Design and Validation of a Capability Measurement Instrument for DevOps Teams

A Participatory Action Research Approach

Olivia H. Plant[1,2](✉), Jos van Hillegersberg[1,3], and Adina Aldea[1,4]

[1] Industrial Engineering and Business Information Systems, University of Twente,
Enschede, The Netherlands
{o.h.plant,j.vanhillegersberg,a.i.aldea}@utwente.nl
[2] Quint Nederland B.V., De Oude Molen 1, Amstelveen, The Netherlands
[3] Tilburg University/JADS, Sint Janssingel 92, Den Bosch, The Netherlands
[4] LeanIX, Prins Bernhardplein 200, Amsterdam, The Netherlands

Abstract. This paper reports on the design and validation of a capability measurement instrument for software delivery teams that make use of the DevOps approach. The instrument is based on the results of a systematic literature review and was developed and validated by involving a total of five domain experts and conducting a field study among six DevOps team members. To this end, we used qualitative and survey-based data collection methods from participatory action research as well as design science. The resulting instrument encompasses five dimensions, covering seventeen capabilities and thirty-eight associated practices. The practices are evaluated on five capability levels. The results of the validation process indicate clear agreement of the domain experts and team members with all aspects of the instrument. As a contribution to practice, this research offers a pragmatic tool for IS practitioners which provides insight into the status of their DevOps transformation and offers directions for improving DevOps team performance. Furthermore, this research contributes to the ongoing research stream on DevOps by providing novel insights into the nature of DevOps capabilities and their potential configurations.

Keywords: DevOps · IS capabilities · Measurement instrument development · DevOps teams · Agile

1 Introduction

A growing amount of organizations is reorganizing their IT functions according to the DevOps paradigm. This calls for the establishment of cross-functional, agile teams that are responsible for development and operations of their systems and automate substantial parts of their processes [6,32]. While DevOps is becoming increasingly popular in practice, the approach has also attracted

V. Stray et al. (Eds.): XP 2022, LNBIP 445, pp. 151–167, 2022.
https://doi.org/10.1007/978-3-031-08169-9_10

growing attention from the IS research community over the past years. Multiple studies have attempted to create standardized definitions of DevOps [26] and identify its core elements [16] in order to foster a shared understanding of the paradigm. However, there is still no uniform definition of DevOps available [6,17]. Furthermore, there is little research-based guidance available to practitioners on how to implement DevOps and assess the current status of their transformation.

Prior research has related the implementation of IT capabilities to an increase in performance, both at team-level as well as on an organizational level [22,30]. We therefore propose to adopt a capability-based perspective when addressing the implementation of DevOps in organizations. Consequently, we argue that a standardized measurement instrument which evaluates the capabilities of DevOps teams will enable IT professionals to identify potential shortcomings or points for improvements in their transformation and will ultimately lead to an increase in team performance if the results of the measurement are addressed successfully.

While there have been efforts to create both industrial and scientific DevOps maturity models [34], to the best of our knowledge there is no instrument available which assesses the state of DevOps capabilities themselves. We therefore aim to develop a capability measurement instrument for DevOps teams which is based in extant academic literature but built in close collaboration with industry professionals in order to ensure its validity and practical use. Such a measurement instrument is expected to contribute to both the lack of a shared definition of DevOps and its practices as pointed out by Lwakatare, Kuvaja & Oivo [17] as well as provide a more structured approach for practitioners in how to implement DevOps and improve the performance of their DevOps teams.

This research makes use of the definition of a capability as proposed by Iacob, Quartel & Jonkers: *"A capability is the ability of an organization to employ resources to achieve some goal"* [14]. We furthermore build on the resource-based view and more specifically on the theory of dynamic capabilities [28] which argues that the competitive advantage of organizations lies within their resource base as well as in their ability to reconfigure their assets to address rapidly changing circumstances. According to Teece, Pisano and Shuen [28], these firm capabilities need to be understood in terms of managerial processes and organizational structures. Dynamic capabilities are idiosyncratic which makes them difficult to imitate for competitors [28]. However, Eisenhardt & Martin [5] suggest that while dynamic capabilities may be idiosyncratic in their details, they constitute a set of specific and clearly identifiable processes at a higher level. We therefore argue that it is possible to define a specific set of capabilities that are relevant to DevOps teams but that any measurement instrument of capabilities will need to capture various configurations of the same capability in order to account for their idiosyncratic implementation. Subsequently, our research is guided by the following main research question and sub-questions:

How to design a capability measurement instrument for DevOps teams?

(a) Which capabilities and practices are relevant for DevOps teams?

(b) How to assess varying configurations of capabilities with a measurement instrument?

