Empirical Studies to Identify Challenges and Probe Good Practices in the Adoption of Scaled Agile Methods in the Field of Vehicle Dynamics Development of an OEM

Clara Lea Buchholz, 05.08.2019, Bachelor's Thesis Final Presentation

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

Outline

1

Introduction

1. Motivation
2. Research Questions
3. Research Approach

2

Case Study

1. Case Study Design
2. Transformation
3. Recurring Concerns
4. Good Practices

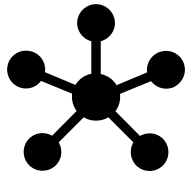
3

Conclusion

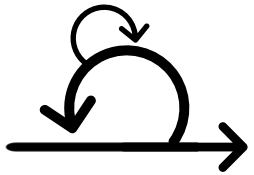
1. Key Findings
2. Future Work

Motivation

Inflexible traditional software development methods [1]



Agile methods were initially designed for small, individual & co-located teams [3]



Agility & Flexibility [2]

Large scaled agile IT organizations [1]

Quickly changing business environment [1]

Challenges [4]

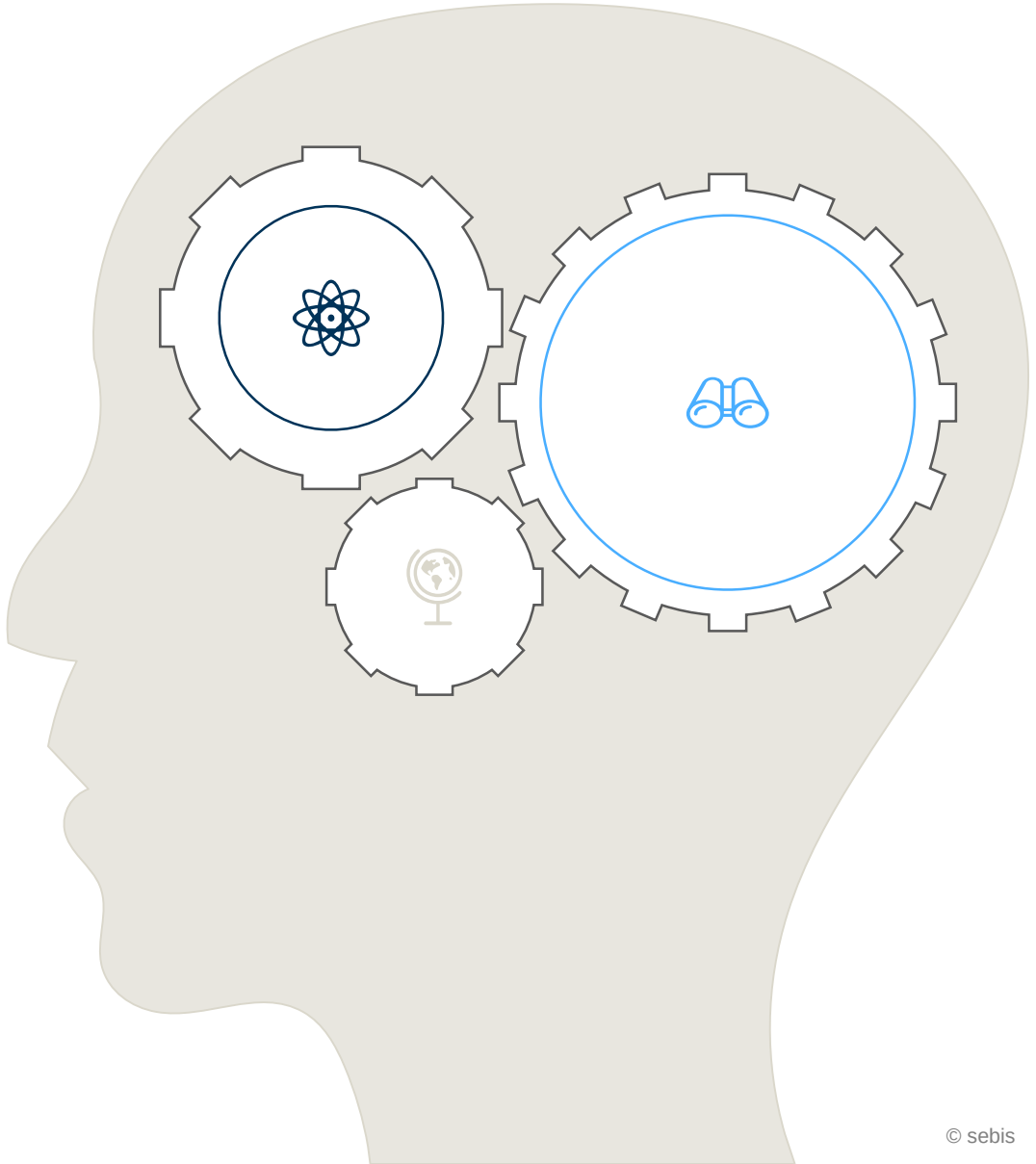
[1] Papadopoulos (2015); [2] Highsmith (2002); [3] Dingsøyr et al. (2012); [4] Uludağ et al. (2018)

