

Empirical Studies to Identify Coordination- and Methodology Patterns in Large-Scale Agile Development

Moritz Schuell, 05.08.2019, Bachelor's Thesis Final Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
www.matthes.in.tum.de

Motivation

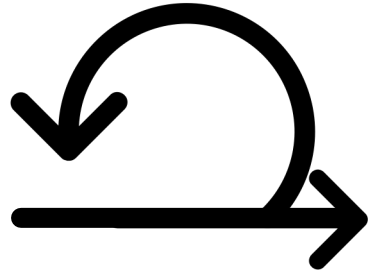
Approach

Case Description

Results

Exemplary Pattern Candidate

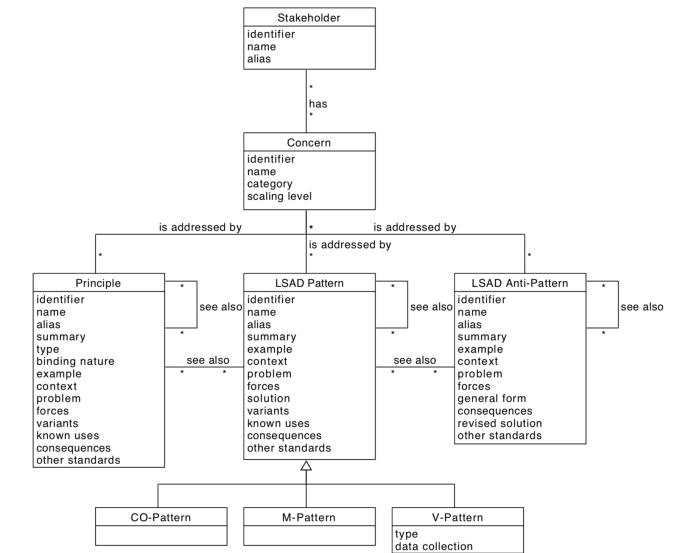
Conclusion



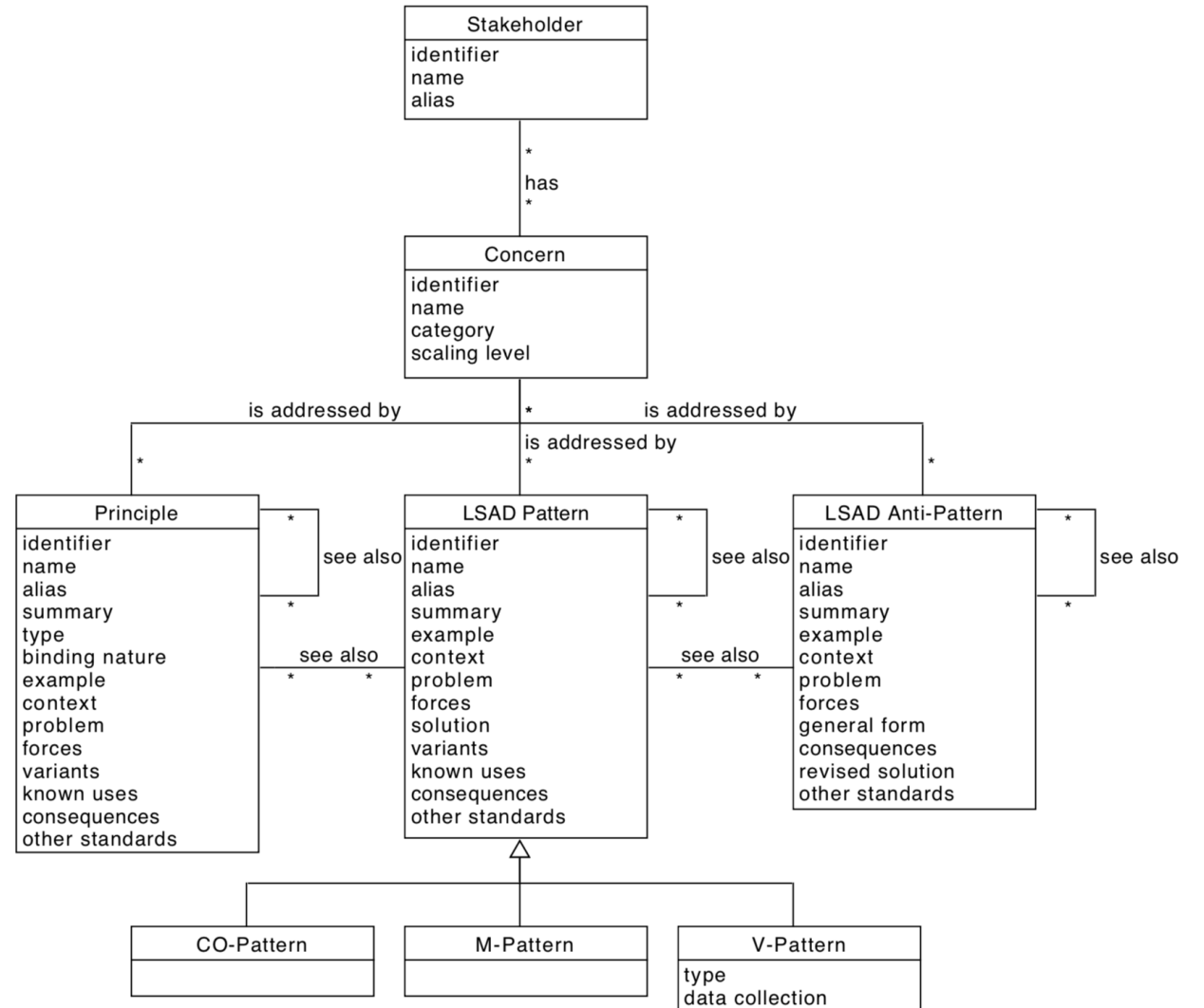
Agile methods are designed for small, co-located teams.

Their shown benefits make them attractive to **larger companies** as well [1].

Applying agile methods on **large-scale projects** leads to several **concerns** [2].



Large-scale agile development **pattern language** created by sebis [3].



RQ 1

What are recurring coordination and methodology concerns in large-scale agile development?

RQ 2

What are good practices for addressing recurring coordination and methodology concerns in large-scale agile development?

RQ 3

Which anti-patterns regarding coordination and methodologies should be avoided in large-scale agile development?