2 Research Methodology

In order to develop the envisioned measurement instrument, we followed the procedural model proposed by Aldea & Sarkar [1] which is meant for developing valid and reliable measurement instruments for theoretical constructs. According to the aforementioned authors, the procedural model is suitable for researches in which the theory on which the instrument is based already exists and is sought to be empirically tested. The first stage of the model involves identifying theoretical constructs and candidate items which represent these constructs. The candidate items are then sorted into separate domain categories (substrata identification) from which a revised set of items is identified. These items are then further revised and improved. Finally, the instrument is validated in order to obtain evidence on the validity and reliability of the instrument.

An overview of all steps of the procedural model and the respective methodology applied in this research can be found in Table 1.

Table 1. Development of the DevOps capability measurement instrument

Instrument development stage [1]	Application to this research
1. Item creation	Systematic literature review
2. Substrata identification	Open and axial coding
3. Item identification	Domain expert workshops
4. Item revision	Domain expert interviews
5. Instrument validation	Domain expert evaluation survey & field study

2.1 Systematic Literature Review

The capabilities and practices that are part of the measurement instrument are based on the results of a systematic literature review (SLR) which we have conducted prior to this research and which we have detailed in a separate publication [21]. The review spanned 37 empirical research papers on DevOps capabilities and concepts. Data was gathered and synthesized by applying open and axial coding techniques in the qualitative data analysis tool Atlas.ti. To this end, we defined and applied codes to paragraphs of the papers which addressed capabilities and practices that were important for DevOps teams. The codes were continuously compared, merged or redefined and relationships between codes were established [33]. We then grouped the single codes into a more comprehensible set of code categories which resulted in an overview of DevOps practices and higher-level DevOps capabilities respectively. The core results of the review are summarized in Sect. 3.

2.2 Instrument Design

The capability measurement instrument was designed in close collaboration with industry practitioners by applying methods from Participatory Action Research (PAR). PAR seeks to combine theory and practice with the pursuit of designing practical solutions to pressing concerns of people [2]. This approach provides an opportunity for mutual learning and enriching dialogue between researchers and practitioners and is especially suitable when the nature of the artifact aligns with the participatory philosophy of PAR [24], as it is the case with our theory-based yet practically applicable measurement instrument.

Domain Expert Workshops. A first draft of the measurement instrument was created by conducting two workshops with a domain expert that served as a senior consultant at a Dutch consulting firm focused on digital transformations. This expert had vast experience with DevOps transformations and automation technologies.

Workshops are frequently used as qualitative data collection methods in PAR designs [3]. During the workshops, all candidate items were discussed in detail. Based on the suggestions made by the domain expert, items that displayed too much similarity to other items were eliminated in order to increase convergent and discriminant validity. Furthermore, one additional practice was added to the reference model based on the expert's suggestion. Additionally, all questions and answer options pertaining to the revised items were discussed and were clarified or supplemented with industry examples where applicable.

Domain Expert Interviews. The measurement items were further revised by interviewing four additional domain experts who also served as senior or principal consultants at a Dutch consulting firm. All of them had vast experience with Agile, DevOps or Lean methodologies and digital transformation projects in general. The capability measurement instrument was shared with the subjects before the interviews via e-mail.

The interviews had a semi-structured nature and were prepared beforehand through means of an interview guide [19]. The interviews lasted between 30 and 45 min. We started the conversation by introducing our research rationale and explaining our interpretation and definition of the concept of capabilities. We then discussed the capability levels with the interviewees and asked for their opinion on whether the scales and their definitions were understandable and covered all possible configurations of a DevOps capability sufficiently. This phase led to some minor adjustments in the capability level definitions. We then discussed the instrument taxonomy with the experts and asked whether the identified capabilities were indeed relevant for DevOps teams, whether there were any capabilities missing or redundant and whether the definitions of the capabilities were clear. The interviews led to the inclusion of another practice in the taxonomy and some minor adjustments regarding the names of some capabilities, the practices assigned to them and in the definitions of the capabilities and their measurement scales.

2.3 Instrument Validation

Maturity models can be evaluated through three different methodologies [23]: The first method is the evaluation of the instrument by the authors themselves. Another technique is the evaluation by domain experts which is performed through interviews, surveys or assignments. The last method is evaluation in a practical setting. The capability measurement instrument at hand was validated by applying a combination of domain expert evaluation and a field study. In doing so, we follow the suggestions of Venable, Pries-Heje and Baskerville [29] who propose to first evaluate design artifacts in an artificial setting, for example by using theoretical arguments, before moving towards a naturalistic evaluation in the real environment of the artifact.

Domain Expert Evaluation Survey. After the interviews, the four domain experts who were involved in the item revision stage were requested to fill in an online survey. They were asked to rate a number of statements regarding the instrument based on a five-point Likert scale, ranging from *strongly disagree* to *strongly agree*. The remaining domain expert who participated in the item identification workshops was not engaged in the validation of the measurement instrument due to their high involvement during the creation of the instrument.