Research Questions

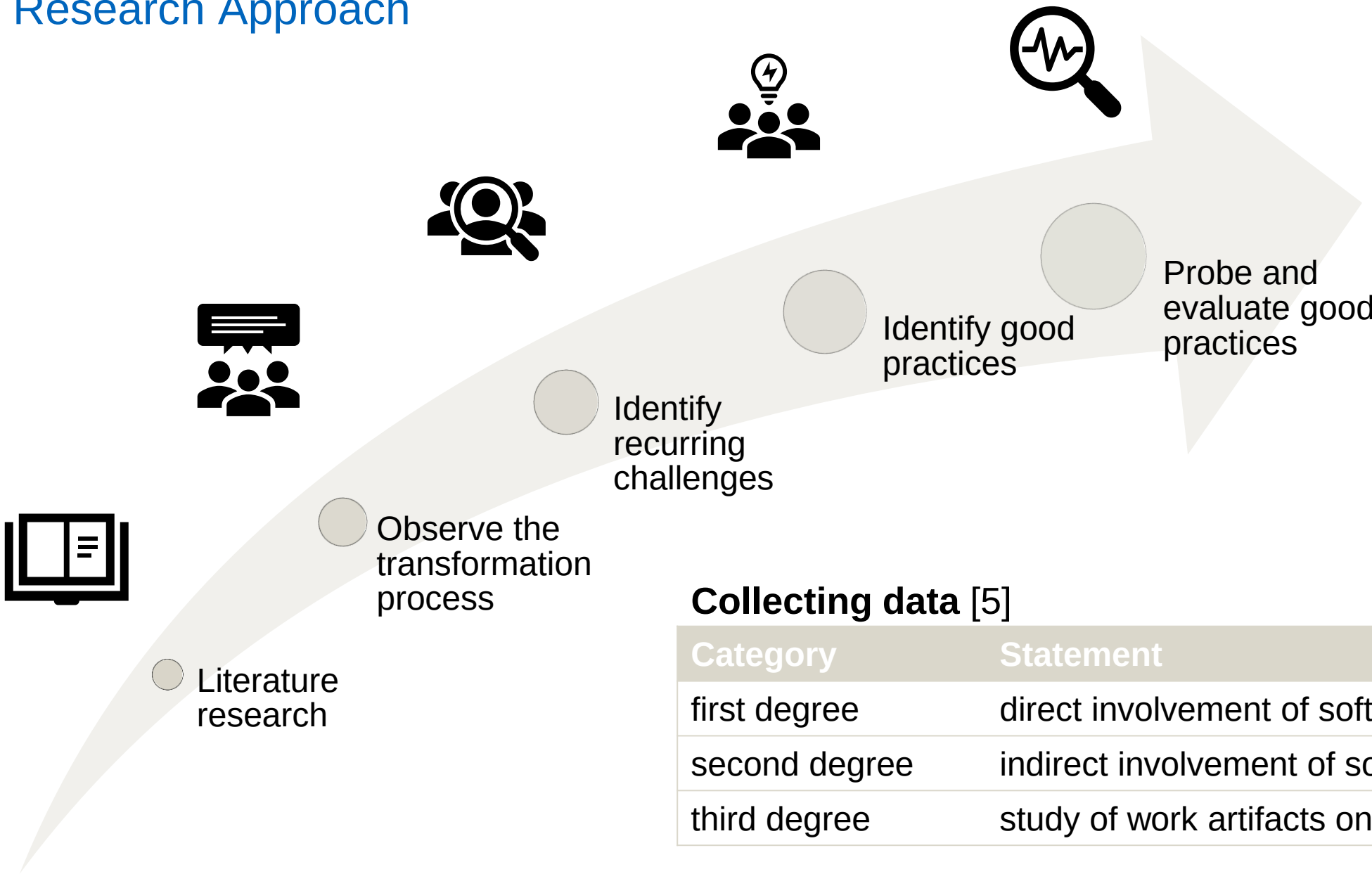
1. How does the large-scale agile transformation take place at the case organization?

2. What are challenges and success factors within the large scale agile transformation at the OEM?

3. What are good practices to address the observed challenges within the large scale agile transformation at the OEM?



Research Approach



Collecting data [5]

Category	Statement
first degree	direct involvement of software engineers
second degree	indirect involvement of software engineers
third degree	study of work artifacts only

[5] Lethbridge et al. (2005)

Outline

1

Introduction

1. Motivation
2. Research Questions
3. Research Approach

2

Case Study

1. Case Study Design
2. Transformation
3. Recurring Concerns
4. Good Practices

3

Conclusion

1. Key Findings
2. Future Work

Data Collection – First Degree [5]

Shadowing

Observation of daily scrum meetings, sprint plannings, sprint reviews, sprint retrospectives, refinements, workshops and more [5]

Feedback Talks

Informal, unstructured interviews with different stakeholders [6]

Interviews

Semi-structured interviews [7] with 14 different interviewees (product owner, scrum master, developer, leader, head of department)

[5] Lethbridge et al. (2005); [6] Robson (2002); [7] Runeson and Höst (2009)

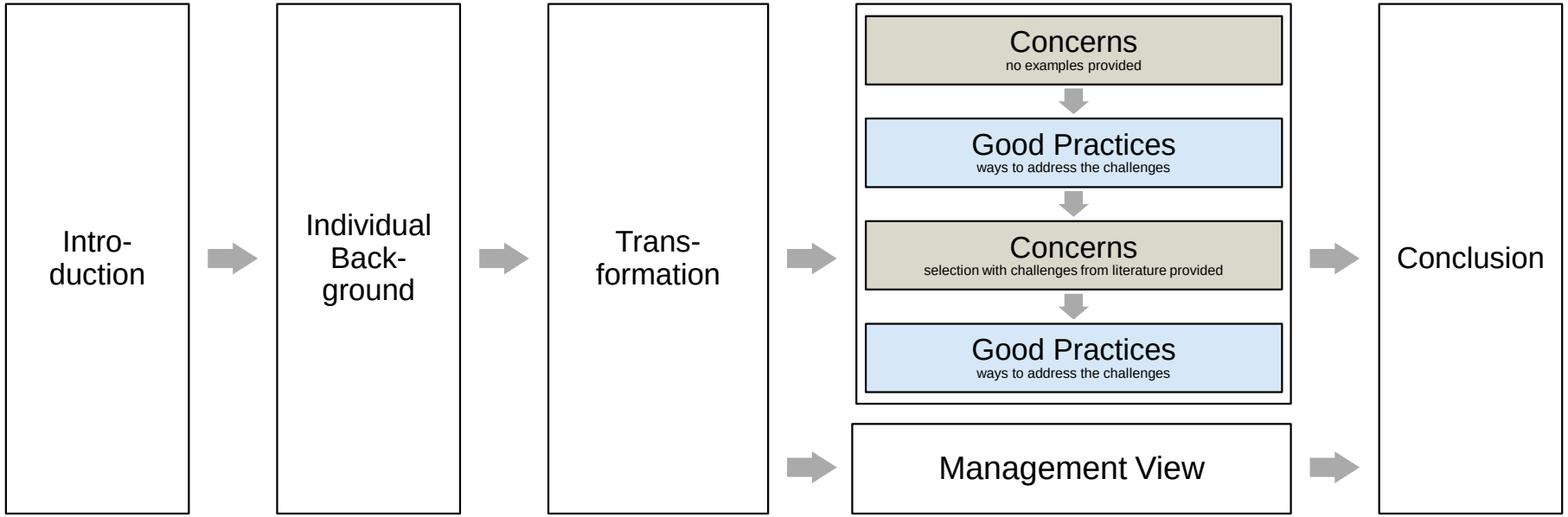
Data Collection: Interviews

14 interviews

semi-structured [7]

