Defining Leadership and its Challenges while Transitioning to DevOps

Krikor Maroukian
Microsoft
Modern Service Management
Athens, Greece
e-mail: krmaro@microsoft.com

Stephen R. Gulliver
Henley Business School
University of Reading
Reading, United Kingdom
e-mail: s.r.gulliver@henley.ac.uk

***Abstract*— DevOps practices and principles adoption is no longer restricted to technology-specific skills. Many studies indicate that successful DevOps adoption is part of continuous corporate transformation at all levels, and that includes cultural and behavioural patterns, process-driven perspective and toolchain readiness for usage. Our research method involves the analysis and evaluation of 30 interviews with participants from the private and public sectors in the EMEA region. Analysis and evaluation of a survey completed by 250 participants (73% from Europe) and 76% who have held previous leadership positions is also included. A mixed methods approach was used. Thirty (30) participants from consultancy firms and service provider organisations generated coded themes to expand our understanding of relevant factors. From the 250 survey participants, 81% had 10+ years of professional experience and two-thirds were currently practicing DevOps. The aim of our research was to unveil leadership-specific observations on characteristics and factors that would indicate certain reasoning behind challenges faced by organisations while transitioning to DevOps. Our results show that top leadership factors identified are: communication and collaboration, customer-centric mindset, having a technical background, and being an active listener. The least important factors identified were: gaining a relevant certification, design thinking, previous experience on transformation projects, and talent seeking.**

Keywords-DevOps adoption; resistance factors; leadership characteristics; metrics.

I. INTRODUCTION

The 1990s saw the birth of pre-agile approaches, such as the Rapid Unified Process [1] and XP [2] [3], which eventually led to Agile Software Development, which is characterised mainly by lightweight, flexible, adaptive processes linked to rapidly changing corporate business environments aiming to eliminate waste [44]. The traditional 'waterfall' approach to release and deployment management requires a release cycle of 6-18 months, which shifts focus to maintenance-only. This practically means that operations teams in Information Technology (IT) organisations are focused on purely reactive maintenance activities including bug fixes. There is, however, a lack of development of new feature development, i.e., change of features or functions that would fundamentally change the program architecture [13].

Software has become pervasive in day-to-day human activities, and the world economy is now dependent on software use. This in turn has increased the importance of having software-intensive products and services that are useful, secure and reliable at all times during operational use.

A retrospective view of the last twenty years of software product development practices and principles shows that a decline of Extreme Programming (XP) publications has been succeeded by the gradual increase, i.e., since about 2009, in the popularity of agile and lean practices, such as SCRUM [42] and Kanban [43]. Moreover, two other areas that seem to be gaining popularity are technical debt and code smells, which address software product development and code maintenance suboptimisation - in terms of agile team velocity to deliver sprint artefacts for the minimum-viable product. Furthermore, certain agile practices, e.g., pair-programming since 2003, user stories since 2003, test-driven development since 2007, and code refactoring since 2009, are relatively stable. In addition, DevOps, Continuous Integration, Continuous Deployment, Continuous Delivery are characterised as 'hot research topics', with considerable increases in popularity since 2014 [14].

Leading DevOps practice and principle adoption has become a fundamental element to the success of DevOps teams [4] [39]. A high-performing organisation is characterised by adoption of DevOps practices by multiple teams and departments, high responsiveness to mean-time-to-recover from product system failure, i.e., end-user experience degradation, mean-time-to-market, change failure rate, and embedding security deep into the source code [15]. However, there is still limited research outlining the leadership style, traits, competencies and skillset accompanied with high-performing DevOps-oriented organisations. Speed in the development and delivery of new software features provides the opportunity to respond quickly to customer needs, business opportunities, and get quick feedback about the new software features [16]. Feedback loops facilitate information that is useful to make informed decisions regarding software development efforts; conducted by different stakeholders of the software product development value stream.

The purposes of our study are (RQ1) to provide a better understanding of the leadership characteristics required to enable DevOps practice and principle adoption, (RQ2) to gain insights into the DevOps adoption inhibitors or resistance factors slowing down change, and (RQ3) to examine the associated metrics to these set of competencies and leadership style.

This paper is divided into four further sections. Section II lays out the thoroughly researched account of literature behind DevOps adoption and pertinent leadership research, the challenges faced in the transitional period towards

DevOps practice and principle adoption and which type of technology-agnostic i.e., team-driven, process driven, metrics can be associated to DevOps. Section III describes the research method, design and collection process. Section IV outlines the analysis and evaluation of the interview and survey results, including an examination of research validity. Lastly, Section V concludes the paper including future research considerations.

II. Background and Related Work

A. *DevOps Adoption and Leadership*

The adoption of DevOps practices and principles requires several factors to be taken into account. The most popular model among DevOps adoption is known as Culture-Automation-Lean-Monitoring-Sharing (CALMS) [17], which requires a change of people's mindset, skill and toolsets. This orientation requires gradual and minor changes in an organisation's daily operations. For companies to move from structured to agile structures in software development, there needs to be first an adoption stage of agile practices and a shift to smaller cross-functional teams, and later, when a certain level of maturity is attained, DevOps practices can be adopted, such as automated system integration and continuous integration [18]. When continuous integration is in place, customers express an interest in receiving enhancements and bug fixes more frequently. Therefore, adoption of continuous delivery practices is required. The final step occurs when the organisation not only releases software continuously, but also develops mechanisms to conduct rapid experimentation in order to drive innovation.

The DevOps Institute's Collective Body of Knowledge (CBOK) focuses on three pillars: DevOps, Lean and Leadership [19]. In addition, successful adoption of DevOps requires agile software development [29]. For practitioners in the industry, there is a decline of interest in XP, and a steady increase in SCRUM over time. Between 2006 and 2015, there was an increase in interest concerning continuous integration, however, there was a sharp increase in DevOps adoption in the last few years [20]. This sharp increase has most likely been triggered by DevOps leaders who have acquired the transformational acumen required to contribute to the design, influence, and motivate cultural transformation, which is proven to be a critical success factor in DevOps adoption: making it a multidisciplinary topic that requires application of a mix of skills, practices, and principles [21].