The statements in the evaluation survey were based on the evaluation template for domain expert reviews of maturity models by Salah, Paige and Cairns [23]. The template was slightly adjusted to suit the nature of our capability measurement instrument better. The results of the survey indicate clear agreement of the domain experts with the validated aspects of the instrument. An overview of all statements and the mean agreement scores given by the four respondents as well as the standard deviations of these scores can be found in Table 2. [1].

Next to these statements, the experts were also asked a number of open questions focused on whether there were any questions, answers or descriptions which the respondents would add, remove or update and whether the model could be improved to make it more useful.

Field Study. Simultaneous to the expert validation, the instrument was presented to six DevOps team members from three different organizations. After taking the assessment, the team members were asked to rate a number of statements which were modified from the domain expert evaluation survey. The participants were solely asked to rate statements related to the understandability and ease of use of the instrument, as well as whether they thought that the capabilities covered all aspects relevant to DevOps teams. The evaluation of the underlying design of the instrument such as the sufficiency and accuracy of the capability levels or the general use in the industry were left to the domain experts and were not part of the field study evaluation. An overview of the validation statements, mean agreement scores and their standard deviations can be found in Table 2, along with the results of the domain expert validation survey.

[1] The individual scores given by the respondents will be provided upon request

Table 2. Validation survey statements and mean agreement scores from domain experts (n = 4) and field study participants (n = 6), based on a five-point Likert scale

Validation statements adapted from [23]	Domain experts	*Stand. Dev.*	Field study	*Stand. Dev.*
Capabilities & practices	**4.3**		**4.3**	
1. The capabilities and practices are relevant to DevOps teams (Relevance)	4.5	*0.5*	4.5	*0.5*
2. The capabilities and practices cover all aspects impacting/ involved in DevOps teams (Comprehensiveness)	4.0	*0.0*	4.0	*0.0*
3. Capabilities and practices are clearly distinct (Mutual Exclusion)	4.3	*0.4*	–	–
4. The answer options are clearly distinct (Mutual exclusion)	–	–	4.2	*0.4*
5. There are no questions asked more than once in the assessment (Mutual Exclusion)	–	–	4.3	*0.5*
Capability levels	**3.9***			
6. The five capability levels are sufficient to represent all states of a team capability (Sufficiency)	4.3	*1.3*	–	–
7. There is no overlap detected between descriptions of capability levels (Accuracy)	3.8	*0.4*	–	–
8. The question answers are correctly assigned to their respective capability level (Accuracy)	3.8	*0.4*	–	–
Capability assessment	**4.5***		**4.1**	
9. The capability descriptions are understandable (Understandability)	4.5	*0.5*	–	–
10. The capability levels are understandable (Understandability)	4.5	*0.5*	–	–
11. The questions and answers are understandable (Understandability)	4.3	*0.4*	4.3	*0.7*
12. The capability assessment is easy to use (Ease of use)	4.5	*0.5*	4.2	*0.9*
13. The capability assessment is easy to evaluate (Ease of use)	4.3	*0.4*	–	–
14. The capability assessment has the right length (Ease of use)	–	–	3.8	*1.2*
15. The capability assessment is useful for conducting assessments (Usefulness and Practicality)	5.0	*0.0*	–	–
16. The capability assessment is practical for use in industry (Usefulness and Practicality)	4.8	*0.4*	–	–

*Deviation from averages of values displayed in the table due to rounding errors.

3 Theoretical Framework

In a previous publication [21], we have extracted DevOps capabilities from extant literature and analyzed these in the light of the dynamic capabilities theory [27]. We then put forward the argument that DevOps teams can contribute to the competitive advantage of organizations by building capabilities that allow them to sense opportunities and threats, seize opportunities and rapidly transform their assets. The success of these capabilities however is dependent on the presence of a set of organizational enabler capabilities that allow the teams to perform their work independently and autonomously and work towards supporting the organizational strategy and vision. If these two sets of capabilities are implemented successfully, organizations can expect to achieve a third set of beneficial outcome capabilities. The identified DevOps team capabilities were divided into the classes *sensing*, *seizing* and *transforming* which is in line with the classification of dynamic capabilities by Teece [27]. An overview of the results of the literature review is given in Fig. 1.

DevOps teams need to develop capabilities on two levels: First, business-related capabilities concern structures, processes and habits in their way of working which the DevOps teams develop. Second, the teams need to develop technology-related capabilities which allow them to automate processes and perform monitoring activities.