Motivation

Approach

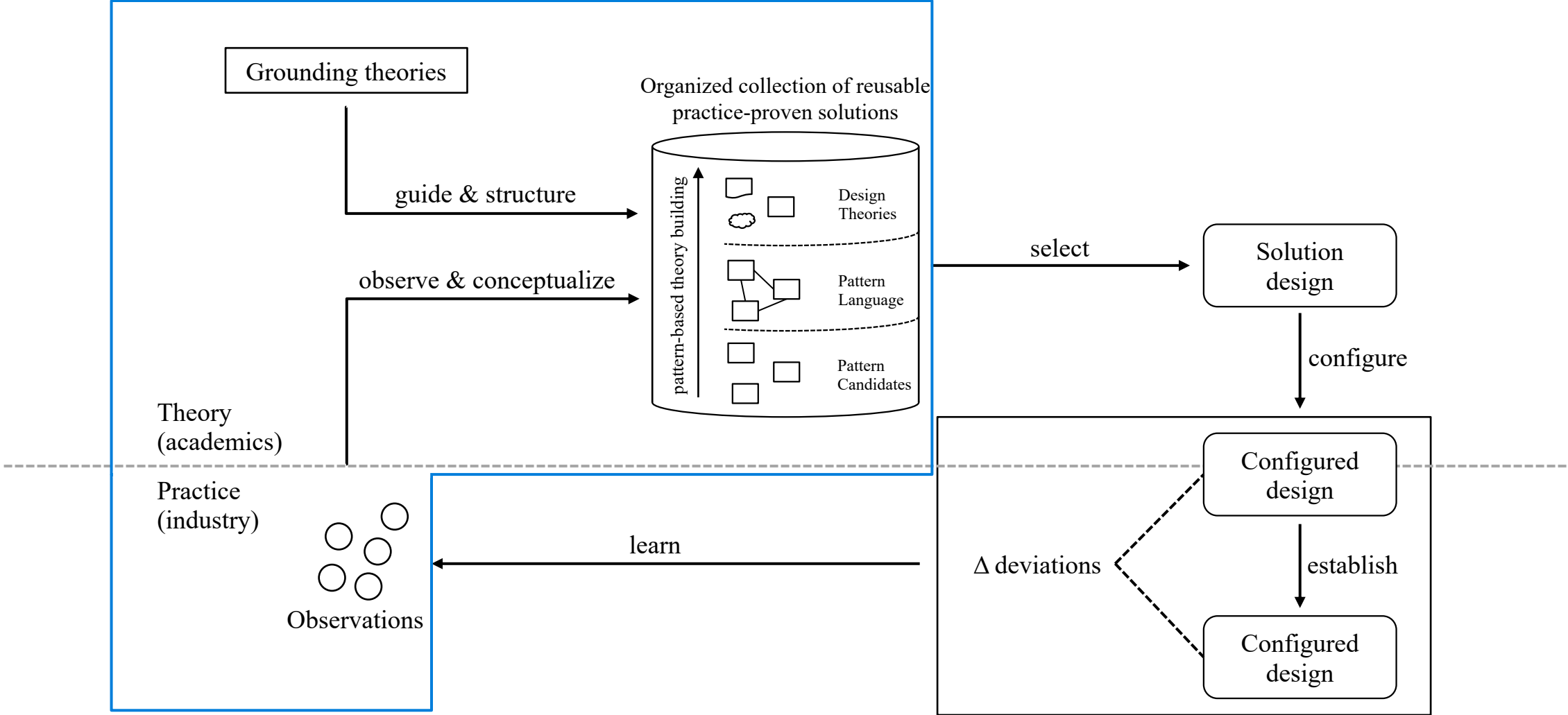
Case Description

Results

Exemplary Pattern Candidate

Conclusion

Approach: Pattern-Based Design Research (PDR) [4]



