

Achieving Product Orientation in DevOps Teams

Presentation at COPPE UFRJ



Building Competence. Crossing Borders.

Dr. Anna Wiedemann

Anna.wiedemann@zhaw.ch Institute for Business Information Systems

Introduction



Agile software development methods have become very popular and provide effective methods to quickly respond to changing customer needs.

However, these agile methods are mainly used by software developers in the IT function.

Organizations want better collaboration between development and operations departments so that bugs are identified faster and resolved early.

Attempting to achieve these opposing functional goals (speed vs. stability) in an internal cross-functional team is likely to create tensions (Onita & Dhaliwal, 2011; Shaft & Vessey, 2006).

Research and practice do not yet fully understand how and why exactly these tensions occur and lack guidance on how to effectively navigate them.

Many scholars have applied control theory to explain the process of managing project teams (Kirsch, 1997; Maruping et al., 2009).

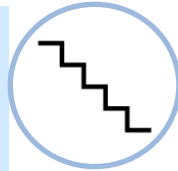
The focus on control in software development projects has led to valuable contributions to research and practice (Wiener et al., 2016).



New Paper: Integrating development and operations teams: A control approach for DevOps
<https://www.sciencedirect.com/science/article/pii/S1471772723000283>

Project Management and DevOps

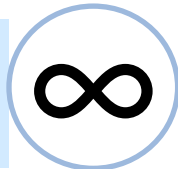
Sequential Procedure



Agile Manifest



DevOps

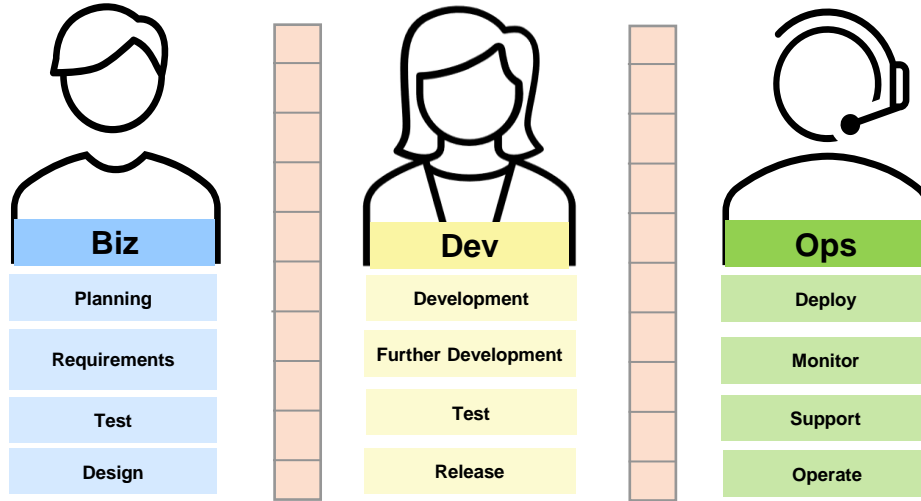


(Hemon et al. 2018) (Beck et al. 2001) (Kim et al. 2016)

DevOps

DevOps is a cultural and technological concept for integrating tasks and skills relating to software product planning, development and operation in cross-functional teams (Wiedemann et al. 2020).