The State of DevOps Report published by Puppet discovered a correlation between transformational leadership and organisational performance [23]. Transformational leadership comprises of four dimensions: idealised influence, inspirational motivation, intellectual stimulation, and individualised consideration and the leader aims to inspire and transform followers by appealing to their ideas and emotions [41]. In addition, the State of DevOps Report conveys that DevOps leaders with a servant leadership mentality inspired better team performance [23]. In essence, the leader is serving rather than being served and, therefore, creates an environment of trust, collaboration and reciprocal service, which ultimately leads higher performance [40]. Servant leadership was developed as a theory of ethical leadership, which is comprised of values, such as integrity, altruism, humility, empathy and healing, personal growth, fairness, justice and empowerment [41]. Our study attempts to identify the characteristics presented by a mixed methods research design approach and how obtained results and outcomes relate to transformational and servant leadership.

B. *DevOps Adoption Challenges*

Following a decade of DevOps, there is no firm consensus amongst software practitioners and scholars as to what the DevOps definition actually includes [4] [30]-[34]. Moreover, DevOps is unclear but also evolving [13]. Literature defines DevOps in numerous ways, although, the majority of descriptions specifies 'DevOps' as a term that is used to emphasise the collaboration between software development and operations. There is, however, a research and industrial need to develop a better understanding of the DevOps scope [20], since DevOps has been described as: a new role within a software organisation [35]; a movement for change in software industry [30]; a set of software development practices [4]; a leagile approach [22] – i.e., the combination of the lean and agile paradigms; and High Velocity IT [5], which ITIL4® defines as involving techniques for valuable investments, fast development, resilient operations, co-created value and assured conformance.

Cultural enablers, used to promote the adoption of DevOps practices, are required, such as focus on decision making, customer focus, engineering practices, learning and development, leadership, team recognition, innovation, guilds and performance feedback [21] [36] [37]. Moreover, to achieve performance gains, while adopting DevOps, the following are shown to be essential [38]:

- Tightened feedback loops between Dev and Ops teams;
- Established practices of automated performance monitoring;
- Measurement of key performance metrics in Continuous Integration, Test and Ops teams;
- Shared tools and performance metrics across teams.

According to the State of DevOps Report [23], published by Puppet and the DevOps Research and Assessment (DORA), there is an increasing inclusion of IT team members into DevOps teams such that:

- Sixteen percent (16%) of the respondents identified themselves as working in DevOps teams in 2014;
- Nineteen percent (19%) of the respondents identified themselves as working in DevOps teams in 2015;
- Twenty-two percent (22%) of the respondents identified themselves as working in DevOps teams in 2016;
- Twenty-seven percent (27%) of the respondents identified themselves as working in DevOps teams in 2017.

Furthermore, there are considerable challenges in DevOps practice adoption in the IT industry. DevOps adoption challenges include, but are not limited to, the insufficient communication, deep-seated company culture, industry constraints and feasibility, heterogeneous environments. Moreover, a Delphi study of 42 Norwegian experts indicated a comprehensive list of problems influencing poor cooperation between software development and operations [24], however, the most serious problems in poor software development – operations cooperation - included the following aspects:

- Operations not being involved in the requirements specifications;
- Poor communication and information flow;
- Unsatisfactory test environment;
- Lack of knowledge transfer;
- Systems being put into production before they are complete;
- Operational routines not being established prior to deployment.

Additionally, the hierarchical approach of organisational structures that welcome static team structures can also become a bottleneck to information flow. Moreover, obstacles to flow can also be characterised as anything that acts as an impediment to cognitive load of a DevOps team topology [45]. Cognitive load refers to the amount of working memory being used at any one moment within a team structure. Flow challenges can be due to disengaged teams, software too big for team structure, confusing organisational design options, team getting pulled into too many directions, painful reorganisation every few years, flow is blocked by certain factors and too many reactive-natured surprises for the team to handle [45].

For modern software companies, speed facilitates fast and repeatable software development and delivery processes [25]. Complexity of performance engineering approaches is a barrier for wide-spread adoption by practitioners. Accordingly, performance engineering approaches must be lightweight and must smoothly integrate with existing tools in the DevOps pipeline [37]. This is evident by the emergence and the growing interest of a continuous deployment paradigm in the software industry. Continuous deployment entails the capability of an organisation to deliver new software features at multiple times and in the shortest time possible. DevOps is an approach that has been reported to enable the continuous deployment paradigm as it embodies a set of useful principles crucial to the development and deployment of software [26]. Practices that have posed as barriers to continuous deployment include time pressure, increased technical debt, customer unwillingness to update and conflicting goals between rapid released and achieving high reliability and test coverage. In addition, the adoption challenges that have also been identified in large scale organisations are cultural barriers, risk of disintermediation of roles, lack of DevOps education and awareness, resistance to change, silo mentality and lack of strategic direction from senior management [36].

In general, organisations and IT professionals place DevOps in high regard, but DevOps practices adoption is associated with challenges. These challenges can arise mainly from a combination of necessity in maintaining a legacy system, lack of senior management buy-in, managerial structure, and resistance [21]. Other points which pose as barriers are blame-culture, communication difficulties, and delays in producing software releases [4] [30]-[34].

C. *Measuring DevOps*

Metrics in traditional highly structured corporate environments produce development cycles that focus a lot on defect density of the software product: yet, this is not the most effective way to measure quality in the context of software product development [6] [7]. The effect that traditional approaches have had to software development is that 'surrogation' can lead to enterprise strategy being replaced with metrics [27], with employees consciously aiming to contribute to local optima rather than global corporate optima to increase flow in the value stream [8].

Software development teams commonly express significant differences in behavioural patterns of developers and testers when senior management first establishes a key performance metric of 'least defects in deployable code' into a production environment and announce the downsizing of the quality assurance team [6]. Software development should be attempting to get closer to the metrics most frequently utilised to evaluate the speed with which releases can move to production environments before performance inefficiencies start to appear [6]. Additionally, software development pipeline health is essential to maintaining high quality software. Measurement approaches in DevOps teams include, but are not limited to, source code version control, optimum branching strategy, static analysis, >80% code coverage, vulnerability scan, open source scan, artifact version control, auto provisioning, immutable servers, integration testing, performance testing, build deploy testing automated for every commit, automated rollback, automated change order, zero downtime release, feature toggle [12].

In addition to the aforementioned, there is increased research interest in understanding how DevOps teams measure cognitive load using relative domain complexity without measuring lines of code produced, number of modules, classes, or methods [7]. This can be further complemented by flow metrics – i.e., flow distribution, flow velocity, flow time, flow load, flow efficiency [10], which represent the proportion of each flow work item being active in a given sprint. In particular, flow velocity measures features, defects, risks and technical debt in the product development flow whereas flow time resembles lead time and process time as defined in value stream maps [27]. Moreover, flow load represents active or waiting work in the value stream, and flow efficiency is the result of measuring flow load, i.e., duration of work inactivity in the value stream.