Motivation

Approach

Case Description

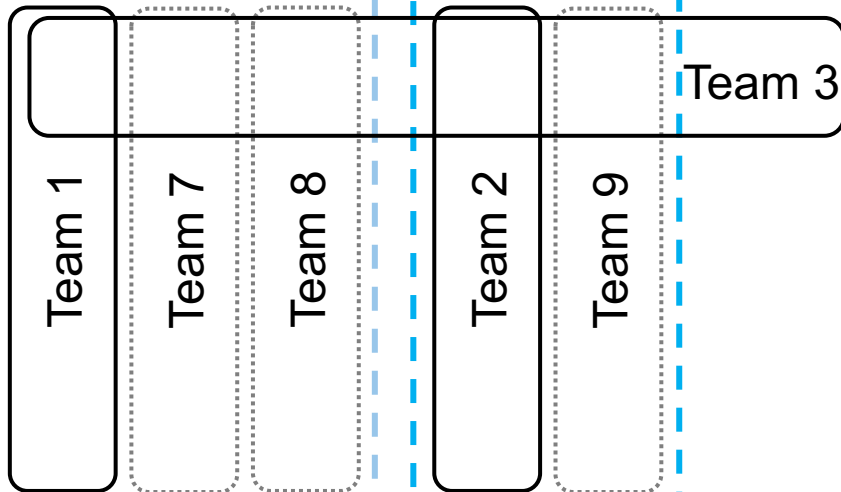
Results

Exemplary Pattern Candidate

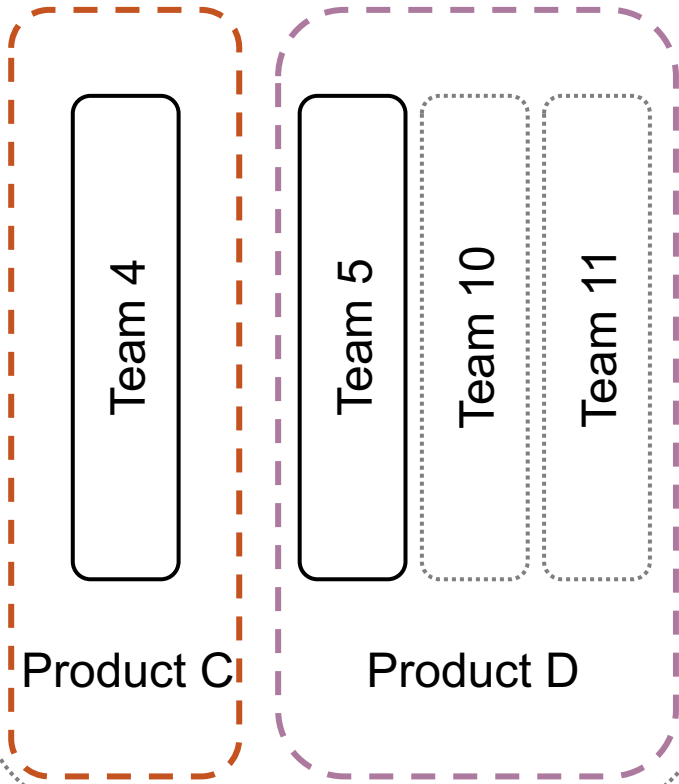
Conclusion

Organization

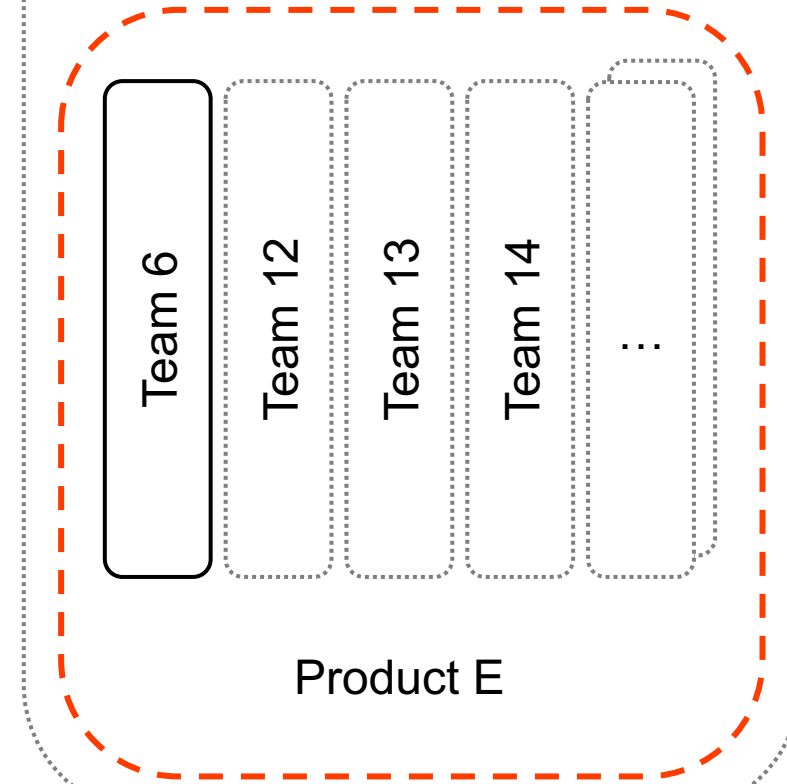
Product Area A



Product Area B



Product Area C



1st degree: direct contact with subjects

- Semi-structured interviews with 15 participants

2nd degree: direct collection without active interaction

- Passive observation of meetings and work of teams

3rd degree: analysis of work artifacts

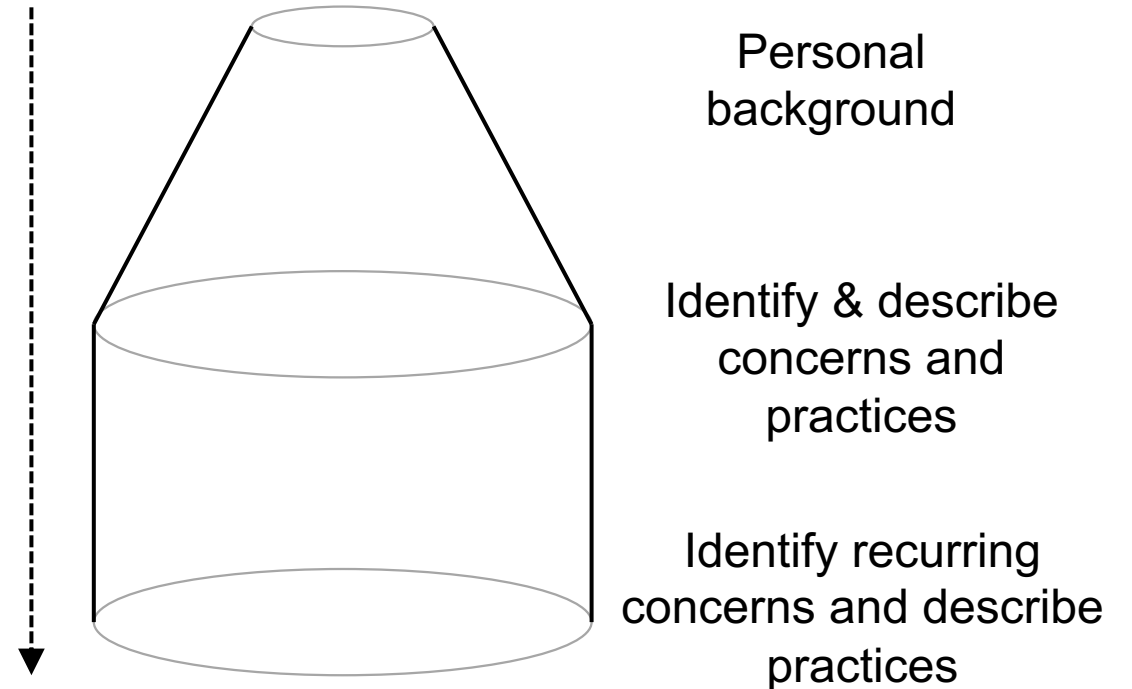
- Data from the intranet used to collect additional information

No.	Role	Team	Experience
1	Agile Consultant	-	6 – 10 years
2	Developer	Team 4	1 – 2 years
3	Developer	Team6	1 – 2 years
4	Developer	Team3	6 – 10 years
5	Development Manager	- / Team5	3 – 5 years
6	Development Manager	-	> 15 years
7	Product Owner	Team4	1 – 2 years
8	Product Owner	Team2	1 – 2 years
9	Product Owner	Team1	11 – 15 years
10	Scrum Master	Team2	3 – 5 years
11	Scrum Master	Team5	3 – 5 years
12	Scrum Master	Team1	1 – 2 years
13	Software Architect	Team1	3 – 5 years
14	Software Architect	Team2	6 – 10 years
15	Developer (Tech Lead)	Team6	11 – 15 years

3 Interview Phases:

- **Introduction:** Information about the participant, role, and team
- **Concerns & Practices I:** Participant identifies top concerns and describes applied solutions
- **Concerns & Practices II:** Participant goes through list of existing concerns, identifies those that apply, and describes solutions

Review session to recap interview and capture feedback.



Coding using MAXQDA2018:

- **Integrated Approach** [9]
- **Provisional Codes:** Existing concerns from literature used as starting point
- **Scheme:** LSADPL used as a general categorization scheme

The screenshot displays the MAXQDA 2018 software interface. The top menu bar includes options like 'Start', 'Import', 'Codes', 'Variablen', 'Analyse', 'Mixed Methods', 'Visual Tools', and 'Reports'. Below the menu is a toolbar with icons for 'Neues Projekt', 'Projekt öffnen', 'Liste der Dokumente', 'Liste der Codes', 'Dokument Browser', 'Liste der Codings', 'Logbuch', and 'Teamwork'. The main workspace is divided into three panes:

- Liste der Dokumente:** A tree view showing a hierarchy of documents. The 'Sets' folder is expanded, showing various interview transcripts such as 'Interview SM3', 'Interview SM1', 'Interview SA1', 'Interview PO2', 'Interview SA2', 'Interview M2', 'Interview M1', 'Interview PO3', 'Interview PO1', 'Interview SM2', 'Interview AC1', 'Interview D2', and 'Interview D4'.