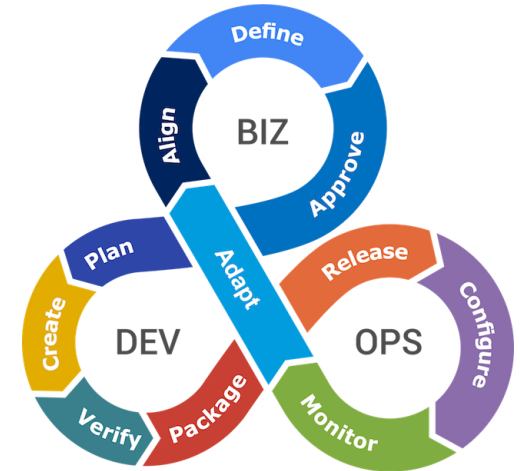
Benefits of DevOps



Agile Software Development

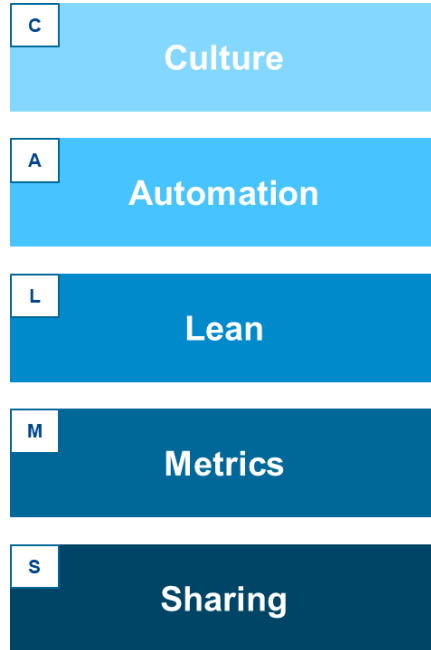
DevOps

BizDevOps – End2End Value Stream



<https://stackify.com/bizdevops-guide/>

DevOps Key Principles



Source: <https://blog.xebia.fr/2017/04/21/introduction-a-devops/>

DevOps Principles

C

Culture

Focus on people:

- Mutual trust
- Willingness to learn and continuously improve
- Constant flow of information
- Openness to change and experimentation

A

Automation

Mapping of simple, recurring processes and/or full automation of entire environments:

- Continuous Integration and Continuous Delivery
- Save time, avoid problems, create consistency and enable self-services

L

Lean

Application of lean principles:

- Minimization of workloads
- Shorten and strengthen feedback loops
- Identify and minimize inefficiencies in processes

M

Metrics

Monitoring of the complete system incl. all components and the underlying (automated) processes:

- Create uniform evaluation criteria
- Measure everything continuously and show improvements
- Basis for continuous and sustainable improvements