In order to sense opportunities and act upon these, DevOps teams should design customer-centric processes [13,20] and have frequent information exchange with stakeholders [12]. Furthermore, they should have a clear process for translating customer wishes into requirements and manage the backlog [9]. At the same time, teams need to be venturous [31] and self-empowered by assuming responsibility and ownership of their system [10,25] so they can operate autonomously and take appropriate decisions quickly. This can be facilitated by building an open team culture which is focused on continuous improvement [20], sharing opinions [6] and in which team members trust and respect each other [26]. In order to shorten decision-making and authorization processes, teams should also be skilled at lean-process management [6] and collaborate well within the team as well as with other teams [7]. Once teams have decided to take action based on an identified opportunity or threat, they need to deal with changes

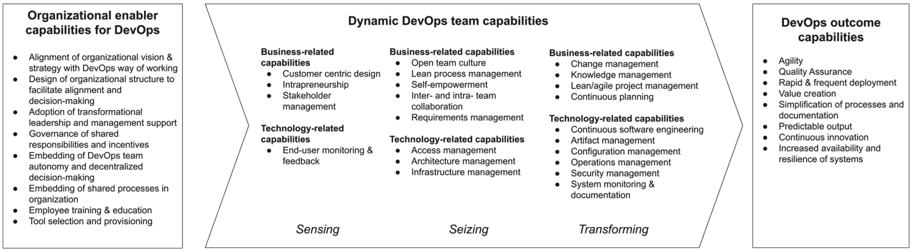
Organizational enabler capabilities for DevOps
Alignment of organizational vision & strategy with DevOps way of working
Design of organizational structure to facilitate alignment and decision-making
Adoption of transformational leadership and management support
Governance of shared responsibilities and incentives
Embedding of DevOps team autonomy and decentralized decision-making
Embedding of shared processes in organization
Employee training & education
Tool selection and provisioning
Dynamic DevOps team capabilities
Business-related capabilities
Customer centric design
Intrapreneurship
Stakeholder management
Technology-related capabilities
End-user monitoring & feedback
Business-related capabilities
Open team culture
Lean process management
Self-empowerment
Inter- and intra- team collaboration
Requirements management
Technology-related capabilities
Access management
Architecture management
Infrastructure management
Business-related capabilities
Change management
Knowledge management
Lean/agile project management
Continuous planning
Technology-related capabilities
Continuous software engineering
Artifact management
Configuration management
Operations management
Security management
System monitoring & documentation
Sensing
Seizing
Transforming
DevOps outcome capabilities
Agility
Quality Assurance
Rapid & frequent deployment
Value creation
Simplification of processes and documentation
Predictable output
Continuous innovation
Increased availability and resilience of systems

Fig. 1. Conceptual model of DevOps capabilities resulting from SLR [21]

effectively and timely [20]. This requires a flexible yet up-to-date planning process [26] as well as continuous exchange of knowledge and information [10] so team-members can assume multiple roles and responsibilities in this process.

On a technology-level, the automation of software delivery and provisioning processes enables DevOps teams to bring changes into production quickly. Most dominantly, many DevOps teams develop continuous engineering capabilities [9] in which they automate their entire delivery process including code testing and deployment activities. This process can be further supported by automation of infrastructure provisioning [15] and configurations [12]. Furthermore, DevOps teams should develop strong monitoring and logging capabilities [6] in order to secure their systems and act quickly in case of irregularities.

4 Results

4.1 Instrument Taxonomy

As an answer to the first sub-research question, we have defined a taxonomy of the capability measurement instrument, which is composed of *dimensions*, *capabilities* and *practices*. An overview of all capabilities, definitions and practices of the instrument is shown in Table 3.

The *dimensions* of the instrument serve as broad categories which enable easy communication of the results to stakeholders. They are represented by the CALMS acronym which was coined by Humble & Molesky [11] and is widely used to address the core components of the DevOps paradigm [8]. The CALMS acronym originally represents the dimensions of *culture*, *automation*, *lean*, *measurement* and *sharing*. However, in consultation with one domain expert it was decided to replace the measurement section in our instrument with the category *monitoring*, since the requirement to measure the progress of any capability is already integrated into the capability measurement scales of our model and is thus an inherent part of every capability which is performed at level four or higher (refer to Subsect. 4.2 for a detailed explanation of the capability levels). Adding this category to the taxonomy is in line with previous research which has defined monitoring to be another integral part of DevOps [16,17].

Every instrument dimension contains a set of *capabilities* which are in turn composed of between one to three *practices*. Each practice is represented by a single question in the assessment. In order to facilitate communication and understanding of the capabilities, we added a definition to each capability which was validated by the domain experts.