- Liste der Codes:** A tree view showing a hierarchy of codes. The 'Sets' folder is expanded, showing various codes such as 'Interview SM3', 'Interview SM1', 'Interview SA1', 'Interview PO2', 'Interview SA2', 'Interview M2', 'Interview M1', 'Interview PO3', 'Interview PO1', 'Interview SM2', 'Interview AC1', 'Interview D2', 'Distributed Component Leads', 'Cross Team Peer Review', 'Too High-Level Scrum of Scrums', 'Bug Triage Meeting', 'SUPER: Follow The Sun/Follow The Sun', 'Lack of Social Binding Between Teams due to Geographical Distri', 'Interview D4', 'Principles', 'M-Patterns', 'CO-Patterns', 'V-Patterns', 'Anti-Patterns', 'Concerns', 'New Cross-Team Feature Originating from Team', 'Corruption of Shared Codebase', and 'Unclear Mutual Exceptions'.
- Table:** A table showing the coding results for the selected document. The table has columns for 'Kommentar', 'Dokumentgruppe', 'Dokumentname', 'Anfang', and 'Code'. The data is as follows:

Kommentar	Dokumentgruppe	Dokumentname	Anfang	Code
	M-Practices	M-908 Dispatcher	1: 508	SUPER: Follow The Sun
	M-Practices	M-908 Dispatcher	1: 508	SUPER: Follow The Sun(Dispatcher
	Anti-Practices	AP-4 Too many Time Zones	2: 1315	SUPER: Follow The Sun(Dispatcher
	[SoM] Interview M2	20190626 Interviewbogen J Recap	3: 41 617	SUPER: Follow The Sun(Dispatcher
	[Dev] Interview D2	20190619 Transkript M	66	SUPER: Follow The Sun(Dispatcher
	[Dev] Interview D2	20190619 Transkript M	87	SUPER: Follow The Sun(Dispatcher
	[Dev] Interview D2	20190619 Transkript M	66	SUPER: Follow The Sun(Follow The Sun
	[AC] Interview AC1	20190619 Transkript A	26	SUPER: Follow The Sun(Follow The Sun
	[AC] Interview AC1	20190619 Transkript A	26	SUPER: Follow The Sun(Follow The Sun
	[SM] Interview SM2	20190606 Transkript R	14	SUPER: Follow The Sun(Dispatcher
	[SoM] Interview M2	20190517 Transkript J	22	SUPER: Follow The Sun(Dispatcher
	[SoM] Interview M2	20190517 Transkript J	23	SUPER: Follow The Sun(Dispatcher
	[SoM] Interview M2	20190517 Transkript J	26	SUPER: Follow The Sun(Dispatcher
	[SoM] Interview M2	20190517 Interviewbogen J	3: 789	SUPER: Follow The Sun(Dispatcher
	[SoM] Interview M2	20190517 interviewbogen J	5: 29 685	SUPER: Follow The Sun(Dispatcher
	[SoM] Interview M2	20190517 interviewbogen J	5: 24 537	SUPER: Follow The Sun(Dispatcher