60-80 min

recorded, transcripted and coded



Team	Scrum Master	Product Owner	Developer	Manager
A	1	1	1	1
B			2	
C	1	1		1
D				
E	1		2	
F	1			
Sum	**4**	**2**	**5**	**2**

+ Head of Department

[7] Runeson and Höst (2009)

Data Collection – Third Degree [5]

Provided Data

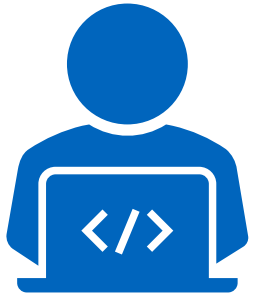
Slides, coaching material for the implementation of the transformation, workshop material about the vision, documentations by other departments, wiki pages, tables about structural, organizational and technical information [5]

[5] Lethbridge et al. (2005)

Case Organization - Vehicle Dynamics Development

Main Department
~ 1000 employees

Department
~ 180 employees



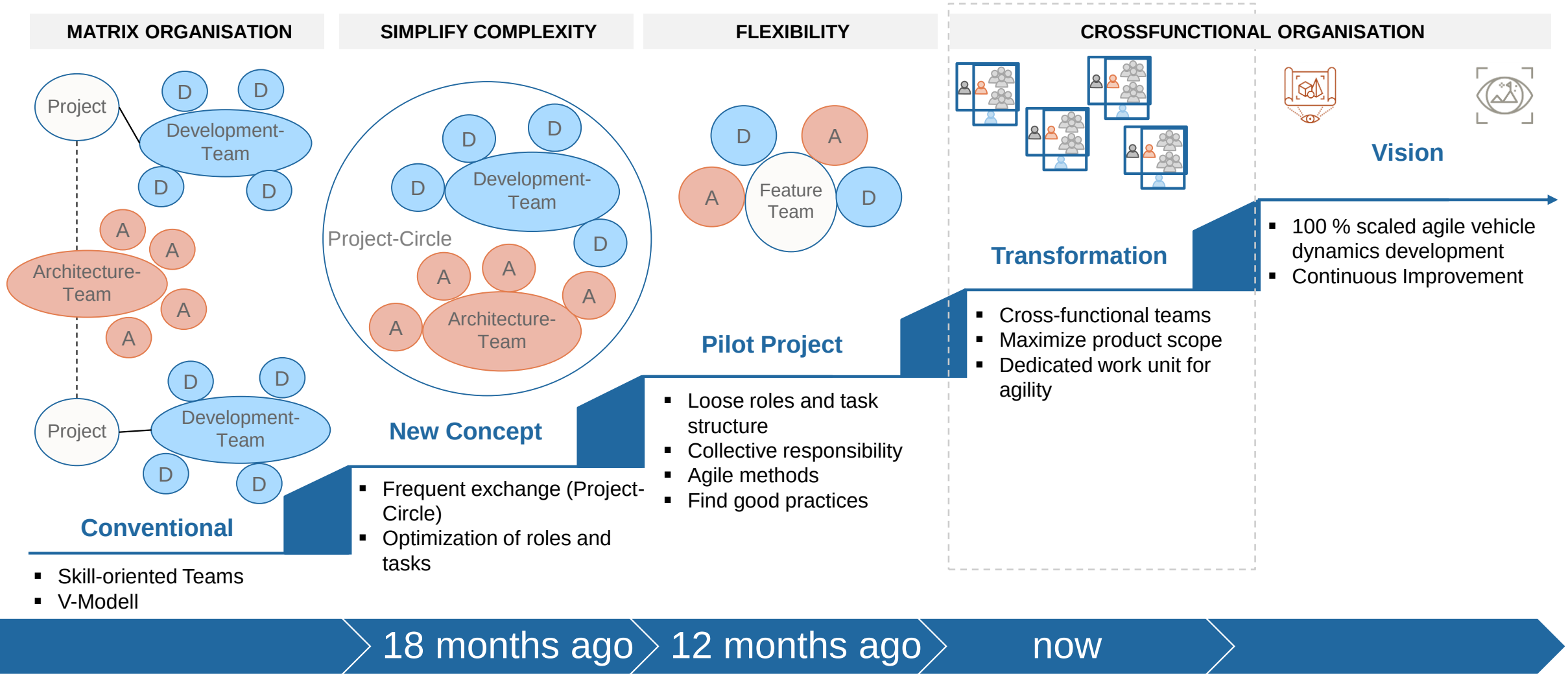
Mainly producing software for intern stakeholders