4.2 Capability Measurement Scales

The second research sub-research question is based on the argument that dynamic capabilities are idiosyncratic in their details [28], which suggests that the identified DevOps team capabilities may be exhibited in distinct ways by different teams. It was therefore decided to design the instrument in such a

Table 3. Final taxonomy of the capability measurement instrument

Capability	Description	Associated practices
Culture		
Intrapreneurship	The team has processes and structures in place to ensure it fulfills all necessary business activities in order to remain relevant to customers	Opportunity scouting Experimentation Problem recognition & solving
Continuous improvement	The team has processes and structures in place to ensure a team culture focused on communication and innovation	Accepting & providing feedback Continuous improvement Sharing goals & values
Self-empowerment	The team has processes and structures in place to ensure it can function independently without intervention from management	Change readiness Decision-making Self-organization
Automation		
Continuous software engineering	The team has processes and structures in place to ensure the continuous release of high quality software	Automated build Automated testing (Continuous) Integration (Continuous) Deployment
Infrastructure & configuration management	The team has processes and structures in place to ensure the necessary infrastructure is available and configured correctly	Infrastructure provisioning & containerization Managing configurations
Artifact management	The team has processes and structures in place to ensure artifacts are stored and versioned in a repository	Use of artifacts
Architecture management	The team has processes and structures in place to ensure the architecture is and remains flexible	Use of microservices or a modular architecture
Security & access management	The team has processes and structures in place to ensure their applications are secure, in line with compliance requirements and may only be accessed by authorized users	Performing risk analysis, risk evaluation, compliance requirements & security testing Using access policies
Lean		
Lean process management	The team has a process or framework in place to ensure optimum flow of work	Lean/Agile way of working Lean/Agile project management
Change & operations management	The team has processes and structures in place to manage change requests and systems operations	Resolving incidents Automated recovery Managing changes
Continuous planning	The team has processes and structures in place to ensure a flexible planning	Planning
Customer-centric design	The team has processes and structures in place to ensure their services are targeted at involving and meeting customer needs	Stakeholder management Product-oriented team setup Cross-functional team setup
Requirement management	The team has processes and structures in place to manage and prioritize system/service requirements	Requirement specification & prioritization Use of NFRs
Monitoring		
End-user monitoring & feedback	The team has processes and structures in place to ensure it is aware of how their system is used and improve it based on end-user behaviour and feedback	Monitor customer systems & receive feedback from end-users
System monitoring & documentation	The team has processes and structures in place to monitor the performance and behavior of their internal systems	Monitoring & logging of internal systems
Sharing		
Knowledge sharing	The team has processes and structures in place to ensure that information, knowledge and skills are equally distributed and disseminated throughout the team	Information sharing Continuous learning Sharing knowledge & skills
Team collaboration	The team has processes and structures in place to ensure regular alignment between team-members and with other teams in the organization	Intra-team alignment Inter-team alignment Sharing priorities

way that it captures numerous possible configurations of a capability instead of merely assessing whether a capability is performed at a sufficient level or not. The capability measurement instrument subsequently uses a continuous representation in which the separate capabilities are assessed on five different capability levels. This is opposed to many maturity models that make use of a staged representation in which the capabilities are assigned to maturity levels.

Table 4. Measurement scales and final definitions used for capability levels per instrument dimension

Dimension	Culture, Sharing *(Definition adapted from Magdaleno CollabMM* [18]*)*	Automation, Lean, Monitoring *(Definition adapted from CMMI capability levels* [4]*)*
Level 1	**Initial*** - The capability is not performed	**Incomplete** - The capability is not performed
Level 2	**Ad-hoc** - The team decides on the spot how to carry out collaboration activities	**Performed -** The capability is carried out and works in practice. However, the team has no agreed way of working for doing this
Level 3	**Planned** - The team has an agreed way of working in which it collaborates, e.g. through regular meetings or platforms	**Managed** - The team has agreed on a specific way of working such as a process or policy to ensure that the capability is performed
Level 4	**Aware** - Team members are aware of their tasks and of the agreed process, no central coordination is necessary for the members to collaborate. The team might include monitoring activities to ensure the way of working is leading to the desired capability	**Defined** - The team has worked out their way of working in detail, e.g. through process descriptions, monitoring performance measurement metrics or adapting the process from the organizational policy to their own needs
Level 5	**Reflexive** - Team members are aware of the agreed way of working and are self-organizing. They can identify which results are relevant and are continuously collaborating, interacting and sharing knowledge among each other. They use double-loop learning to recognize whether the desired goal state is still applicable.	**Optimizing*** - The team does not only have an elaborate way of working but also continuously reflects on the process and improves this to perform the capability even better

*Levels added by researchers to equalize scales.

Given the diverging nature of capabilities in the relationship-oriented dimensions of culture and sharing and the more traditional, process-oriented dimensions of automation, lean and monitoring, it was decided to use two different, yet comparable measurement scales to define the capability levels in our instrument.

The answer options to questions related to the culture and sharing dimensions were adapted from the Collaboration Maturity Model (CollabMM) by Magdaleno, Araujo and Werner [18]. This scale was chosen due to its explicit focus on team collaboration, as opposed to the more process-oriented focus of many other models. Although the CollabMM scale is originally used in a staged representation, we found the scale to also be useful for assessing the separate capabilities and have developed descriptions which suit this aim.

The capability levels of the dimensions automation, lean and monitoring were adapted from the CMMI continuous representation capability levels [4]. This measurement scale was chosen due to its wide recognition and use in both academia and practice, as well as the continuous nature of the scale.