Outline



Motivation

Approach

Case Description

Results

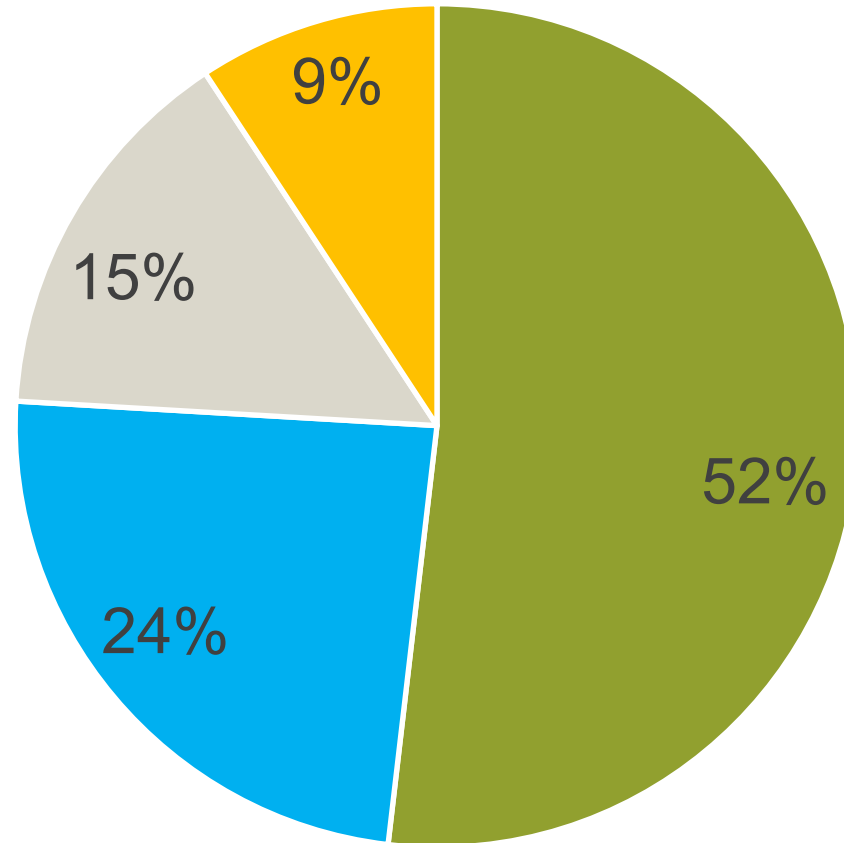
Exemplary Pattern Candidate

Conclusion

Concerns



New Concerns – Scaling Levels



■ Team Level ■ Program Level ■ Organization Level ■ Enterprise Level

Identified Good and Bad Practices

In the study we identified:

- 24 Coordination practices
- 8 Methodology practices
- 6 Viewpoint practices
- 4 Bad practices
- 2 Principles

M-1 ** Mono-Repo	M-7 * Dedicated Person to Deal with Annoyments	V-5 **** Team Homepage	P-1 * Prerequisites of Autonomous Teams
M-2 * Mixed Sprints	M-8 * Requirement Separation	V-6 ** Persona	P-2 * Spread Knowledge
M-3 ** Assigning Rights	V-1 * Roadmap	AP-1 * Rantrospective	
M-4 *** Follow the Sun	V-2 * Task Dependency Mapping	AP-2 *** Demo Driven Development	
M-5 ** Ship-Captain	V-3 * Milestone Planning Board	AP-3 ** Don't Use Agile as Magic Bullet	
M-6 *** Sprint Zero	V-4 * Sailboat Retrospective	AP-4 * Too High-Level Scrum of Scrums	

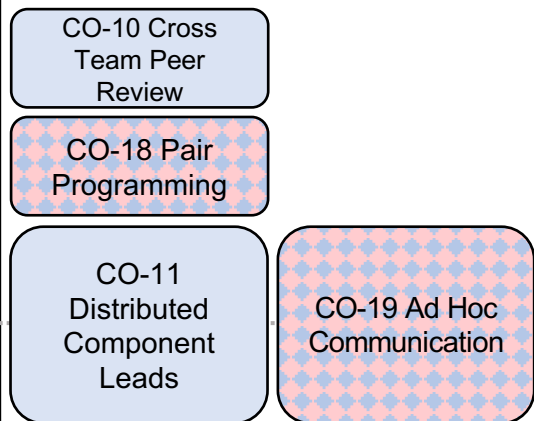
Coordination Mode

Inter-Team Coordination (light blue box) Intra-Team Coordination (light red box)

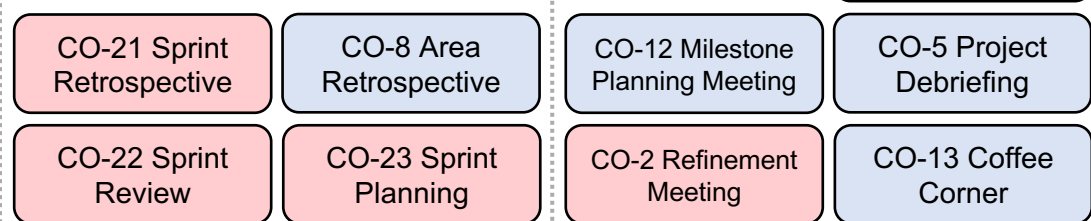
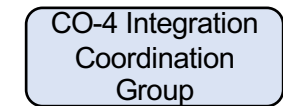
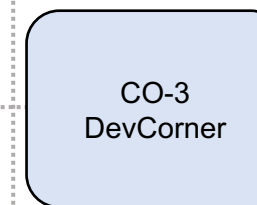
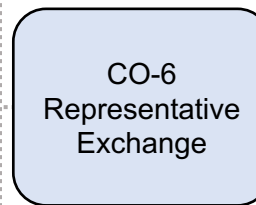
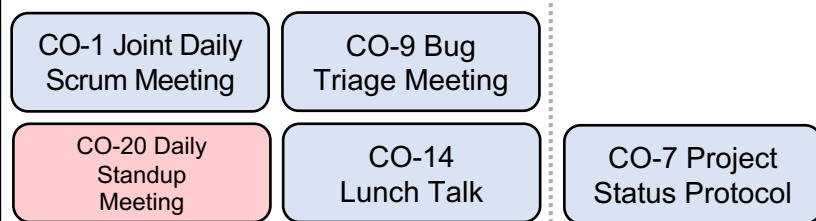
Impersonal Mode