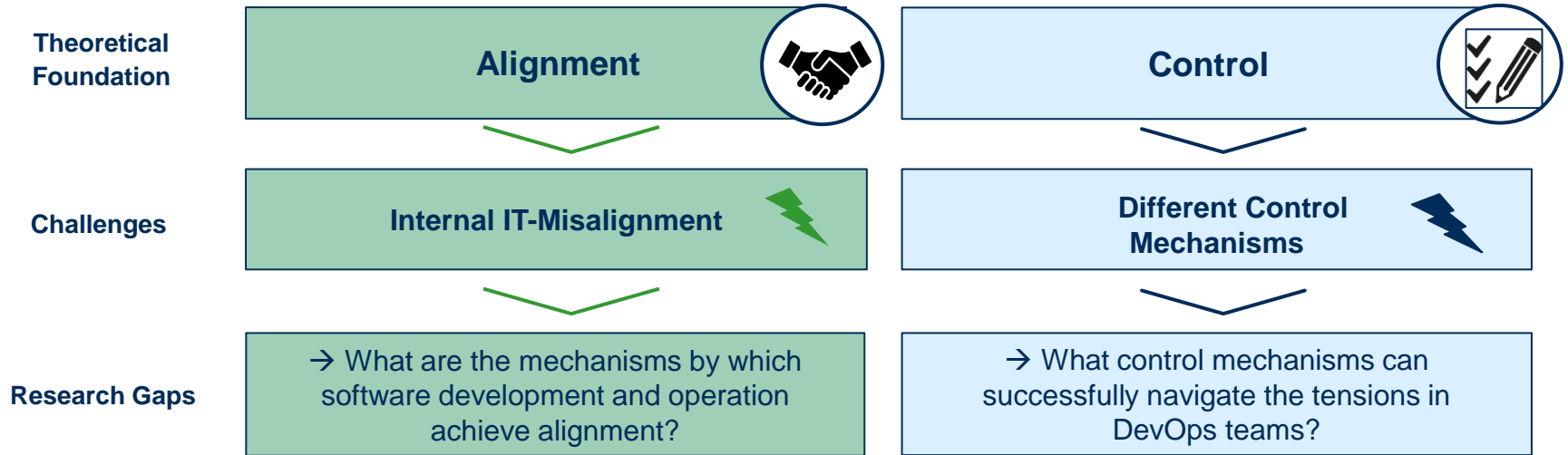
S

Sharing

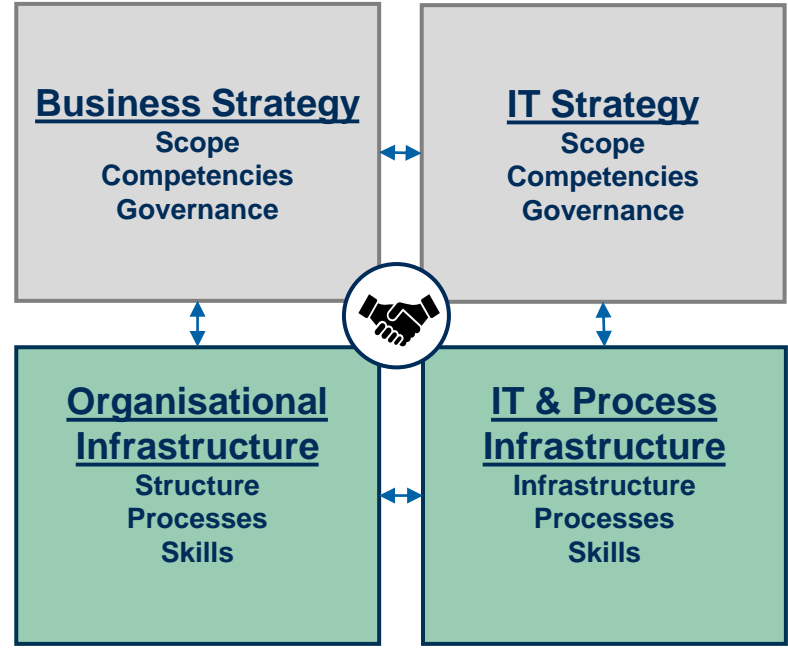
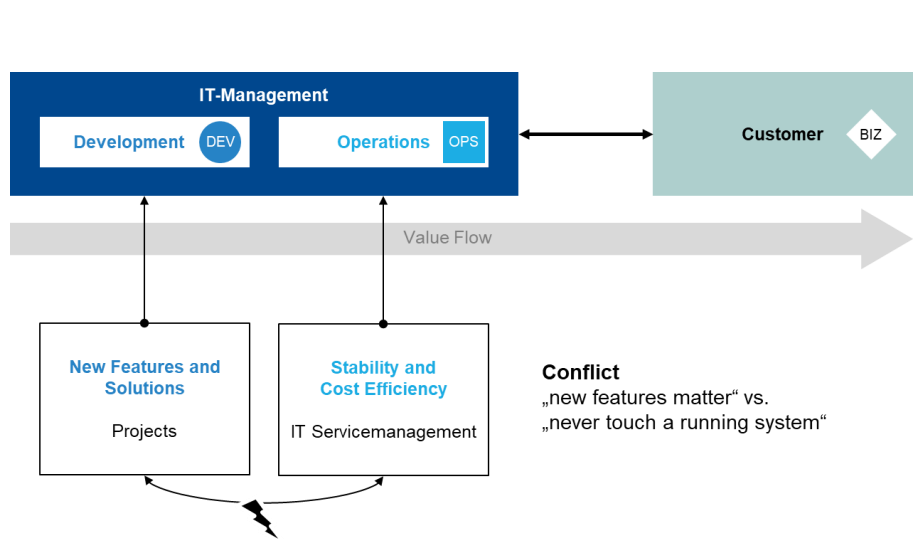
"Sharing is Caring" readiness within the organization and across organizational boundaries:

Sharing knowledge, tools, and source code, learning from each other, and proactively sharing insights.