Workflow can be further categorised according to the Deployment Pipeline stages [11]. At the requirements planning level, new and unique work, including repetitive work, is considered for optimisation purposes. Moreover, optimizing it requires fast feedback and a focus on end-to-end cycle time for an all-round customer feedback.

Another dimension to DevOps can be Microsoft's perception on the triage of people, process and technology while providing a strong focus on the following seven DevOps habits [28]:

1. Flow of customer value.
2. Team autonomy and enterprise alignment.
3. Backlog groomed with learning.
4. Evidence gathered in production.
5. Managing technical debt.
6. Production-first mindset.
7. Infrastructure is a flexible resource.

In regard to the seven habits, firstly, flow of customer value entails automated testing, Continuous Integration (CI), Continuous Deployment (CD) and release engineering and management. Moreover, scaling that in terms of agile to self-managing teams and feature crews regards team autonomy and enterprise alignment. Thirdly, within Microsoft feature crews, another habit is to refine and reprioritise backlog items through usage monitoring, telemetry, Testing In Production (TIP) and stakeholder feedback. In fact, evidence collected from production environments include all aforementioned steps for backlog refinement plus the use of feature flags and continuous experimentation, regarded as one of key DevOps practices. In addition, managing technical debt concerns peer code reviews, automated testing, continuous measurement and agile documentation. In terms of production first mindset application performance management and Infrastructure-as-Code (IaC) play big role in achieving it, coupled with configuration management and automated recovery. Finally, IaC, automated scaling, sandboxing for development and test environments as well as the usage of microservices and containers make Infrastructure a flexible resource to work with while adopting DevOps practices and principles.

The aforementioned literature on DevOps metrics at the team structure-process-toolset level should be taken into account in a cross-functional manner and be communicated transparently to both leadership and engineering teams to establish progress and quality in a consistent format [9].

III. RESEARCH METHOD

Having defined the agile, lean, and DevOps adoption benefits and challenges described in literature, it is crucial to determine whether these views align with industry domain practitioners.

A. *Research Design*

This paper presents contextually relevant data generated from thirty (30) semi-structured interviews, see Figure 1, that were conducted between September 2018 and January 2019 with practitioners in companies working within a wide range of countries (Czech Republic, Estonia, Italy, Georgia, Greece, The Netherlands, Saudi Arabia, South Africa, United Arab Emirates (UAE), United Kingdom). Additionally, a survey was conducted during the period August 2019 and December 2019 whereby the responses of 250 participants were recorded.

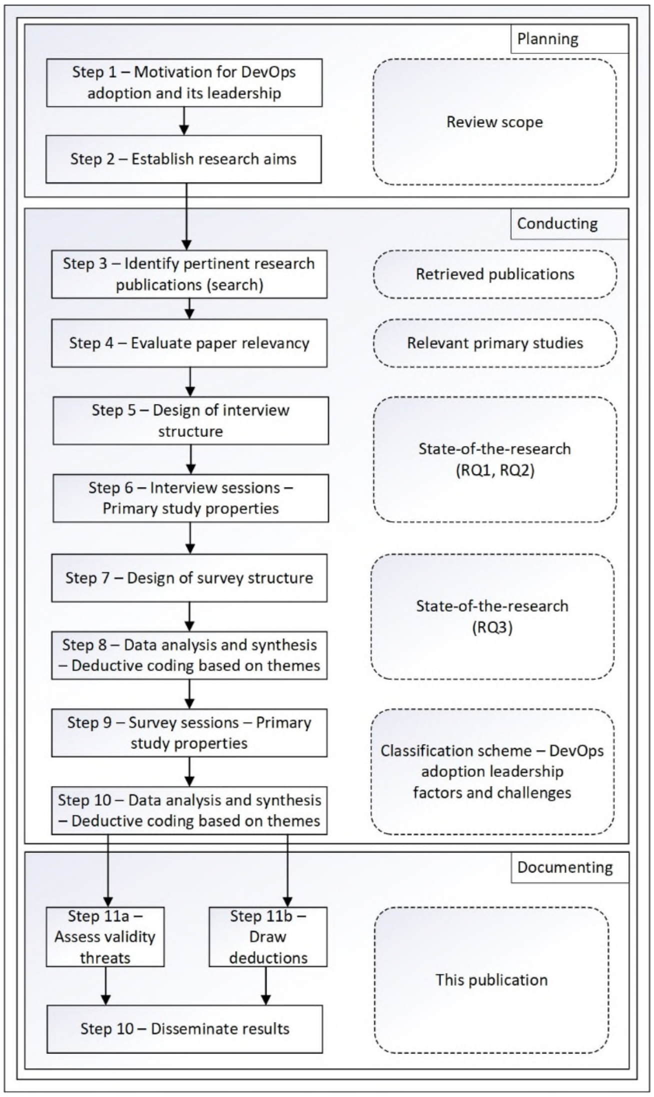
Planning
Step 1 – Motivation for DevOps adoption and its leadership
Step 2 – Establish research aims
Review scope
Conducting
Step 3 – Identify pertinent research publications (search)
Retrieved publications
Step 4 – Evaluate paper relevancy
Relevant primary studies
Step 5 – Design of interview structure
Step 6 – Interview sessions – Primary study properties
State-of-the-research (RQ1, RQ2)
Step 7 – Design of survey structure
Step 8 – Data analysis and synthesis – Deductive coding based on themes
State-of-the-research (RQ3)
Step 9 – Survey sessions – Primary study properties
Step 10 – Data analysis and synthesis – Deductive coding based on themes
Classification scheme – DevOps adoption leadership factors and challenges
Documenting
Step 11a – Assess validity threats
Step 11b – Draw deductions
Step 10 – Disseminate results
This publication

Figure 1. Research study process.

B. *Data Collection for Interviews*

The interview participants were identified with their roles, organisation size and country within which they work. Participants selected for the interview process had previous experience of agile, lean and DevOps practices and principles. We invited participants through IT events in Europe and through professional social media networks, see Table I.