Personal Mode



Group Mode



> weekly

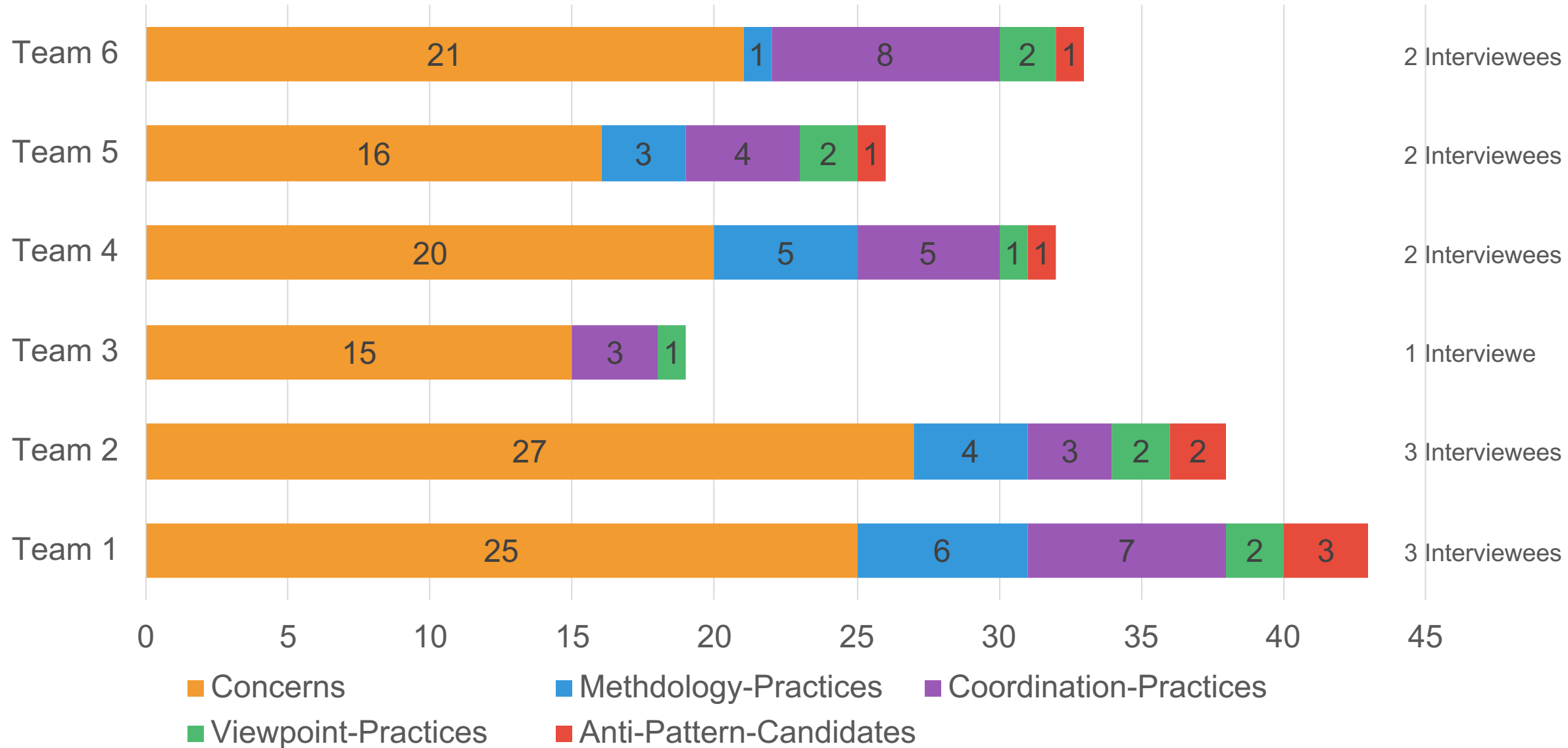
weekly

bi-weekly

< bi-weekly

Frequency

Results per Team



Outline



Motivation

Approach

Case Description

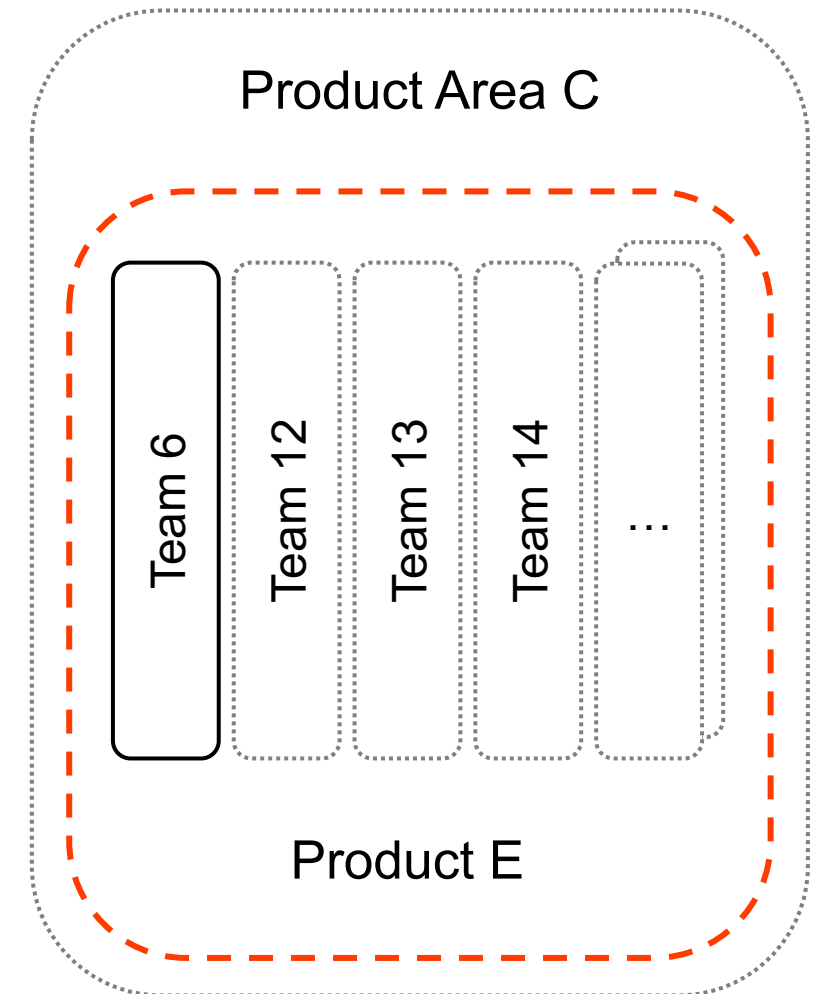
Results

Exemplary Pattern Candidate

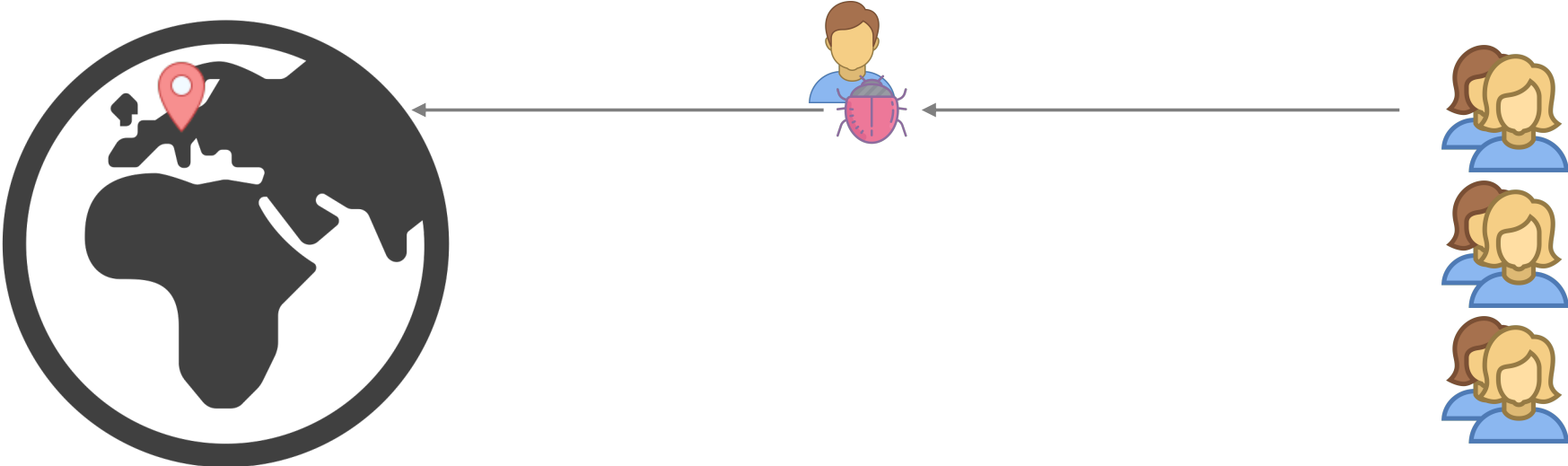
Conclusion

M-4 Follow the Sun

- Applied by Team 6 and Product E
- Also used by one of the Development Managers in his former company (also as a Development Manager)
- The Agile Consultant mentioned this practice is one of the view benefits of distribution across multiple time zones



Exemplary Pattern



Outline



Motivation

Approach

Case Description

Results

Exemplary Pattern Candidate

Conclusion

RQs

RQ1: 29 concerns identified

RQ2: 24 Coordination-Practices, 8 Methodology-Practices, and 6 Viewpoint-Practices

RQ3: 4 Anti-Pattern Candidates

Key Findings

- Most concerns on team level
- Concerns often relevant for multiple stakeholders
- Case organization largely relies on group mode coordination
- Open work area facilitates ad hoc communication and personal information exchange between people

Outlook

- *Solution design & application and evaluation & learning* phases of PDR
- Identify actual patterns from good practices by conducting similar projects at other organizations & conducting quantitative studies

- [1] Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review.
- [2] Uludağ, Ö., Kleehaus, M., Caprano, C., & Matthes, F. (2018). Identifying and Structuring Challenges in Large-Scale Agile Development based on a Structured Literature Review.
- [3] Uludağ, Ö., Harders, N.-M., & Matthes, F. (2019). Documenting Recurring Concerns and Patterns in Large-Scale Agile Development.
- [4] Buckl S., Matthes F., Schneider A.W., & Schweda C.M. (2013) Pattern-Based Design Research – An Iterative Research Method Balancing Rigor and Relevance.
- [5] Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering.
- [6] Van De Ven, A., Delbecq, A., & Koenig, R. (1976). Determinants of Coordination Modes within Organizations.
- [7] Maryam Kausar and Adil Al-Yasiri. (2015). Distributed agile patterns for offshore software development.
- [8] Erran Carmel, Yael Dubinsky, and Alberto Espinosa. (2009). Follow The Sun Software Development: New Perspectives, Conceptual Foundation, and Exploratory Field Study.
- [9] Daniela S. Cruzes and Tore Dybå. (2011). Recommended Steps for Thematic Synthesis in Software Engineering



Moritz Schüll

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business
Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.17132
Fax +49.89.289.17136

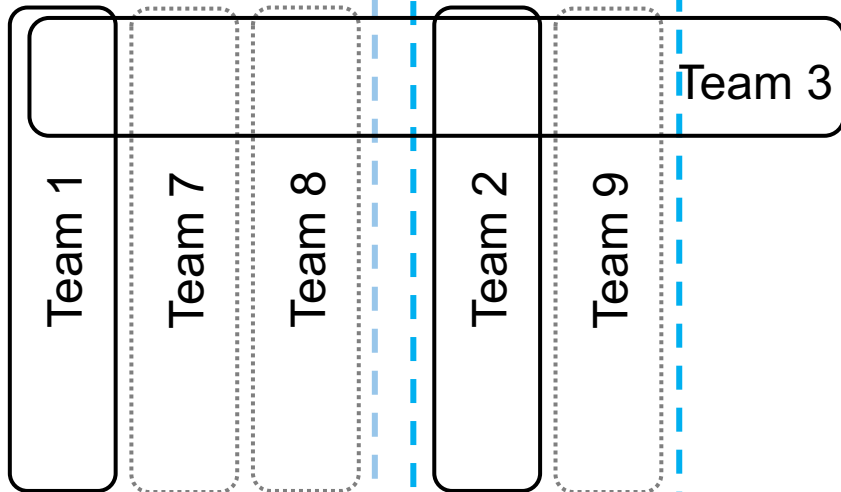
moritz.schuell@tum.de
www.matthes.in.tum.de