DevOps Research

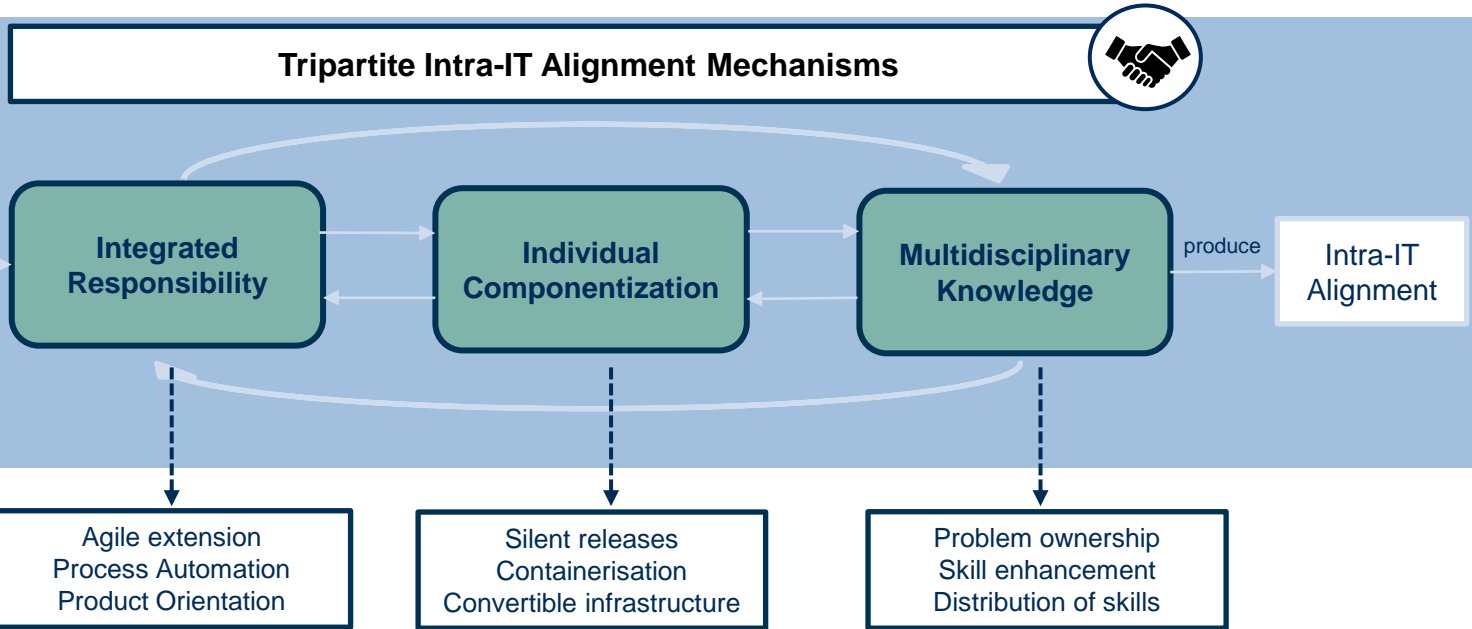


Business-IT Alignment – Some Conflicts



Simplified representation from J. C. Henderson and N. Venkatraman (1993)

Tripartite Intra-IT Alignment Mechanisms



Own Depiction

Grounded theory model with three mechanisms for achieving intra-IT alignment in DevOps teams.

Information Systems Control Literature

In software development literature control is defined as management's ***“attempts to ensure that individuals working on organizational projects act according to an agreed-upon strategy to achieve desired objectives”*** (Kirsch, 1996, p. 1).



Control effects

focus on IS project performance as a consequence of control-mode choices measured in terms of quality and efficiency (Barki et al. 2001; Wiener et al. 2016).



Control choices

refer to the antecedents of the specific projects' control portfolio configuration with control modes and control amounts (Wiener et al. 2016).



Control context

refers to the project size, project tasks, etc. (Kirsch 1996; Kirsch 1997), **and the stakeholder context refers to the characteristics of the controller and controllee and their relationships** (Kirsch 2004).



Control dynamics

concentrate on the style according to which control modes and amounts are exercised in practice (Choudhury and Sabherwal 2003; Kirsch 2004; Remus and Wiener 2012).