Traditional matrix organization → **Agile feature teams**

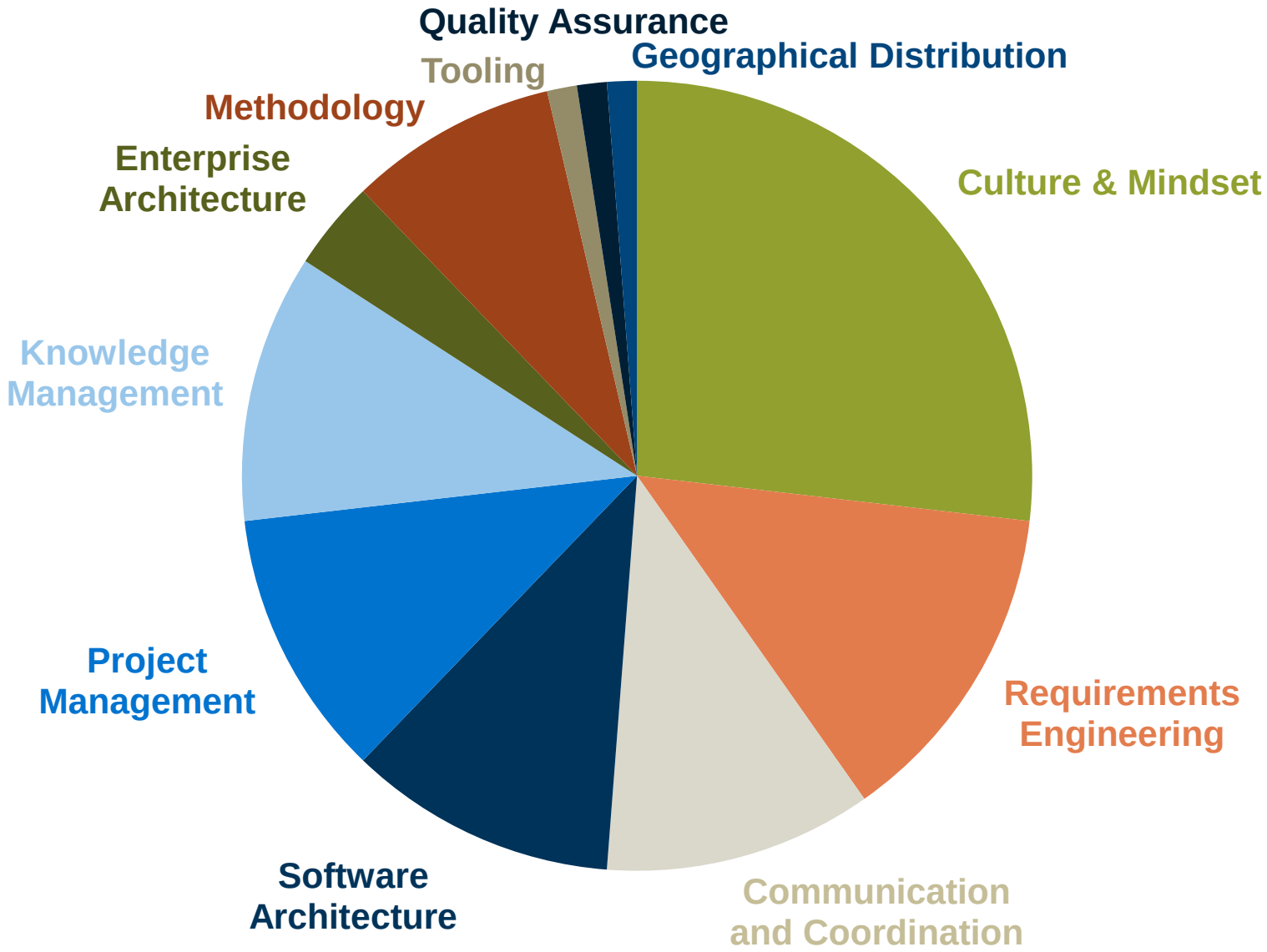
Transformation (since April 2019)