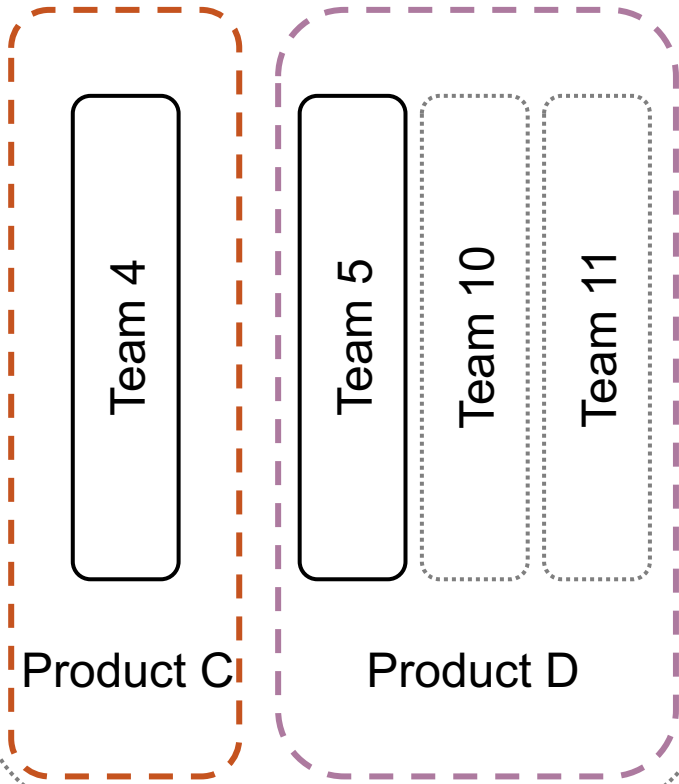
Backup Slides

Organization

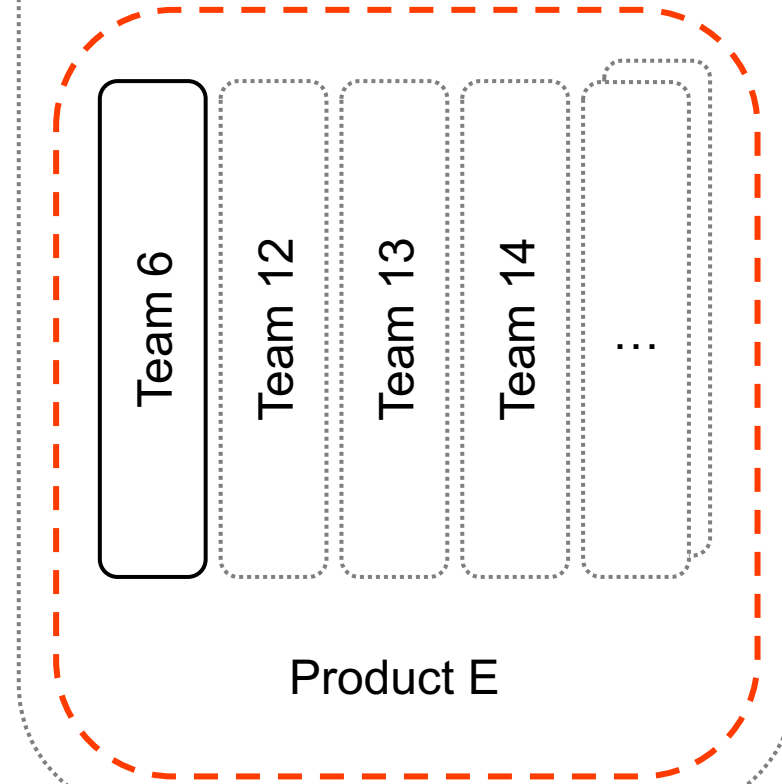
Product Area A



Product Area B



Product Area C



Team1

- 14 members (Scrum Master, Product Owner, Software Architect, DevOps Engineer from Team3, and 10 Developers)
- Developing a solution to make all business objects maintained by the product suite of the organization accessible through a central API.
- Working together with teams from Walldorf and Potsdam

Team2

- 7 members (part-time Scrum Master, Product Owner, Tech Writer, Software Architect, DevOps Engineer from Team3, and 4 Developers)
- Developing a cloud service for authentication and authorization that will be used within the whole product suite. Intended to replace the product-level identity management with suite-level identity management.
- Working together with team from Walldorf

Team3

- 4 members (Team Lead, 3 DevOps Engineers)
- Takes care of creating and maintaining continuous integration and deployment pipelines for the teams on Product A and B.

Team4

- 7 members (Product Owner, and 6 Developers)
- Develops a platform that enables real-time analysis of software logs, and reaction to events in the logs. Builds on the product built by Team5 etc.

Team5

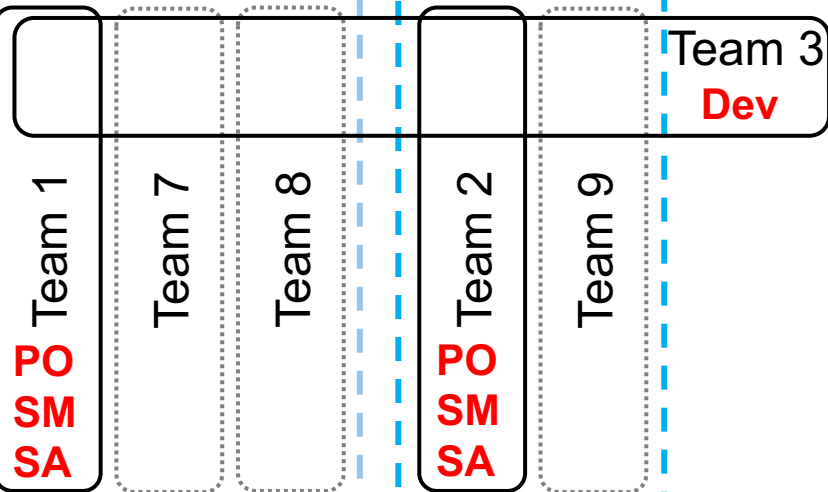
- 5 members (part-time Scrum Master, Product Owner (= Development Manager), and 3 Developers)
- Develops a log management solution, focusing on retrieval and collection of logs generated by products of the organization. Makes logs searchable and analyzable.
- Working together with teams from Canada and Poland

Team6

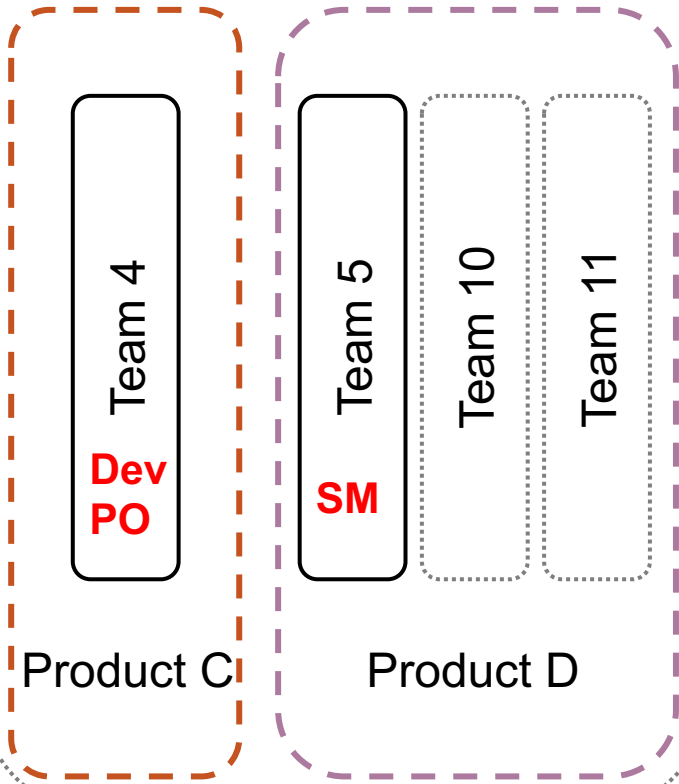
- 12 members (Scrum Master, Product Owner, Enterprise Architect, Quality Assurance, Tech Writer, and 7 Developers)
- Develops a software that automates the deployment and operation of other products of the case organization.
- Working together with teams from Germany, Poland, Canada, and USA