TABLE I. LIST OF INTERVIEW PARTICIPANTS

	Industry Practitioner Profile		
	Job Title	***Country of Work***	***Domain***
P1	PMO Director	Saudi Arabia	Aviation
P2	Principal Consultant,	Italy	IT

	Industry Practitioner Profile		
	Job Title	***Country of Work***	***Domain***
	IT Service Management		Consulting Services
P3	CIO	Greece	Insurance
P4	Principal Consultant, IT Service Management	UK	IT Consulting Services
P5	Managing Director, IT Service Management	UK	IT Consulting Services
P6	Smart Systems Manager	Greece	IT Consulting Services
P7	Senior Digital Transformation Technologist & Solution Practice Lead	United Arab Emirates	IT Consulting Services
P8	Principal Consultant, IT Service Management	United Kingdom	IT Consulting Services
P9	Founding Consultant, IT Service Management	United Kingdom	IT Consulting Services
P10	Managing Director	United Kingdom	IT Consulting Services
P11	Head of Remote Transactions	Greece	Banking
P12	Consultant	Netherlands	IT Consulting Services
P13	Deputy Chief Information Officer	Greece	Construction Management
P14	Head of Applications	Greece	Lottery
P15	Principal Consultant, IT Service Management	South Africa	IT Consulting Services
P16	Founding Consultant, IT Service Management	United Kingdom	IT Consulting Services
P17	Managing Director, IT Service Management	United Kingdom	IT Consulting Services
P18	Managing Director and Lead Consultant	United Kingdom	IT Consulting Services
P19	IT Operations Manager	Greece	Lottery
P20	IT Operations Manager	United Kingdom	Government
P21	Founding Consultant, IT Service Management	United Kingdom	IT Consulting Services
P22	Assistant General Manager, IT Operations	Greece	Banking
P23	Chief Digital Office	Estonia	Government
P24	Chief Information Officer	Greece	Insurance
P25	Chief Information Officer	Greece	Aviation
P26	Development Team Lead	Greece	Lottery
P27	IT Operations Lead	Georgia	Government
P28	Business Development Director	Greece	IT Consulting Services
P29	Operations and Innovation Lead, IT Services	Czech Republic	Courier Services
P30	CIO	Greece	Automotive

To achieve a heterogeneous perspective, and to increase the wealth of information, practitioners from a variety of organisations were invited and consulted. The information provided to interview participants prior to the interview commencing stated that names or organisation titles would not be disclosed as part of this research.

Data collection and analysis was mapped to answer the research questions posed at the end of Introduction section, see Table II. The entire set of interview questions is accessible at [46].

TABLE II. RESEARCH QUESTIONS MAPPED TO INTERVIEW QUESTIONS

Research Question	**Interview Question (No.)**
Data collection for segmentation purposes e.g., participant age, professional experience, job role, country of work, industry of work.	1, 2, 3, 21
RQ1) Leadership characteristics required to enable DevOps practice and principle adoption	17, 18, 19, 20, 21
RQ2) What are the DevOps adoption inhibitors (resistance factors)?	4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 21
RQ3) How should DevOps leadership be measured?	4, 20, 21

There were twenty (20) questions - consisting of two types of questions – participant demographics questions and questions mapped to the three research questions in this paper.

The interview series consisted of thirty (30) participants from nine countries Greece (11), UK, (10), Saudi Arabia (2), Czech Republic (1), Estonia (1), Georgia (1), Italy (1), Netherlands (1), South Africa (1), United Arab Emirates (1). Fifteen (15) were IT consultants and another fifteen (15) were from service provider organisations. The service consumers of IT consultants can be service providers or other IT consultants. The service consumers for the service provider organisation can also be either internal or external. All Greek participants were service providers. UK participants consisted of nine (9) consultants and one (1) service provider. There was a distinct diversity of participant roles, e.g., Principal Consultant (10), Managing Director (4), Chief Information Officer / Chief Digital Officer (6), IT Operations Manager (3), PMO Director (1), Head of Remote Transactions (1), Smart Systems Manager (1), Head of Applications (1), Development Team Lead (1), Business Development Director (1), Operations and Innovation Lead (1). Furthermore, the industries of participants were Consulting Services (14), Aviation (3), Government (3), Lottery (2), Insurance (2), Finance (2), Manufacturing (1), Logistics (1), ISV (1), Automotive (1).

The interview participants were aware of, and had considerable previously experience applying a range of frameworks, international standards, methodologies, practices, and principles; such as ITIL (26), SCRUM (22), DevOps (19), Lean IT (15), ISO20000 (8), PMBOK (10), PRINCE2 (8), XP (4), SAFe (3).

C. Data Collection for Survey

The survey was divided into four sections: 1) questions about the participant's professional information; 2) questions about DevOps practices adopted, 3) questions about leadership related to DevOps, and 4) questions on DevOps metrics. The target audience of the survey is defined mainly as Consultant, Product/Software Developer, C-Suite, Operations engineer, IT Architect. The entire set of survey questions is accessible at [47].

TABLE III. RESEARCH QUESTIONS MAPPED TO SURVEY QUESTIONS

Research Question	Survey Question (No.)
Data collection for segmentation purposes e.g., participant age, professional experience, job role, country of work, industry of work.	1, 2, 3, 4, 5, 20
RQ1) Leadership characteristics required to enable DevOps practice and principle adoption	7, 8, 13, 14, 17, 18, 19, 20
RQ2) What are the DevOps adoption inhibitors (resistance factors)?	1, 12, 13, 14, 20
RQ3) How should DevOps leadership be measured?	8, 11, 15, 16, 20

The 250 participants of the survey answered six demographics questions. The participant role segmentation is shown in Figure 2.

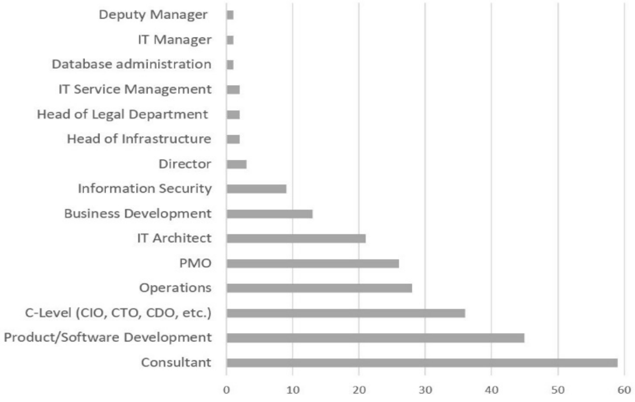

Figure 2. Survey participant job role.