Goal: 100% agile

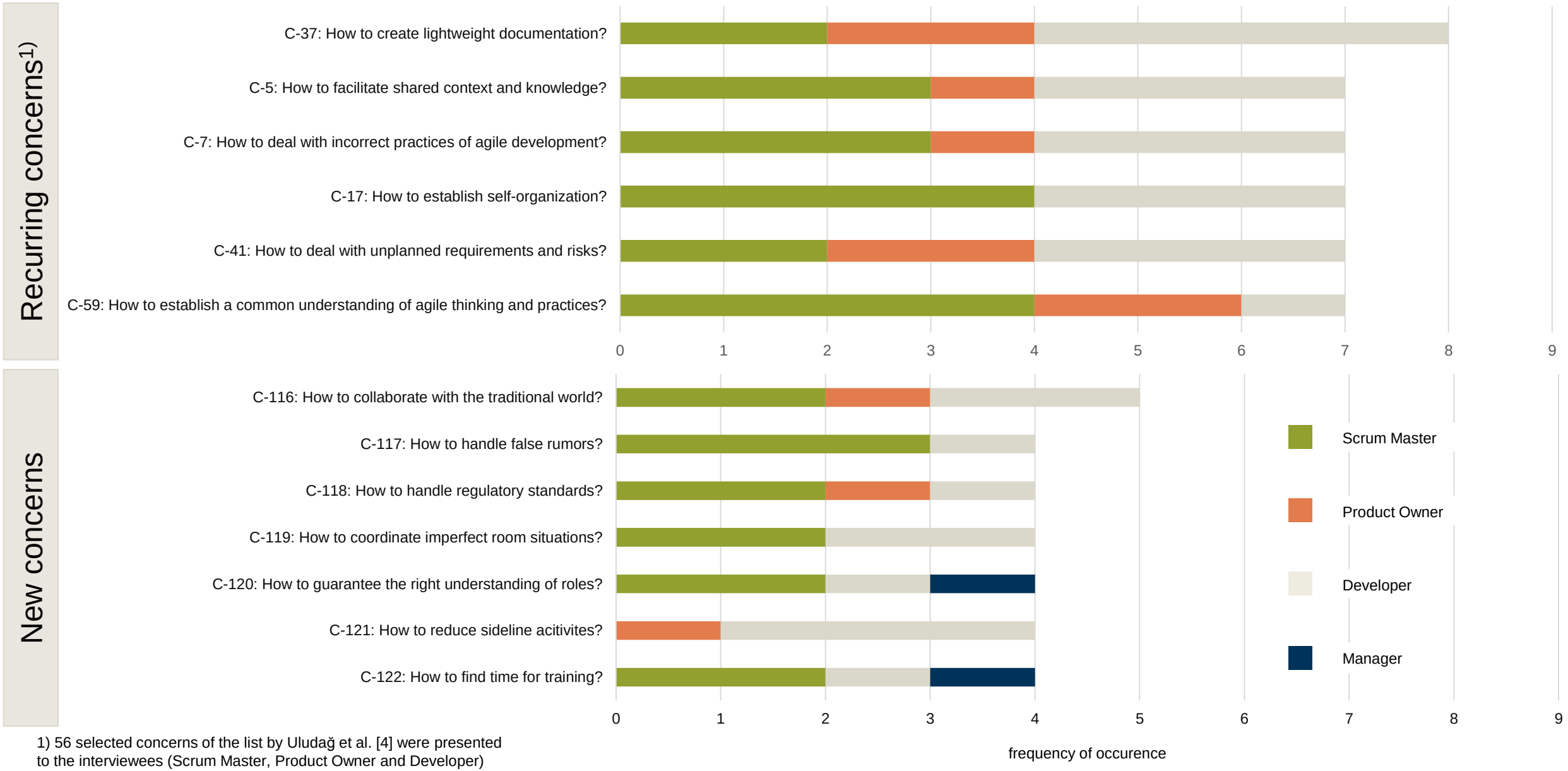
Transformation
Progress



Categories

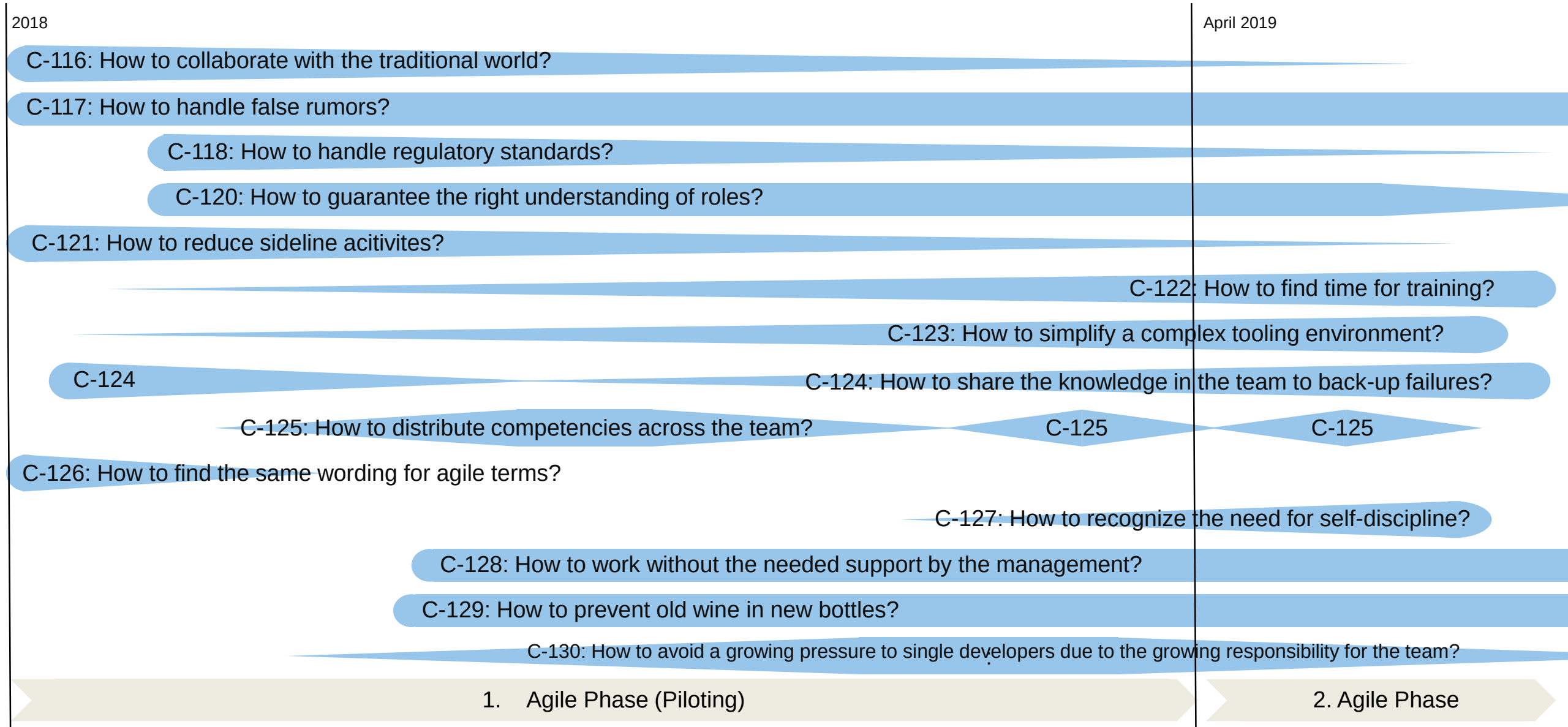


Most Frequently Occurring Concerns



1) 56 selected concerns of the list by Uludağ et al. [4] were presented to the interviewees (Scrum Master, Product Owner and Developer)

Progress of Concerns



Main Scaling Practices

Special work unit for agility

Scrum master coordination and agile coaching

(Empowered) Communities of Practice

Networking,
competency and knowledge management,
improvement of processes and tools

Product Owner Board

Circle including product owners and managers to coordinate additional tasks

Good Practices

Identified Pattern Candidates

P-01
Create T-Shaped People

CO-01
Community of Practice

CO-02
Backlog Grooming

A-01
Dont‘t Combine Developers from Different Organizational Units in One Development Team