Differences Between Development and Operations by IS Control (excerpt)

Theme	Development	Operations
Control effects	<ul style="list-style-type: none"> High efficiency and quality project outcome (Barki et al. 2001; Wiener et al. 2016) measured by project completeness in terms of time, budget (Keil et al. 2013), and meeting user requirements (Kirsch 1996) Provision of rapid software features (Fitzgerald and Stol 2017) 	<ul style="list-style-type: none"> Measurement of Service Level Agreements (SLAs) via ticket management tools (e.g., ticket resolution time) (Trusson et al. 2014). Aiming at the stability of running software due to the reduction of production releases (Kim and Westin 1988)
Control choices	<ul style="list-style-type: none"> Combination of control modes (formal: input, behavior, and output control; and informal: clan and self-control) for project control portfolios (Kirsch 1997). Control degree and influence on control style choices (Heumann et al. 2015; Kirsch 1996). 	<ul style="list-style-type: none"> Application and measurement of standardized operations processes (ITIL) (Pollard and Cater-Steel 2009; Trusson et al. 2014) and corresponding governance frameworks (COBIT) (April et al. 2005) Avoidance of software outages due to monitoring (Nelson et al. 2000; Shaft and Vessey 2006).
Control contexts	<ul style="list-style-type: none"> Project and stakeholder context (Kirsch 1996; Kirsch 1997; Wiener et al. 2016). Collaboration in teams using agile project management or waterfall methods (Cram et al. 2016b). 	<ul style="list-style-type: none"> End-user and stakeholder-oriented (Nelson et al. 2000). Usage of IT service management approaches as a management tool (Trusson et al. 2014).
Control dynamics	<ul style="list-style-type: none"> Rapid changes across project lifecycle phases (Wiener et al. 2016). Changes in control modes, control amounts (Choudhury and Sabherwal 2003; Kirsch 2004), and control styles (Choudhury and Sabherwal 2003; Gregory et al. 2013; Kirsch 2004) 	<ul style="list-style-type: none"> Adoption of new or modified demands in terms of technology and platform changes (Edberg et al. 2012; Nelson et al. 2000). Seeking quality and stability of software (Hemon-Hildgen et al. 2020)

Tensions Resulting from Differences in Control Themes

A tension is defined as **“stress, anxiety, discomfort, or tightness in making choices, responding to, and moving forward in organizational situations”** (Putnam et al. 2016, p. 67).

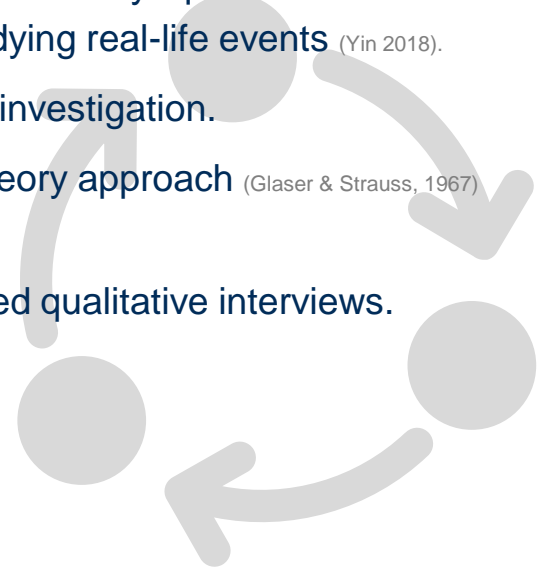
- Based on the extant literature on the different approaches of software development and operations, we derived four control tensions: method discomfort, goal conflict, decision rights, and time rhythm.
- When a company decides to combine both functions into DevOps, the tensions need to be assessed and managed—otherwise, the new structure will not be successful.
- Some companies have successfully managed the transition toward a combined DevOps model and have had success in managing the tensions arising within these departments.

Tensions in Software Development and Operations Teams (excerpt)