Moreover, the industries in which the survey participants worked in are IT Services/Consulting (33%), Government (22%), Financial Services (13%), Technology/Telecommunications (8%), Manufacturing (4%), Financial Services/Consulting (3%), Aviation (3%), Construction (3%), Retail/Consumer Services (2%), Healthcare (2%), Education (2%), Recycling (1%), Insurance (1%), Energy/Utilities (1%), Leisure & Hospitality (1%).

IV. RESULTS

A. DevOps Adoption and its Challenges

In terms of DevOps adoption inhibitors and resistance factors, P15 (Principal Consultant, South Africa) mentioned that "Extremely hierarchical organisational structures are communication barriers to DevOps adoption". Another failure point for DevOps adoption can be that "DevOps practice adoption has to be at a wider enterprise scale for it to be labelled successful". In addition, P27 (IT Operations Manager, Georgia) stated that "Top management is not interested in agile and DevOps practice adoption. They do care about customer satisfaction levels, which can mean a reactive attitude towards the number of complaints received". Notably, P3 (CIO, Greece) mentioned that "We identified the bottlenecks that we adopted while adopting these structured approaches". However, P8 (Principal Consultant, UK) argues that "senior management and team members should not blame the person who introduced the new practice" since "continuous experimentation is crucial to the success of DevOps adoption and any new practice adoption". It is vital to establish the right organisational culture when it comes to the shift of mindset that DevOps adoption requires. To that extent P10 (Managing Director, UK) stated that "the team leading the adoption of the new way of working has to have the right skills and cultural drivers to succeed".

In the survey of 250 participants, there were certain close-ended questions which aimed to unravel more around DevOps practice and principle adoption and Figure 3 indicates the results.

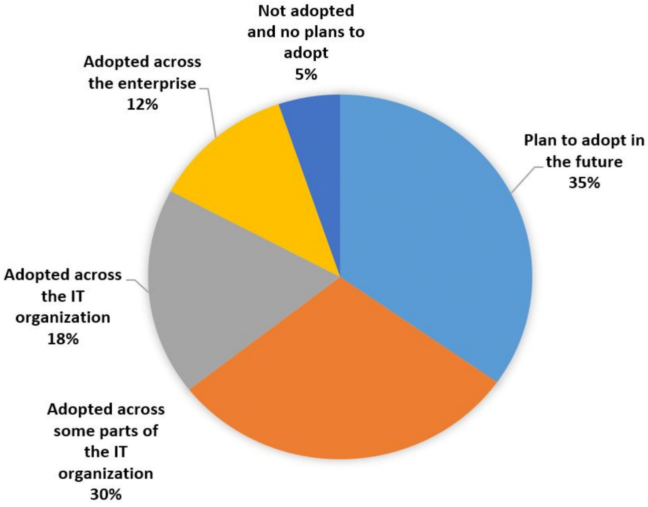

Figure 3. DevOps adoption stage of survey participants.

In addition, the roles responsible for the decision making process in DevOps adoption initiatives are shown in Table IV.

TABLE IV. DECISION MAKING ROLE IN DEVOPS ADOPTION PROCESS

Role responsible for decision making in DevOps adoption process	Participant Preference (%)
C-Level (Chief Information Officer, Chief Digital Officer, etc.)	33.6
Development Lead	20.8
Product Owner	16
Architect	10.4
Operations Lead	6
Business Domain	3.6
DevOps Engineer	3.2
Developer	3.2
System/Network/Database Administrator	1.2
Executive Committee	0.8
Team Leader	0.4
Analyst	0.4
Not Sure	0.4

It is worrisome that Information Security and DevOps Engineer are given low importance. In addition, there seems to be low involvement of the business domain in DevOps adoption initiative. On the other hand, the high concentration of responses to C-level executive (Chief Information Officer, Chief Digital Officer, etc.) and development team lead could suggest that the development teams themselves have to shift from a highly hierarchical organisational structure to more autonomous self-organising team behaviours, which characterise DevOps teams.

Lack of commitment by customer is recognised as the top inhibitor and resistance factor of DevOps adoption followed by lack of organisational practice adoption capability. A 4-point Likert scale was chosen for this question to record opinions. These results are similar to the overall expressed opinion during the interviews and indicate that there is overwhelming agreement on these types of inhibitors to DevOps adoption. Having identified the set of most frequently adopted DevOps practice and principles, the next section attempts to provide clarity on DevOps adoption leadership.

B. DevOps Leadership

It is worth looking into the level of acceptance of a leadership role being an individual or team role and the influential effect it can have on team performance in the context of software product development and coding pipeline health. Nine (9) service providers and (6) consultants agreed that the leadership role should be an individual role whereas five (5) service providers and five (5) consultants agreed that the leadership role should be a team role. Lastly, one (1) service provider and three (3) consultants stated that both approaches are required interchangeably throughout the course of a transitioning initiative towards DevOps practice and principle adoption.

Throughout the series of interviews, there was focus on DevOps adoption and the leadership role. In fact, P5 (Managing Director, UK) and P19 (IT Operations Manager, Greece) stated that "Leadership skillset is the most important thing to adoption barrier breakdown". P7 (Consultant, United Arab Emirates) stated that "In the beginning of an adoption initiative there is a constant link to fear of people for loss of power, loss of position, etc.". Moreover, P12 (Principal Consultant, Netherlands) mentioned that there is "Lack of Leadership (walk-the-talk, lead by example, confront 'undesirable behaviours, reward new behaviours)". In addition, P23 (CDO, Estonia, P28 (Business Development Director, Greece) and P30 (CIO, Greece) added that "end-to-end ownership of the leadership role is required in terms of cross-functional team leadership".

Moreover, the survey showed that 76% of participants have held or hold a leadership position and 91% claimed that DevOps leadership role is required and that it should be an individual role (67%). These results are similar to the results produced from the thirty (30) interview participants.

C. DevOps Metrics

The interview series revealed that version control and issue tracking have been vastly adopted by the respondents i.e., 95%. Additionally, performance monitoring, test automation and automated deployment seem to have important penetration in the software product development practices. On the contrary, Infrastructure-as-Code, code coverage, static code analysis, trunk-based development, automated provisioning of IT resources, and containerised environments didn't score as high as the aforementioned, three areas.

The main aim of this survey section was to uncover more around the metrics related to DevOps adoption and its leadership role. DevOps adoption practices and principles adoption levels can be measured with the ways indicated in Figure 4.