M-01
Empowered Community of Practice

M-02
Piloting

M-03
Travelling

M-04
Share the Change

M-05
Come to our Demos

M-06
Shadowing

M-07
Share a Mailbox

V-01
Radar Chart

V-02
Roadmap

V-03
Project Plan

V-04
Synchronized Calendar

V-05
Starfish

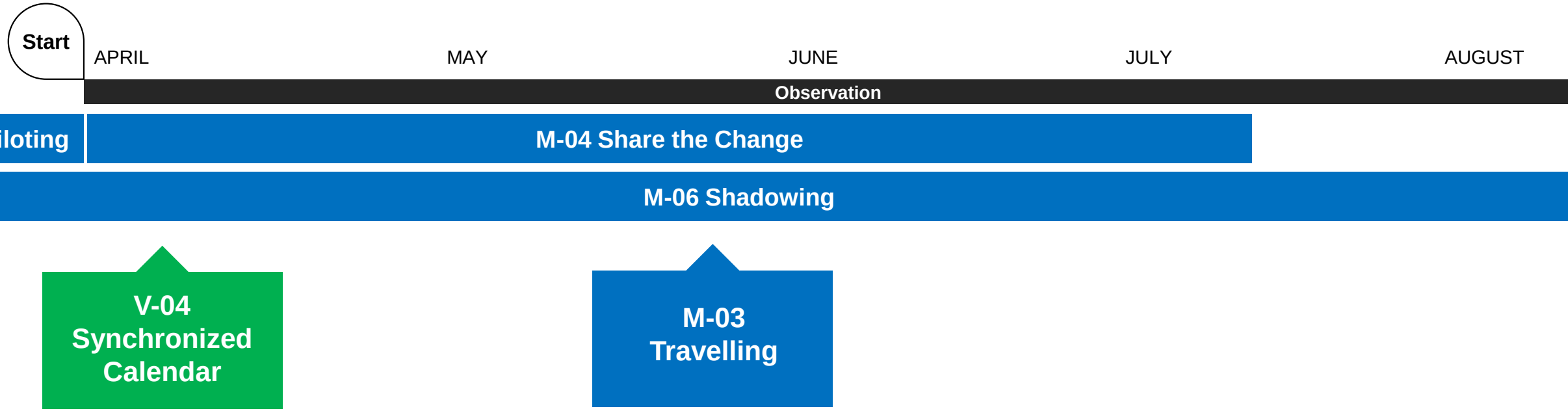
V-06
Burndown Chart

Scaling Practices

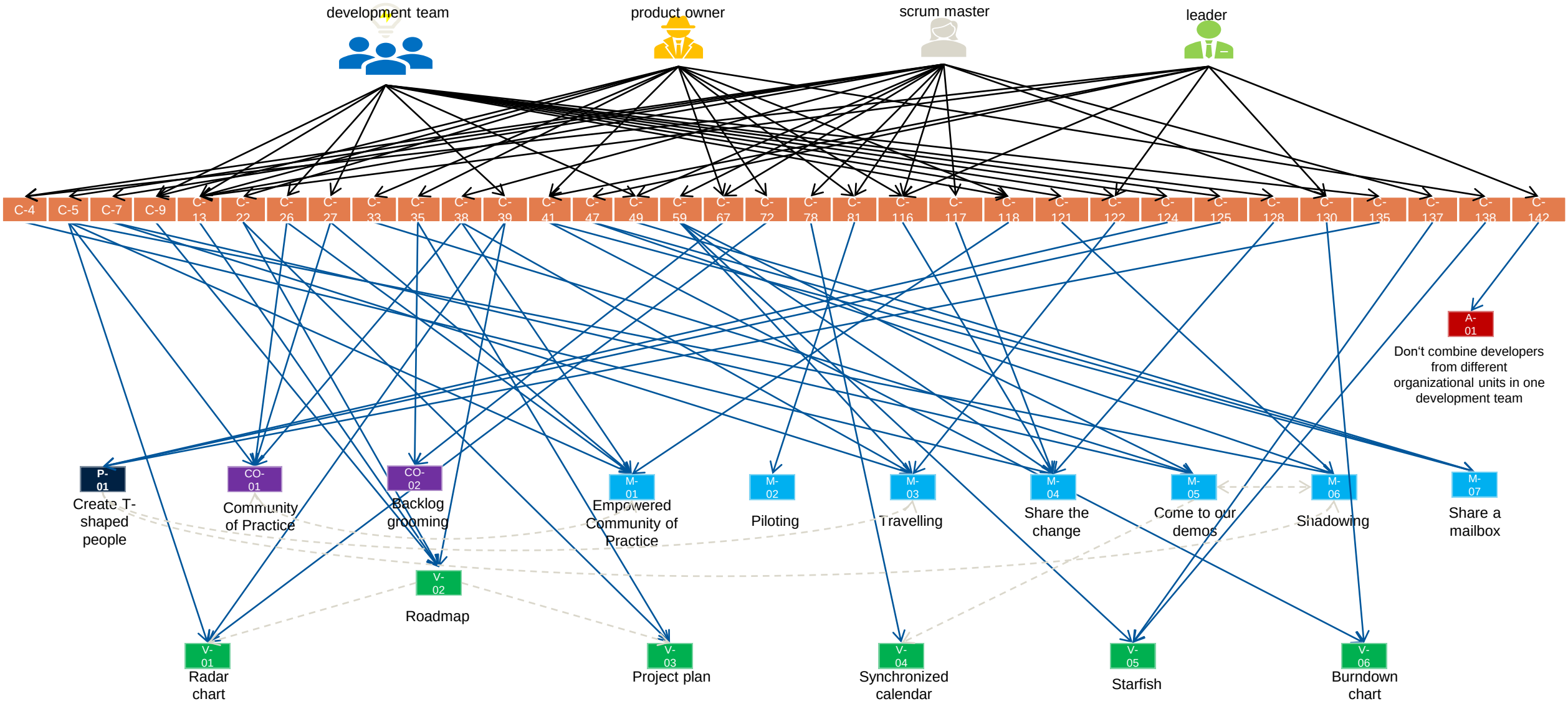
Special work unit for agility

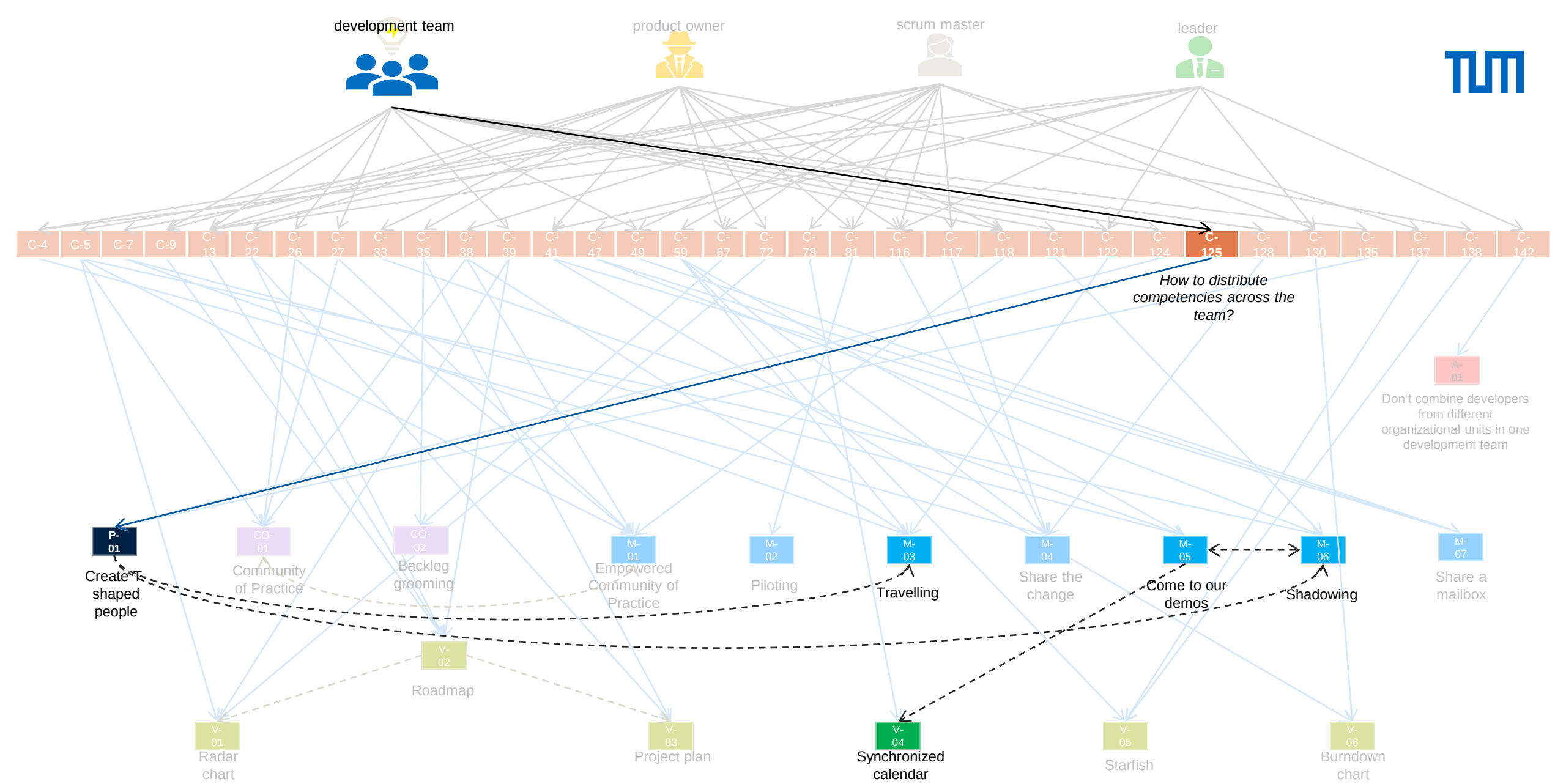
CO-01, M-01 (Empowered) Communities of Practice

Product Owner Board



Relationship between Roles, Concerns and Pattern Candidates




development team
product owner
scrum master
leader
C-4
C-5
C-7
C-9
C-13
C-22
C-26
C-27
C-33
C-35
C-38
C-39
C-41
C-47
C-49
C-59
C-67
C-72
C-78
C-81
C-116
C-117
C-118
C-121
C-122
C-124
C-125
C-128
C-130
C-135
C-137
C-138
C-142
How to distribute competencies across the team?
A-01
Don't combine developers from different organizational units in one development team
P-01
Create T-shaped people
CO-01
Community of Practice
CO-02
Backlog grooming
M-01
Empowered Community of Practice
M-02
Piloting
M-03
Travelling
M-04
Share the change
M-05
Come to our demos
M-06
Shadowing
M-07
Share a mailbox
V-02
Roadmap
V-01
Radar chart
V-03
Project plan
V-04
Synchronized calendar
V-05
Starfish
V-06
Burndown chart
C
role faces concern
C
A/P/CO/M/V-Pattern C.
concern is addressed by pattern
P/CO/M/V-Pattern C.
M-Pattern Candidate
combined use recommended
P/M/V-Pattern C. (1)
M/V-Pattern C. (2)
to fullfill (1) use (2)

Documentation of Pattern Candidates [8]

B.1.9. Project Plan