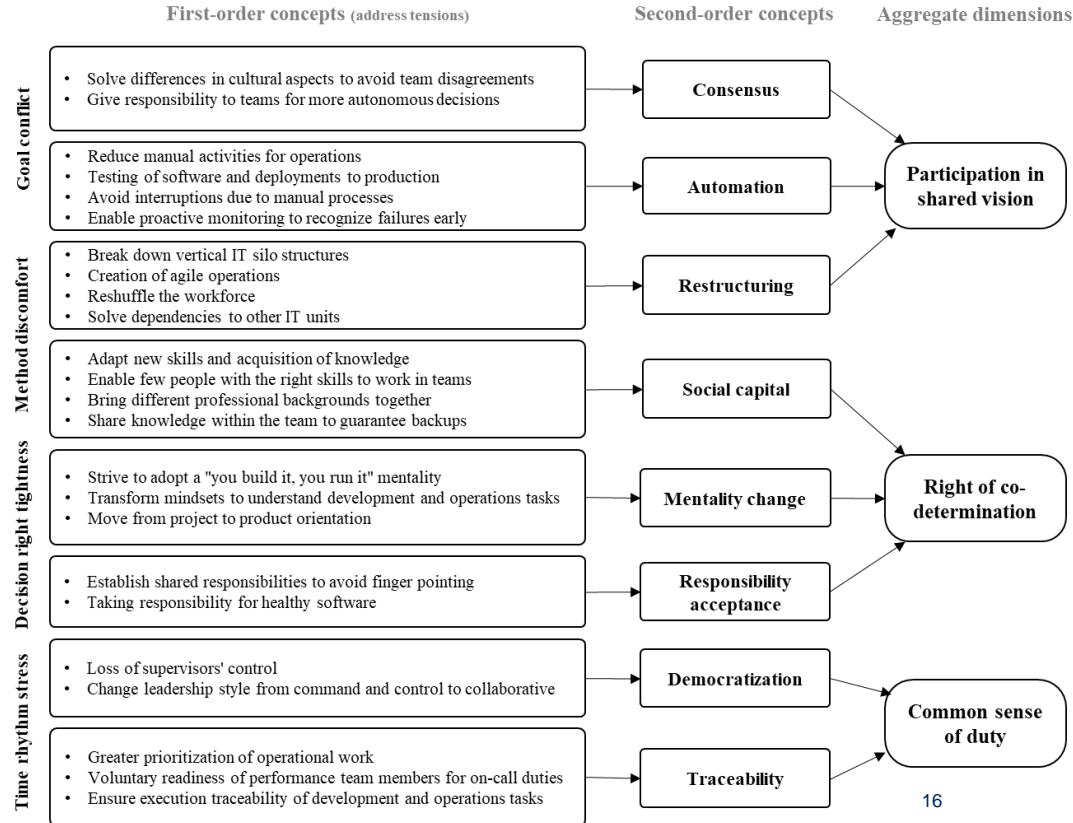
Source of Tension	How Tensions Manifest	Literature
Tension 1: Goal conflict		
Speed vs. Stability	High numbers of software changes inherently jeopardize stable production, as every change carries the risk of a system failure. Operations favor few updates and additions, whereas development seeks the opposite.	Edberg et al. (2012); Fitzgerald and Stol (2017); Keil et al. (2013); Maruping et al. (2009); Trusson et al. (2014)
Tensions 2: Method discomfort		
Agility vs. Structure	The different methodological approaches are a breeding ground for tension. While development heralds a “learn from mistakes” and “fail fast, fail often” culture, operations maintain a “failure is not an option” mindset.	Tarafdar and Tanriverdi (2018); Fitzgerald and Stol (2017); Hemon-Hildgen et al. (2020); Krancher et al. (2018); Vääätäjä et al. (2016)
Tension 3: Decision rights		
Project vs. Service	The two management styles differ and the responsible managers are measured by metrics that are not aligned with each other, tensions arise, as both parties aim to achieve their aims for their work stream.	Fitzgerald and Stol (2017); Kirsch et al. (2002); Shaft and Vessey (2006); Wiener et al. (2016) Edberg et al. (2012)
Tension 4: Time rhythm		
Short Term vs. Long Term	Development is used to deliver projects quickly, whereas operations seek to provide long-term stability—leading to distinct tensions. Development might describe operations as “too slow,” whereas operation’s reply would be “too hectic, not well thought through.”	Edberg et al. (2012); Kirsch (2004); Trusson et al. (2014)

Research Method

- Qualitative research methods are well-suited to analyzing novel phenomena and deriving compelling explanations.
- Case studies are recommended for conducting research to answer “how” or “why” questions about contemporary issues such as DevOps teams and are well-suited for studying real-life events (Yin 2018).
- We collected primary and secondary qualitative data in an in-depth field investigation.
- The analysis of the transcripts was guided by the Glaserian grounded theory approach (Glaser & Strauss, 1967) and the inductive analytical approach presented by Gioia et al. (2013).
- The primary data consisted of a case study comprising 21 semi-structured qualitative interviews.