Organization

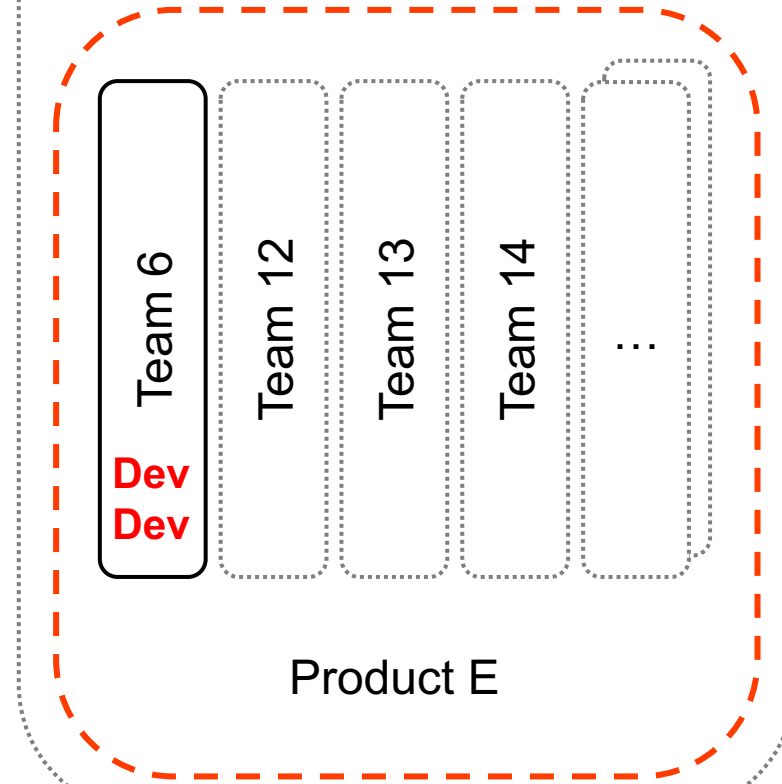
Product Area A



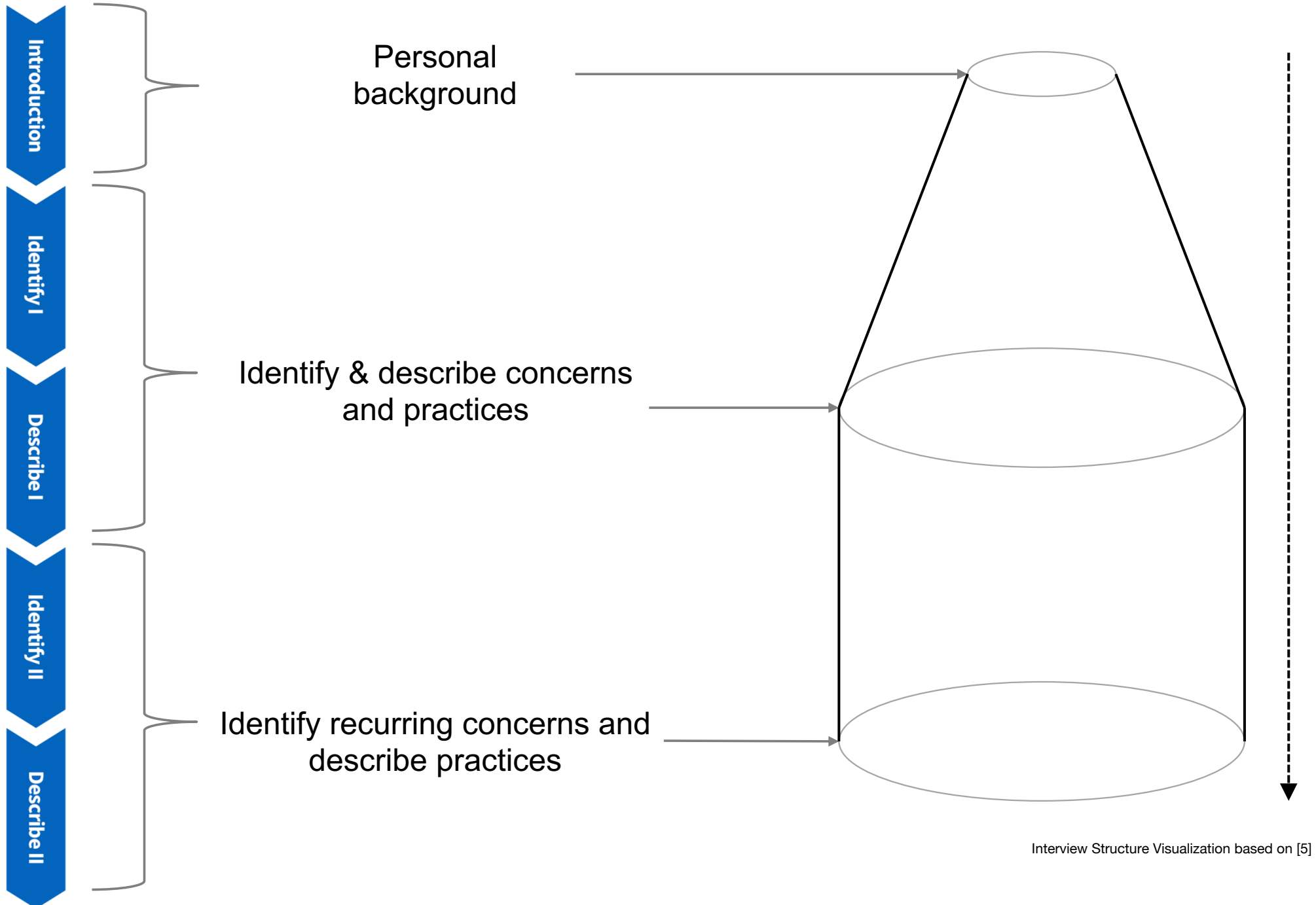
Product Area B



Product Area C







Three different coordination modes in organizations [6]:

Impersonal Mode	Personal Mode	Group Mode
<ul style="list-style-type: none"> • Plans • Schedules • Policies • Information & Communication Tools 	<ul style="list-style-type: none"> • Individual role occupants • Vertical or horizontal 	<ul style="list-style-type: none"> • Staff meetings or committees • Scheduled or unscheduled
<p style="text-align: center;">Coordination by Programming</p> <p style="text-align: center;">Codified blueprints; actions are impersonally specified</p>	<p style="text-align: center;">Coordination by Mutual Adjustment</p> <p style="text-align: center;">Coordination based on feedback and new / changing information</p>	

Pattern Overview

ID	M-4
Name	Follow the Sun
Alias	Dispatcher
Summary	To deal with urgent bug fix requests and customer issues, the Follow the Sun practice helps to guarantee a certain reaction time. The distribution of the teams across multiple time zones, together with the role of a 'Dispatcher', is leveraged to have 24/7 availability for urgent issues.

Example

The initial version of Product E of the case organization has been released half a year ago. Since then, the live running installations of the product have significantly increased. This led to a high inflow of bug reports and urgent customer issues. The company is obliged by contract to react to reported issues in less than two hours. To keep up with this, the project teams set up a 'Dispatcher' role and have a rotating on-call team in each time zone.

Context

The company has to react to issue reports in a given time frame. The number of incoming issues and reports is high. Teams are distributed across multiple time zones.

Problem

C-20 Facilitating Communication between Agile Teams and other Teams using Traditional Practices
C-114 Dealing with Urgent Bugfix Requests
C-120 Working across Multiple Time Zones

Forces

Customers can report issues at any given time, all around the globe. The company needs to make sure to be on call all day. The protective setup of Scrum and other agile methods during a sprint make it hard to address urgent issues during a running sprint.

Solution

To achieve very quick reaction times to urgent bug fix requests, set up a '24/7 Team'. This team is distributed across different time-zones, in such a way that 24/7 coverage is ensured by always having one active team. Create a 'Dispatcher' role, that takes bugs and reported issues and directly assigns them to development teams or developers – even inside a running sprint. Each time-zone team has one Dispatcher that is responsible for six hours a day, so 24/7 team availability is achieved by rotation. The Dispatcher has access to the bug database / error tracking system. The Dispatcher has the right to interfere the work of the Scrum development teams and can assign tasks even inside a running sprint. If necessary, the Dispatcher can also release code bypassing the normal review procedure. During the remaining two hours of the workday, the Dispatcher ensures that all currently running work is handed over to the following Dispatcher and 24/7 team members.

Consequences

Benefits:

- The reaction time to urgent requests is guaranteed to be low around the clock.
- The normal Scrum development setup can continue in parallel without changes.

Liabilities:

- The Scrum sprints and meeting cycle can be disturbed.

See Also

This practice can be applied together with V-2: Task Dependency Mapping to visualize the necessary handovers across time zones.

Other Standards

A similar pattern is documented in the pattern catalogue by [7] and in [8].

- Construct Validity
 - data source triangulation (multiple sources of data)
 - chain of evidence by coding
 - BUT: cannot make study database public
- Internal Validity
 - preparation document sent alongside invitation to interviews
 - Already existing concern only shown at the end of interviews
- External Validity
 - pattern as artifacts → abstraction
 - BUT: findings (most likely) restricted to agile environments
- Reliability
 - findings may not be reproducible by similar research (very specific to organization and time of observation)