Pattern Overview	
ID	V-03
Name	PROJECT PLAN
Alias	Calendar Overview
Summary	In every planning meeting the product owner presents a project plan to justify his prioritization of the sprint backlog and give the development teams an ov… relevant tasks beyond the current sprint.

Problem

- **C-22:** *How to balance short-term and long-term goals?*
- **C-35:** *How to define clear and visible priorities?*

Forces

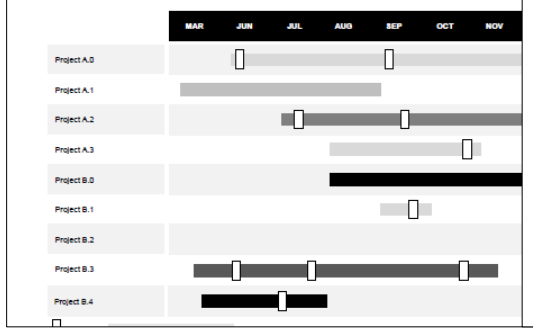
Product owner must summarize all dates in one table, that can lead to a…

Solution

To inform the development teams of a product about the deadlines for… new appointments the product owner can bring a calendar as overv… meeting. The product owner marks the relevant dates for the next few… why he prioritized which task in the current sprint. The development… request if something is not clear and have access to the calendar ov… The calendar is organized as a table with columns as days/weeks a… projects/series. For a better understanding different types of deadl… (example clustering for these milestones: model, implementation, t… …).

Variants

The calendar can include as much information as useful for the team. … is important to limit the information to guarantee a clear overview. … can vary for different teams.



B.1.4. Come to Our Demos

Pattern Overview	
ID	M-05
Name	COME TO OUR DEMOS
Alias	*none*
Summary	All Stakeholders and other interested colleagues are invited to join the meetings of a development team to share the used methodology and get an overview of the current status.