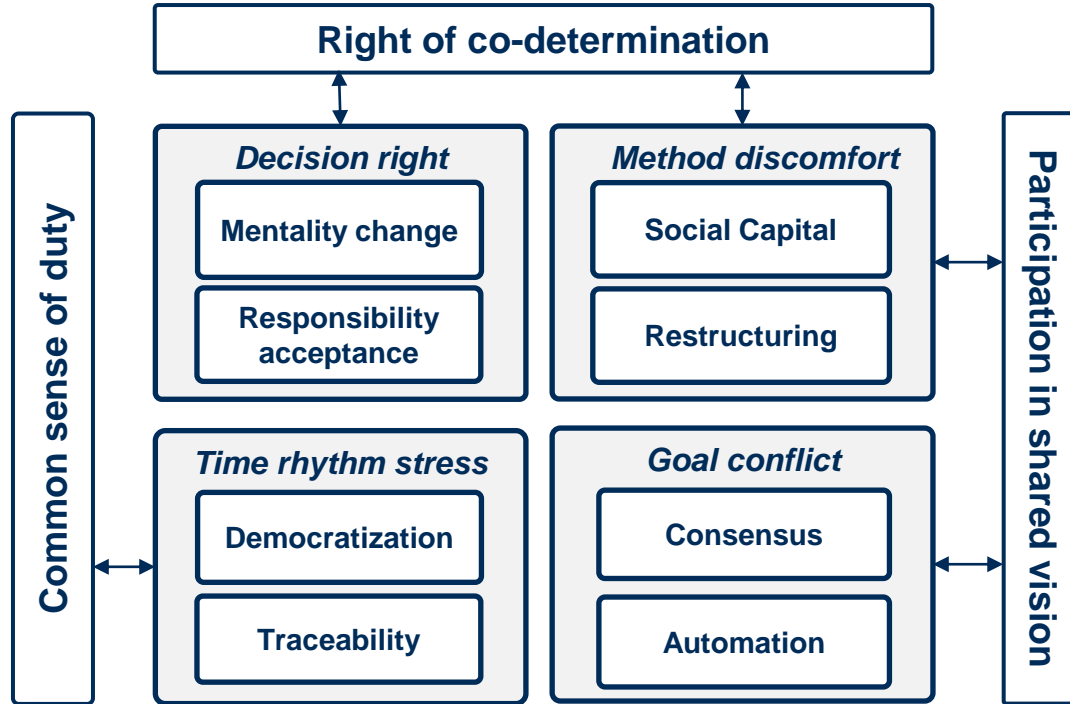
Data Structure



Key Dimensions and Mechanisms that Mitigate Tensions in DevOps

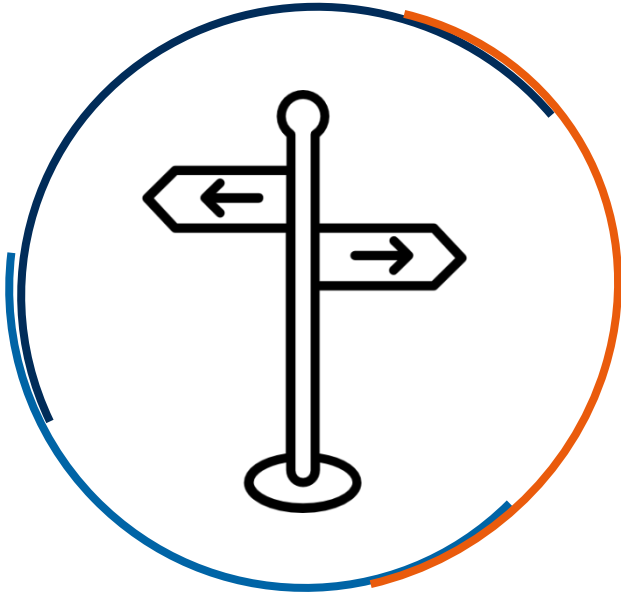
Dimensions	Description	Associated Tensions	Key mechanisms and explanations
Participation in shared vision	DevOps teams develop and enact a common view for a collaborative-working style.	Goal conflict	Consensus: DevOps teams assume responsibility for the product and work toward a common goal through coaching.
		Method discomfort	Automation: Resolving manual activities and automating DevOps processes such as testing and deploying updates.
Right of co-determination	DevOps teams determine how to assume responsibility and navigate joint decision rights for the software delivery lifecycle.	Method discomfort	Restructuring: DevOps teams are set up and integrated with established IT functions. Social capital: Depending on the product, DevOps teams are responsible for the necessary development and operations skills that must be learned within the team.
		Decision rights	Mentality change: DevOps team members work in teams with end-to-end responsibility for the software delivery lifecycle. Accepting responsibility: DevOps team members with different backgrounds are responsible for software quality and for managing all software-relevant problems.
Common sense of duty	DevOps teams achieve ownership through the adaption of the leadership style and transparency for all activities.	Time rhythm stress	Democratization: A DevOps leadership style that moves from command-and-control to a more collaborative control approach. Traceability: The DevOps team's understanding and assimilation of activities from both development and operations.