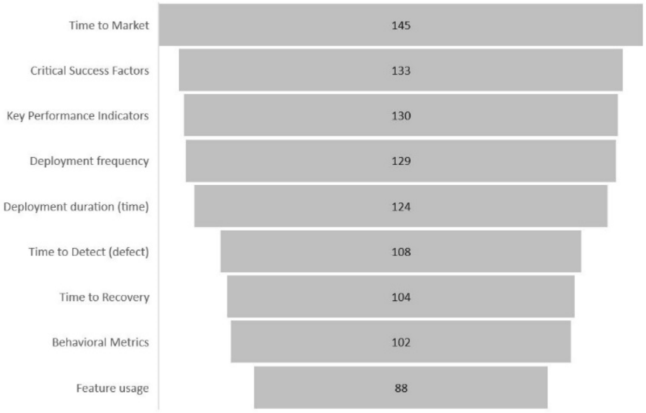

Figure 4. DevOps adoption metrics indicated by survey participants.

The traditional approach to measuring adoption in software development surfaced in the results shown in

Figure 4, in terms of time to market, Key Performance Indicators (KPI) and Critical Success Factors (CSF). The most prominent DevOps oriented metrics were deployment frequency, deployment duration, time to detect a defect, time to recovery and behavioural metrics. Feature usage seems to be an emerging practice for DevOps adoption. Moreover, 88% of respondents agreed that the leadership role should be associated and have ownership of the aforementioned metrics in order to facilitate the DevOps teams efforts in the adoption of practices and principles. Lastly, regarding the software development-oriented metrics described in section Background and Related Work – Measuring DevOps, there was negligible mention in the interviews and the survey.

D. Research Validity

Initially, we considered the internal validity. The main validity threat relates to possible bias in the participant selection process. The communication channels, utilised to invite interview and survey participants, were conferences of DevOps, Lean IT and IT service management. In addition, the majority of interview participants related their work to closed-sourced software products. Future study could focus on DevOps adoption leadership by considering Open-Source Software (OSS) products. Next, we considered the construct validity. A threat to construct validity is that half of the questions of the online survey consisted of closed-ended questions. The authors evaluated the survey structure and deduced that the advantages of closed-ended questions outweighed the disadvantages. Furthermore, concerning external validity, although the viewpoint of the interviewed and surveyed practitioners is considered with different backgrounds, working in varying industry domains and geographical regions, the authors do not claim that research results from this contribution are valid to other scenarios. However, saturation was achieved after the 20th interview.

V. CONCLUSIONS

This paper indicates that DevOps practice and principle adoption maintained strong linkage to agile and lean practice and principle adoption for thirty (30) interview participants from private and public sectors in the EMEA region. In addition, the evaluation of a survey completed by 250 participants, of which 76% have held previous leadership positions further enhanced the linkage of DevOps, agile and lean practices and principles. Moreover, a mixed methods approach was used. The 30 interviews generated coded themes to expand our understanding of relevant factors – from most to least recurring in interview transcripts; (1) DevOps leadership, (2) practice and principle adoption, (3) employee culture, (4) product development and (5) skills. The data collected from a series of interviews and a survey indicate a clear list of specific agile, lean and DevOps practices and principles including leadership characteristics which form a crucial part to DevOps adoption theory.

A. Discussion

The most important findings of this review, which are organised according to the study's research questions, are summarised below.

RQ1) What are the leadership characteristics required to enable DevOps practice and principle adoption?

From the 250 survey participants, with 81% possessing over 10+ years of professional experience, results indicated that a new practice and principle adoption leadership role should exist for transformation initiatives; i.e., that the C-Suite should be the direct report of the DevOps leader. The most prominent identified DevOps leadership characteristics associated to leadership skills were: communication and collaboration, customer-centric mindset, having a technical background, and being an active listener.

The results obtained from the survey participants shed more light on the already established beliefs extracted from the interview participants. For instance, there was strong indication by interview participants that a shift of skillset towards acquiring, developing and applying more soft skills is necessary to achieve new practice and principle adoption, in this case agile, lean and DevOps. In fact, communication and collaboration as well as customer-centric or even customer-obsessed mindset is an extension to that viewpoint. Another example spurs from the technical and/or business backgrounds that could play a role in DevOps adoption leadership. Ever since the term "DevOps" was coined back in 2009, the worldwide IT and business community have come to an assumingly obvious realisation; "DevOps" is associated to the IT organisation and that is where it stays. This belief seems to reflect in the survey findings where possessing a technical background is more important than a business background by as much as 15% in the "Strongly Agree" category. However, the survey findings also suggest that possessing a business background is beneficial to a certain extent with interview participants state that a balanced background is preferable to technical-only or business-only.

The least important DevOps leadership characteristics were: gaining a relevant certification, design thinking, previous experience on transformation projects and talent seeking. Furthermore, the Information Security Officer is mostly seen as a collaborator to the DevOps adoption leader.

Survey results indicate that certification was, by a considerable degree, the least preferred characteristic for the DevOps leader. Although there is availability of DevOps leader certifications e.g., DevOps Leader (DOL) certification, by the DevOps Institute, it seems that the desire to become certified in DevOps leadership is not regarded to be an important characteristic or requirement. In addition, design thinking which entails observation, insights generation, ideation, prototype and testing for product development purposes was clearly not considered a crucial characteristic or requirement. Furthermore, previous experience of transformation projects did not yield any connection to DevOps leadership. The authors' intent was to investigate a finding from the interview series, where there was an indication that constant coaching by an external entity e.g., consultant is required, although not always, to sustain transformation initiatives. However, most of the time, the IT organisation cannot sustain newly adopted practices in their structure and default to the "old habits of working", which could suggest that an individual with previous experience on transformation projects would know how to avoid a similar situation in the transition process to DevOps practice and principle adoption.

RQ2) What are the DevOps adoption inhibitors?

The analysis and evaluation of interviews showed that several DevOps adoption inhibitors were recognised (1) communication barriers, (2) lack of cross-functional collaboration, (3) lack of senior management buy-in, (4) lack of leadership, (5) lack of cross-functional leadership, (6) lack of enterprise-wide DevOps adoption, (7) plethora of IT systems coupled with numerous IT support roles and (8) lack of cross-functional collaboration. In addition, the survey added the (9) lack of commitment by customer and (10) lack of organisational practice adoption capability.