In order to equalize the scales, we added a capability level to the lower end of the CollabMM and to the upper end of the CMMI capability level descriptions. The descriptions of each capability level were validated and adjusted based on feedback given by the domain experts. The final definitions can be found in Table 4.

4.3 Assessment Items

The practices and capability levels which we previously discussed were translated to fitting questions and answer options and were supplemented with industry examples with the help of a domain expert during the item identification stage. The final version of the instrument contains 38 assessment items which represent the practices in Table 3. Two example questions and answer options are displayed in Table 5.

5 Discussion and Conclusion

The research at hand describes the design and validation of a capability measurement instrument for DevOps teams. To arrive at this artifact, we have investigated the sub-research questions *“Which capabilities and practices are relevant to DevOps teams?”* and *“How to assess varying configurations of capabilities with a measurement instrument?”*. As an answer to these questions, we offer a comprehensive taxonomy of DevOps capabilities and practices and describe two measurement scales on which the varying configurations of a capability can be measured. Due to the taxonomy being based on the results of a SLR, the capabilities and practices in our measurement instrument are supported by existing literature on DevOps capabilities [17,25,26] but extend the aforementioned works. The resulting instrument was developed and validated in close collaboration with industry practitioners, using qualitative research approaches from PAR as well by collecting data via surveys. The results of the validation phase

Table 5. Measurement instrument

Dimension:	Culture	*Capability*:	Intrapreneurship
Practice:	Experimentation	*Measurement Scale*:	CollabMM [18]
Does your team experiment with new ideas or techniques regarding your way of working or product/service?			
Level 1:	We never experiment with new techniques. We rather stick to what we know and what works for us		
Level 2:	We sometimes experiment with new ideas but not in a coordinated way		
Level 3:	Experimentation is a planned and coordinated part of our work, e.g. we free up time during our sprints to try new things		
Level 4:	We regularly experiment with new techniques to improve our product and way of working as part of our daily work. This happens inside and outside of planned events		
Level 5:	We regularly experiment with new techniques to improve our product and way of working, inside and outside of planned events. These insights often lead to improvements in our product or way or working		
Dimension:	Lean	*Capability*:	Change & operations mngmt.
Practice:	Resolving incidents	*Measurement Scale*:	CMMI capability levels [4]
How do you deal with incidents?			
Level 1:	We do not have a procedure for this. We deal with incidents when they arise		
Level 2:	When an incidents arises we decide on a case-to-case basis based on our own judgement if we deal with it directly or later		
Level 3:	We have a standardized procedure for classifying and dealing with incidents, e.g. based on ITIL		
Level 4:	Dealing with incidents is part of our way of working, e.g. incidents are prioritized and placed on the backlog or we have reserved time every day to deal with important incidents		
Level 5:	Dealing with incidents is part of our way of working, e.g. incidents are prioritized and placed on the backlog or we have reserved time every day to deal with important incidents. We regularly reflect on our incident handling process and improve it, e.g. by performing a blameless post-mortem analysis		

indicate clear agreement of the experts and the DevOps team members with all aspects of the measurement instrument, resulting in high mean agreement scores as shown in Table 2.

Nevertheless, participants had varying opinions regarding the appropriateness of the length of the instrument and the associated number of questions which resulted in a high standard deviation of validation item number 14 (Table 2). When asked about the amount of time it took them to complete the survey, participants reported values between 10 and 30 min. Furthermore, the domain experts disagreed on the sufficiency of the five capability levels to represent all possible states of a team capability. Three respondents strongly agreed (score of 5) with this statement whereas one respondent disagreed (score of 2). One of the interviewed domain experts pointed out that a five-point scale is the industry standard on which many assessments and maturity models are based and that the scale should therefore be kept this way.

During the interview phase, multiple domain experts pointed out that they would like to include behavioural or intangible aspects such as trust and respect between the team members in the assessment. This is supported by the results of our literature review which has revealed the above mentioned factors to be essential to the performance of DevOps teams [26]. However, while we find these traits to be invaluable for DevOps teams, they did not fit our definition of a capability and could not be measured using one of our proposed measurement scales. We have therefore decided to not include these aspects in the assessment.

The proposed measurement instrument is designed to be used as a self-assessment. This is different to traditional capability maturity models, in which the researcher is often required to evaluate the organization in question based on pre-defined guidelines and templates [23]. One of the interviewed domain experts pointed out that a strong aspect of the proposed type of self-assessment is its ability to measure the capabilities over a large amount of teams. Furthermore, the standardized measurement instrument may help to compare the capabilities of different teams. However, the same interviewee indicated their preference for a more qualitative, in-depth approach when dealing with a smaller sample size of teams. This approach ensures that the neutral opinion and observations of the assessor are taken into account when conducting the assessment whereas our proposed approach is entirely dependent on the judgement of the team members using the measurement instrument.