Mitigation of Tensions Through Control Mechanisms



Own Depiction

Further Research...



examine suitable measures of success and failure for DevOps teams and address the influence of control characteristics.

focus on the potential role of inter-organizational control and its effect on control (outsourcing).

investigate which products are suitable for DevOps.

validate with quantitative methods.

Possible Research Collaboration Topics

1. **DataOps/ AIOps Case Study**
2. **AI applications in cross-functional Teams (DevOps)**
3. **Autonomous product teams (e.g. DevOps) and the application of low-code and no-code platforms**
4. **The role of the individual IT identity in self-organizing product teams (DevOps)**
5. **A practice guide for DevOps implementation – A step-by-step introduction using a company example**



Thank you.

Email: Anna.wiedemann@zhaw.ch

[ZHAW Website](#)

[LinkedIn](#)



References

- Beck, K., Beedle, M., Bennekum, v. A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifesto for agile software development. In. <http://agilemanifesto.org>
- Bass, L. 2018. "The Software Architect and DevOps," IEEE Software (35:1), pp. 8-10.
- Conboy, K. 2009. "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development," Information Systems Research (20:3), pp. 329-354.
- Computerwoche: <https://www.computerwoche.de/a/was-ist-dataops.3553890>
- Fischbach, M., Puschmann, T., and Alt, R. 2013. "Service Lifecycle Management," Business & Information Systems Engineering (5:1), pp. 45-49.
- Hemon, A., et al. (2018). Job Satisfaction Factors and Risks Perception: An Embedded Case Study of DevOps and Agile Teams. International Conference on Information Systems. San Francisco.
- Kim et al. 2016, The DevOps Handbook. IT Revolution Press
- Krancher, O., Luther, P., and Jost, M. 2018. "Key Affordances of Platform-as-a-Service: Self-Organization and Continuous Feedback," Journal of Management Information Systems (35:3), pp. 776-812.
- Forsgren, N., et al. (2019). State of DevOps Report, Google Cloud.
- Lewis, M. O., Mathiassen, L., and Rai, A. 2011. "Scalable Growth in It-Enabled Service Provisioning: A Sensemaking Perspective," European Journal of Information Systems (20:3), pp. 285-302.
- Medienmitteilung Nextthink: <https://www.personal-schweiz.ch/aktuelles/article/nextthink-studie-mitarbeiter-verlieren-durch-it-ausfaelle-jaehrlich-zwei-wochen-an-arbeitszeit/>
- Peters, C., Farley, D., Villalba, D., Stanke, D., DeBellis, D., Maxwell, E., Meyxer, J. S., Xu, K., Harvey, N., and Kulesza, T. 2022. "Accelerate State of Devops 2022." Retrieved 07-12-2022, 2022, from <https://cloud.google.com/devops/state-of-devops>
- VersionOne. 2022. "The 16th Annual State of Agile Report." Retrieved 07-12-2022, 2022, from <https://stateofagile.com/>
- Wiedemann, A., Wiesche, M., Gewalt, H., & Krcmar, H. (2020). Understanding How DevOps Aligns Development and Operations: A Tripartite Model of Intra-IT Alignment, <https://www.tandfonline.com/doi/full/10.1080/0960085X.2020.1782277>
- Slides 1, 10-18 are based on the paper: <https://www.sciencedirect.com/science/article/pii/S1471772723000283> References can be found there.