The interview participants established that the cultural behaviour of making organisational group distinctions of defining responsibility, especially in terms of "us" and "them", is immensely detrimental to the cross-functional team collaboration mode and the cross-functional leadership DevOps aims to achieve. In essence, this inhibitor leads to DevOps enterprise-wide adoption facing failure from the off start of such an initiative, implying that it is important to first let the cultural character within the IT organisation take form and shape and then aim for adoption at a wider scale, outside the IT organisation. To that extent, the interviews showed that Human Resources departments can be a first step outside the IT organisation where DevOps adoption can contribute in terms of shift of culture-skillset-toolset. Simply put, as one interviewee stated, "Leadership skillset is the most important thing to adoption barrier breakdown". In addition, the set of inhibitors identified could have a direct cause of exacerbation from the perspective of the Human Resources department, whereby utilising a rudimentary selection approach that qualifies new hires based on the right toolset experience without considering mindset and skillset-specific aspects falls short of DevOps-oriented team structure expectations. Thus, this selection process could insinuate that IT teams that fail or partially fail to adopt DevOps practices and principles are because the transition to the right mindset e.g., embrace continuous experimentation, cross-collaboration between development, operations, quality assurance and information security teams, etc. and skillset is simply, under-developed. There are findings in the survey to indicate that talent seeking is not considered an important characteristic of the DevOps leader since that is a responsibility area normally covered by Human Resources. Therefore, the perception that DevOps teams and their leaders should not engage or engage minimally with talent seeking opportunities could affect the future staffing of those teams.

RQ3) How should DevOps leadership be measured?

During the survey, participants indicated that DevOps adoption leadership practices should still be governed by traditional approaches, such as CSF, KPI and time-to-market. However, agile and lean metrics formed a significant part of the wider picture with the top five most popular being (1) deployment frequency, (2) deployment duration, (3) time to detect defect, (4) time to recovery , and (5) behavioural metrics. The DevOps-oriented metrics from the aforementioned, i.e., (1) to (4), indicate software product development measurements that can apply to a DevOps team structure as well as to the DevOps leadership role. From the cultural perspective, (5) can refer to behaviour that aims to increase knowledge sharing in cross-functional fashion, the frequency that a leader performs one-to-ones with DevOps teams and their members to understand what is on top of mind, similar to Gemba walks in the physical or virtual format, the number of absentees during DevOps adoption sessions, etc.

Moreover, feature usage is an emerging practice for DevOps adoption and it regards monitoring usage of a released product feature in a production system environment and whether performance is as expected. Lastly, the vast majority of respondents agreed that the leadership role should be associated and have ownership of the aforementioned metrics.

Presently, we conclude that DevOps adoption leadership is very much a multidisciplinary topic requiring a specific identified skillset coupled with a set of DevOps practices and principles. The leadership approach of the organisational structure is vital to the level of resistance exhibited by IT professionals during the transitioning period from a highly structured software product development approach to DevOps. We also deduce that the transitional phase of DevOps adoption requires an individual to lead DevOps teams which leads to the belief that DevOps has a substantial leadership component at its transitional level.

B. Future Research Directions

DevOps adoption leadership and its relationship to software product development teams is becoming a vastly popular research topic. The authors' intent is to maintain focus on the analysis and evaluation of presently collected research data and to provide further insights relative to current findings in order to witness which leadership styles can become part of the transitional journey of organisations towards DevOps practice and principle adoption. The organisational change required to achieve a successful state of a DevOps-oriented environment in today's global market raises a number of challenges and resistance factors in terms of the triage of mindset-skillset-toolset. The effects of the change need to be continuously monitored to identify the link to the shift of the triage experienced. In that aspect, one of the future research aims could be to invite and/or select IT practitioners with prior and/or current Open Source Software (OSS) experience. Additionally, it is the authors' belief that DevOps adoption velocity and its continuous applicability whether in an IT organisation-wide or enterprise-wide context, regards the levels of cognitive load under which DevOps team structures learn to perform. Therefore, future research could focus in gaining more insights on the extent of influence posed on DevOps teams and their leadership role due to cognitive load. Lastly, the current pandemic crisis, which has shifted the working experience to its virtual format, colocation; one of DevOps practices, for software product development, operations, quality assurance and information security teams is no longer the case. As long as the "work-from-home" paradigm is enforced in the global software product development community teams, that could potentially be affecting the interplay of DevOps adoption leadership characteristics and can be part of future research considerations.

REFERENCES

[1] P. Kruchten, "The rational unified process: an introduction," Addison-Wesley Longman Publishing Co., Inc., USA, 1999.

[2] K. Beck, "Extreme programming explained: embrace change," Addison-Wesley, Don Mills, Ontario, Canada, 2000.

[3] M. Fowler, "Refactoring: improving the design of existing code," Addison-Wesley, Don Mills, Ontario, Canada, 1999.

[4] L. Bass, I. Weber and L. Zhu, "DevOps: A Software Architect's Perspective," Addison Wesley, 2015.

[5] AXELOS, "ITIL4® Managing Professional High Velocity IT," The Stationery Office, London, UK, 2020, ISBN: 9780113316403.

[6] M. Herring, "DevOps for the Modern Enterprise," IT Revolution, Portland, Oregon, 2018.

[7] M. Kersten, "From Project to Product," IT Revolution, Portland, Oregon, 2018.

[8] E. Goldratt, "Theory of Constraints and How it Should be Implemented," North River Press, 1994.

[9] M. Herring, D. DeGrandis, N. Forsgren and S. Guckenheimer, "Measure efficiency, effectiveness and culture to optimize devops," IT Revolution, Portland, Oregon, 2015.

[10] G. Gruver, "Start and Scaling DevOps in the Enterprise," Bookbaby, 2016.

[11] K. Martin and M. Osterling, "Value stream mapping: how to visualize work and align leadership for organizational transformation," McGraw-Hill Education, UK, 2014.

[12] M. Nygard, T. Pal, S. Magill, S. Guckenheimer and J. Willis, "DevOps Governance Architecture," IT Revolution, Portland, Oregon, 2019.

[13] H. Alahyari, T. Gorschek and R. B. Svensson, "An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations," Information and Software Technology, vol. 105, pp. 78-94, Jan. 2019, doi.org/10.1016/j.infsof.2018.08.006.

[14] P. Rodríguez et al., "Chapter Four - Advances in Using Agile and Lean Processes for Software Development," Advances in Computers, Elsevier, vol. 113, pp. 135-224, 2019, doi.org/10.1016/bs.adcom.2018.03.014.

[15] W. J. W. Geurts, "Faster is Better and Cheaper," Wiley Online, vol. 26, pp. 1002-1015, Jul. 2016, doi.org/10.1002/j.2334-5837.2016.00207.x.