5.1 Contributions to Theory and Practice

The research at hand provides novel contributions to both theory and practice. On the practical side, we contribute a tool that may be used by IT professionals to measure the capability configuration of DevOps teams. The results of the measurement provide valuable information into the status of the transformation process of DevOps teams and offer directions for further improving their team performance. The tool may also contribute to fostering a shared understanding of a DevOps definition and associated capabilities.

On the theory side, we provide insights into the nature of DevOps capabilities, the different configurations which they may take on as well as propose suitable scales to measure their maturity. Different to extant models and research on DevOps capabilities, our measurement instrument accounts for the idiosyncrasy of capabilities. Present DevOps maturity models are primarily focused on mapping capabilities to maturity levels [34] but did not investigate the potential ways in which a capability may be implemented. We therefore adopted a continuous representation in which we measure the configuration of DevOps capabilities in themselves on a five-level scale, but do not imply any hierarchy of capabilities or succession regarding their implementation as it would be the case in a staged representation maturity model.

5.2 Limitations and Further Research

Our research and the accompanying DevOps team capability assessment are limited by a number of factors. Primarily, our research was predominantly based on qualitative research approaches which was done to support the design of theory behind the instrument. No statistical methods were used to judge the validity and internal consistency of the categories. Future research should therefore further validate and improve our taxonomy by using techniques such as factor analysis or Cronbach's alpha. Collecting a larger number of responses on the survey would also support an in-depth psychometric analysis. Furthermore, our research solely focuses on the implementation and configuration of capabilities, to be understood in terms of underlying processes and structures. Behavioural and intangible aspects such as trust or respect were therefore excluded from our model and warrant further investigation in terms of how to measure and include these in a measurement instrument.

5.3 Conclusion

The research at hand proposes a capability measurement instrument for DevOps teams. Based on a systematic literature review and in close collaboration with industry practitioners, we developed a taxonomy which encompasses seventeen capabilities and thirty-eight associated practices that are measured on five capability levels. The resulting instrument and its taxonomy provide insights into the nature and configuration of DevOps capabilities as well as a standardized approach to measuring these and improving DevOps team performance.

References

1. Aldea, A., Sarkar, A.: A measurement instrument for enterprise architecture resilience research: a pilot study on digital transformation. In: Proceedings of the 55th Hawaii International Conference on System Sciences, pp. 7182–7191 (1 2022)
2. Brydon-Miller, M., Greenwood, D., Maguire, P.: Why action research? Action Res. **1**(1), 9–28 (7 2003). https://doi.org/10.1177/14767503030011002
3. Caretta, M.A., Vacchelli, E.: Re-thinking the boundaries of the focus group: a reflexive analysis on the use and legitimacy of group methodologies in qualitative research. Sociolog. Res. Online **20**(4), 58–70 (2015). https://doi.org/10.5153/sro.3812
4. Carnegie Mellon University Software Engineering Institute: CMMI for Development, Version 1.3. Technical report, November 2010
5. Eisenhardt, K.M., Martin, J.A.: Dynamic capabilities: what are they? Strateg. Manag. J. **21**(10/11), 1105–1121 (2000). http://www.jstor.org/stable/3094429
6. Erich, F.M.A., Amrit, C., Daneva, M.: A qualitative study of DevOps usage in practice. J. Softw. Evol. Process **29**(6) (2017). https://doi.org/10.1002/smr.1885

