and Nicole Forsgren, Dr. "2017 State of DevOps Report." Puppet, 6 June 2017. Web. 6 June 2017.

[27] Lwakatare, Lucy Ellen, Pasi Kuvaja, and Markku Oivo. "Relationship of DevOps to Agile, Lean and Continuous Deployment: A Multivocal Literature Review Study." *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17.* Springer International Publishing, 2016.

[28] "Case Study: Smartbit." Puppet, n.d. Web. 6 June 2017. https://puppet.com/resources/case-study/smartbit.

[29] Fletcher, Jonathan. "Case Study: Hiscox." Puppet, n.d. Web. 6 Apr. 2017.

[30] Kirsten, Nigel, Alanna Brown, Jene Kim, Jez Humble, and Nicole Forsgren, Dr. "2017 State of DevOps Report." Puppet, 6 June 2016. Web. 1 June 2017.