[16] T. Schlossnagle, "Monitoring in a DevOps world," ACM Queue, 2017, dl.acm.org/doi/pdf/10.1145/3178368.3178371.

[17] J. Willis. *What DevOps means to me*. [Online]. Available from: https://blog.chef.io/what-devops-means-to-me/, 2020.10.16

[18] P. Rodríguez et al., "Continuous deployment of software intensive products and services: A systematic mapping study," Journal of Systems and Software, vol. 123, pp. 263-291, 2017, doi.org/10.1016/j.jss.2015.12.015.

[19] DevOps Institute. *DevOps Collective Body of Knowledge* [Online]. Available from: devopsinstitute.com/resources/, 2020.10.16

[20] T. Dingsøyr and C. Lassenius, "Emerging themes in agile software development: Introduction to the special section on continuous value delivery," Information and Software Technology, pp. 56-60, 2016

[21] S. Jones, J. Noppen and F. Lettice, "Management challenges for DevOps adoption within UK SMEs." ACM, 2016, 978-1-4503-4411-1/16/17.

[22] W. Xiaofeng, K. Conboy and O. Cawley, ""Leagile" software development: An experience report analysis of the application of lean approaches in agile software development," J. Syst. Softw. Vol. 85, 2012.

[23] Puppet, DORA. *State of DevOps Report 2019* [Online]. Available from: puppet.com/resources/report/state-of-devops-report/, 2020.10.16

[24] J. Iden, B. Tessem and T. Päivärinta, "Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts," Information and Software Technology, vol. 53(4), pp. 394-406, 2011, DOI: 10.1016/j.infsof.2010.12.002.

[25] D. Feitelson, E. Frachtenberg and K. Beck, "Development and Deployment at Facebook," IEEE 1089-7801/13, 2013.

[26] J. Humble and J. Molesky, "Why enterprises must adopt devops to enable continuous delivery," Cutter IT Journal, vol. 24(8), pp. 6-12, 2011.

[27] Harvard Business Review, M. Harris, B. Tayler. *Don't Let Metrics Undermine Your Business* [Online]. Available from: hbr.org/2019/09/dont-let-metrics-undermine-your-business/, 2020.10.16

[28] Azure DevOps Microsoft Documentation. *DevOps at Microsoft* [Online]. Available from: docs.microsoft.com/en-us/azure/devops/learn/devops-at-microsoft/, 2020.10.16

[29] L. E. Lwakatare, P. Kuvaja and M. Oivo, "Relationship of DevOps to Agile, Lean and Continuous Deployment," 17th International Conference on Product-Focused Software Process Improvement (PROFES), Nov. 2016, pp. 399-415, doi:10.1007/978-3-319-49094-6.

[30] B. B. N. de França, H. Jeronimo and G. H. Travassos, "Characterizing DevOps by hearing multiple voices," Proceedings of the 30th Brazilian Symposium on Software Engineering (SBES), Association for Computing Machinery, New York, pp. 53–62, 2016.

[31] A. Dyck, R. Penners and H. Lichter. 2015. Towards Definitions for Release Engineering and DevOps, 3rd International Workshop on Release Engineering (RELENG), doi: 10.1109/RELENG.2015.10

[32] L. E. Lwakatare, P. Kuvaja and M. Oivo, "An exploratory study of DevOps: extending the dimensions of DevOps with practices," 11th International Conference on Software Engineering Advances, pp. 91–99, IARIA, Rome, 2016.

[33] J. Smeds, K. Nybom and I. Porres, "DevOps: A definition and perceived adoption impediments," Agile Processes in Software Engineering and Extreme Programming (XP2015), Lecture Notes in Business Information Processing, vol. 212, Springer, Cham, 2015.

[34] R. Jabbari, N. bin Ali, K. Petersen and B. Tanveer, "What is DevOps? A Systematic Mapping Study on Definitions and Practices," Proceedings of the Scientific Workshop Proceedings (XP2016). Association for Computing Machinery, New York, NY, USA, Article 12, pp. 1–11, doi.org/10.1145/2962695.2962707.

[35] N. Kerzazi and B. Adams, "Who needs release and devops engineers, and why?," Proceedings of the International Workshop on Continuous Software Evolution and Delivery (CSED2016). Association for Computing Machinery, New York, NY, USA, pp. 77–83, 2016, doi.org/10.1145/2896941.2896957.

[36] M. B. Kamuto and J. J. Langerman, "Factors inhibiting the adoption of DevOps in large organisations: South African context," 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, DOI: 10.1109/RTEICT.2017.8256556.

[37] C. P. Bezemer et al., "How is Performance Addressed in DevOps?" Proceedings of ACM/SPEC International Conference on Performance Engineering (ICPE 2019), Association for Computing Machinery, New York, NY, USA, pp. 45–50, 2019, doi.org/10.1145/3297663.3309672.

[38] W. Gottesheim, "Challenges, benefits and best practices of performance focused DevOps," Proceedings of the 4th International Workshop on Large-Scale Testing (LT2015), 2015.

[39] K. Maroukian and S. R. Gulliver, "Leading DevOps practice and principle adoption," Proceedings of the 9th International Conference on Information Technology Convergence and Services (ITCSE2020), AIRCC, Computer Science and Information Technology, pp. 41-56, 2020, ISBN13: 978-1-925953-19-0.

[40] R. K. Greenleaf, "Servant leadership: A journey into the nature of legitimate power and greatness", Paulist Press, 2002, ISBN: 978-0809105540

[41] G. A. Yukl and W. L. Gardner, III, "Leadership in organizations", Pearson Education, Essex, 2020, ISBN13:978-1-292-31440-2.

[42] J. Sutherland and K. Schwaber, "The Definitive Guide to Scrum: The Rules of the Game", 2017, USA.

[43] J. Anderson, "Kanban: Successful Evolutionary Change for Your Technology Business", 2010, Blue Hole Press, USA.

[44] M. Poppendieck and T. Poppendieck, "Lean Software Development: An Agile Toolkit", Addison-Wesley Professional, 2003, Boston, USA.

[45] M. Skelton, M. Pais, "Team Topologies", IT Revolution, 2019, Portland, USA.

[46] Research Interview Structure. *Interview Introduction and Settings* [Online]. Available from: https://tinyurl.com/ybxrcujq, 2020.10.16

[47] Research Survey Structure. *Survey* [Online]. Available from: https://tinyurl.com/yapl9u3u, 2020.10.16