7. de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., Brinkkemper, S.: DevOps competences and maturity for software producing organizations. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2018. LNBIP, vol. 318, pp. 244–259. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_16
8. Fitzgerald, B., Stol, K.J.: Continuous software engineering: a roadmap and agenda. J. Syst. Softw. **123**, 176–189 (2017)
9. Gruhn, V., Schäfer, C.: BizDevOps: because DevOps is not the end of the story. In: Fujita, H., Guizzi, G. (eds.) SoMeT 2015. CCIS, vol. 532, pp. 388–398. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22689-7_30
10. Hemon, A., Fitzgerald, B., Lyonnet, B., Rowe, F.: Innovative practices for knowledge sharing in large-scale DevOps. IEEE Softw. **37**(3), 30–37 (2020)
11. Humble, J., Molesky, J.: Why enterprises must adopt DevOps to enable continuous delivery. Cutter IT J. **24**(8), 6–12 (2011)
12. Hussain, W., Clear, T., MacDonell, S.: Emerging trends for global DevOps: a New Zealand perspective. In: Proceedings - 2017 IEEE 12th International Conference on Global Software Engineering, ICGSE 2017, pp. 21–30. Software Engineering Research Lab (SERL), School of Engineering, Computer and Mathematical Sciences (SECMS), Auckland University of Technology (AUT), Auckland, New Zealand (2017). https://doi.org/10.1109/ICGSE.2017.16
13. Hussaini, S.W.: Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 178–183 (2014). https://doi.org/10.1109/ITSC.2014.6957687
14. Iacob, M.E., Quartel, D., Jonkers, H.: Capturing business strategy and value in enterprise architecture to support portfolio valuation. In: Proceedings of the 2012 IEEE 16th International Enterprise Distributed Object Computing Conference, EDOC 2012, pp. 11–20 (2012). https://doi.org/10.1109/EDOC.2012.12
15. Luz, W.P., Pinto, G., Bonifácio, R.: Adopting DevOps in the real world: a theory, a model, and a case study. J. Syst. Softw. **157** (2019). https://doi.org/10.1016/j.jss.2019.07.083
16. Lwakatare, L.E., Kuvaja, P., Oivo, M.: Dimensions of DevOps. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 212–217. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_19
17. Lwakatare, L.E., Kuvaja, P., Oivo, M.: An exploratory study of DevOps: extending the dimensions of DevOps with practices. In: 11th International Conference on Software Engineering Advances, ICSEA 2016, pp. 91–99, Rome, Italy (2016)
18. Magdaleno, A.M., Araujo, R.M.D., Werner, C.M.L.: A roadmap to the Collaboration Maturity Model (CollabMM) evolution. In: Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2011, pp. 105–112 (2011). https://doi.org/10.1109/CSCWD.2011.5960062
19. Myers, M.D., Newman, M.: The qualitative interview in IS research: examining the craft. Inf. Organ. **17**(1), 2–26 (2007). https://doi.org/10.1016/j.infoandorg.2006.11.001
20. Nagarajan, A.D., Overbeek, S.J.: A DevOps implementation framework for large agile-based financial organizations. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) OTM 2018. LNCS, vol. 11229, pp. 172–188. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_10

21. Plant, O.H., van Hillegersberg, J., Aldea, A.: How DevOps capabilities leverage firm competitive advantage: a systematic review of empirical evidence. In: 2021 IEEE 23rd Conference on Business Informatics (CBI) 2021, pp. 141–150. Institute of Electrical and Electronics Engineers (IEEE) (2021). https://doi.org/10.1109/cbi52690.2021.00025
22. Ravichandran, T., Lertwongsatien, C.: Effect of information systems resources and capabilities on firm performance: a resource-based perspective. J. Manag. Inf. Syst. **21**(4), 237–276 (2005). https://doi.org/10.1080/07421222.2005.11045820
23. Salah, D., Paige, R., Cairns, P.: An evaluation template for expert review of maturity models. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (eds.) PROFES 2014. LNCS, vol. 8892, pp. 318–321. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13835-0_31
24. Santini, C., Marinelli, E., Boden, M., Cavicchi, A., Haegeman, K.: Reducing the distance between thinkers and doers in the entrepreneurial discovery process: an exploratory study. J. Bus. Res. **69**(5), 1840–1844 (2016). https://doi.org/10.1016/j.jbusres.2015.10.066
25. Senapathi, M., Buchan, J., Osman, H.: DevOps capabilities, practices, and challenges: insights from a case study. In: ACM International Conference Proceeding Series, EASE 2018, vol. Part F1377, pp. 57–67. ACM, New York (2018)
26. Smeds, J., Nybom, K., Porres, I.: DevOps: a definition and perceived adoption impediments. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 166–177. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_14
27. Teece, D.J.: Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. Strateg. Manag. J. **28**(13), 1319–1350 (2007)
28. Teece, D.J., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. Strateg. Manag. J. **18**(7), 509–533 (1997)
29. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. Eur. J. Inf. Syst. **25**(1), 77–89 (2016). https://doi.org/10.1057/ejis.2014.36
30. Vishnubhotla, S.D., Mendes, E., Lundberg, L.: Understanding the perceived relevance of capability measures: a survey of agile software development practitioners. J. Syst. Softw. **180**, 111013 (2021). https://doi.org/10.1016/j.jss.2021.111013
31. Wiedemann, A., Schulz, T.: Key capabilities of DevOps teams and their influence on software process innovation: a resource-based view. In: Proceedings of the 23rd Americas Conference on Information Systems, AMCIS 2017. Neu-Ulm University of Applied Sciences, Germany (2017)
32. Wiedemann, A., Wiesche, M., Gewald, H., Krcmar, H.: Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment. Eur. J. Inf. Syst., 1–16 (2020). https://doi.org/10.1080/0960085X.2020.1782277

33. Wolfswinkel, J.F., Furtmueller, E., Wilderom, C.P.: Using grounded theory as a method for rigorously reviewing literature (2013). https://doi.org/10.1057/ejis.2011.51

34. Zarour, M., Alhammad, N., Alenezi, M., Alsarayrah, K.: A research on DevOps maturity models. Int. J. Recent Technol. Eng. **8**(3), 4854–4862 (2019). https://doi.org/10.35940/ijrte.C6888.098319