Context

During a transformation new teams lack agile experience in the first… Stakeholders lack the knowledge of the agile mindset and distrust the…

Problem

- **C-7:** *How to deal with incorrect practices of agile development?*
- **C-33:** *How to build trust of stakeholders in agile practices?*
- **C-59:** *How to establish a common understanding of agile thinking and…*

Forces

Stakeholder as well as some new developer using agile practices may… development methods and do not have time to come to all meetings. … visitors may not understand all content which is discussed in the mee…

Solution

Interested Colleagues (for example inexperienced new developer, scru… uct owner) can join the meetings (especially the daily scrum meetin… the review meeting) of experienced development teams to get insig… technical but also the agile work (see also **M-06** SHADOWING). Mana… should use the review meeting to get an overview of the current wor… team instead of request updates. To be as transparent as possible a ag… the visitors. So they can check, whether the content of the review me… them. To get the highest possible outcome of the meetings demos sh… tured and easy understandable for all visitors. The development team… the content which is presented.

5.2.2. Community of Practice

Pattern Overview	
ID	CO-01
Name	COMMUNITY OF PRACTICE
Alias	*none*
Summary	A COMMUNITY OF PRACTICE is an informal network between development teams regarding a special domain. The domain can be technical, agile and more.

Context

Agile working employees may seek for interaction and discussion with like-minded people (cf. [50]). Knowledge management across different development teams must be guaranteed: Knowledge sharing in product-oriented development teams differs from the knowledge sharing in skill-oriented development teams.

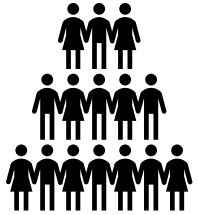
Problem

- **C-5:** *How to facilitate shared context and knowledge?*
- **C-26:** *How to align and communicate architectural decisions?*
- **C-27:** *How to manage and share knowledge about system components and their dependen…*

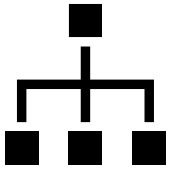
[8] Uludağ et al. (2019)

Exemplary Demonstration of Pattern Candidates

Automotive manufacturer

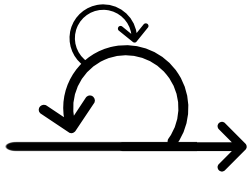


More than 1000 employees



More than 10 teams for developing software

High regulatory standards and norms

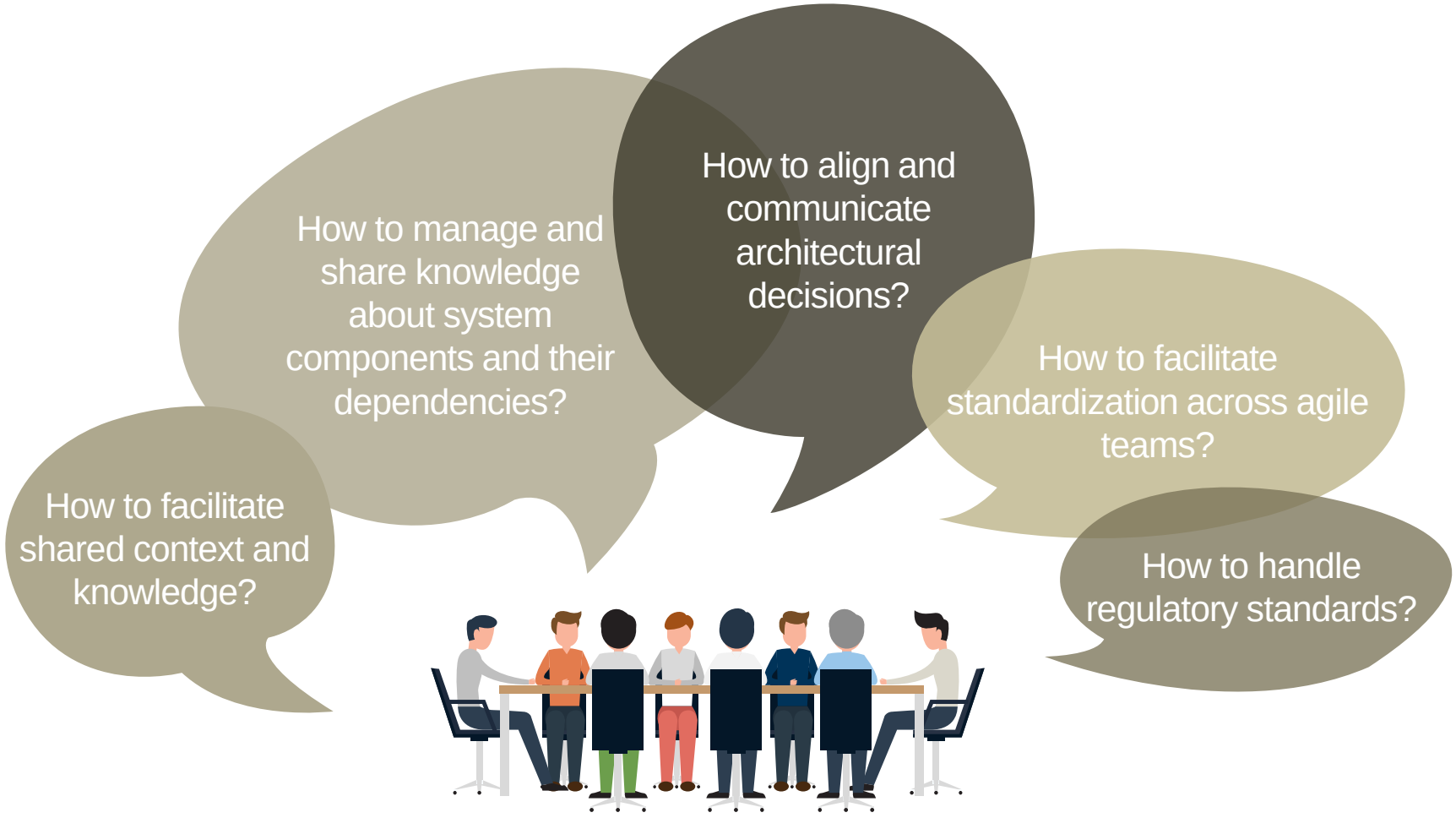


Initial phase of the transformation to agile development

Management meeting with scrum masters

Goal: Improve customer satisfaction and flexibility

Good Practices
Exemplary Demonstration of Pattern Candidates



Management meeting with scrum masters

Good Practices
Exemplary Demonstration of Pattern Candidates

Communities of Practice are an informal network between development teams regarding a special domain (technical or agile). The participation is voluntary for all interested colleagues & they can manage their Community individual.

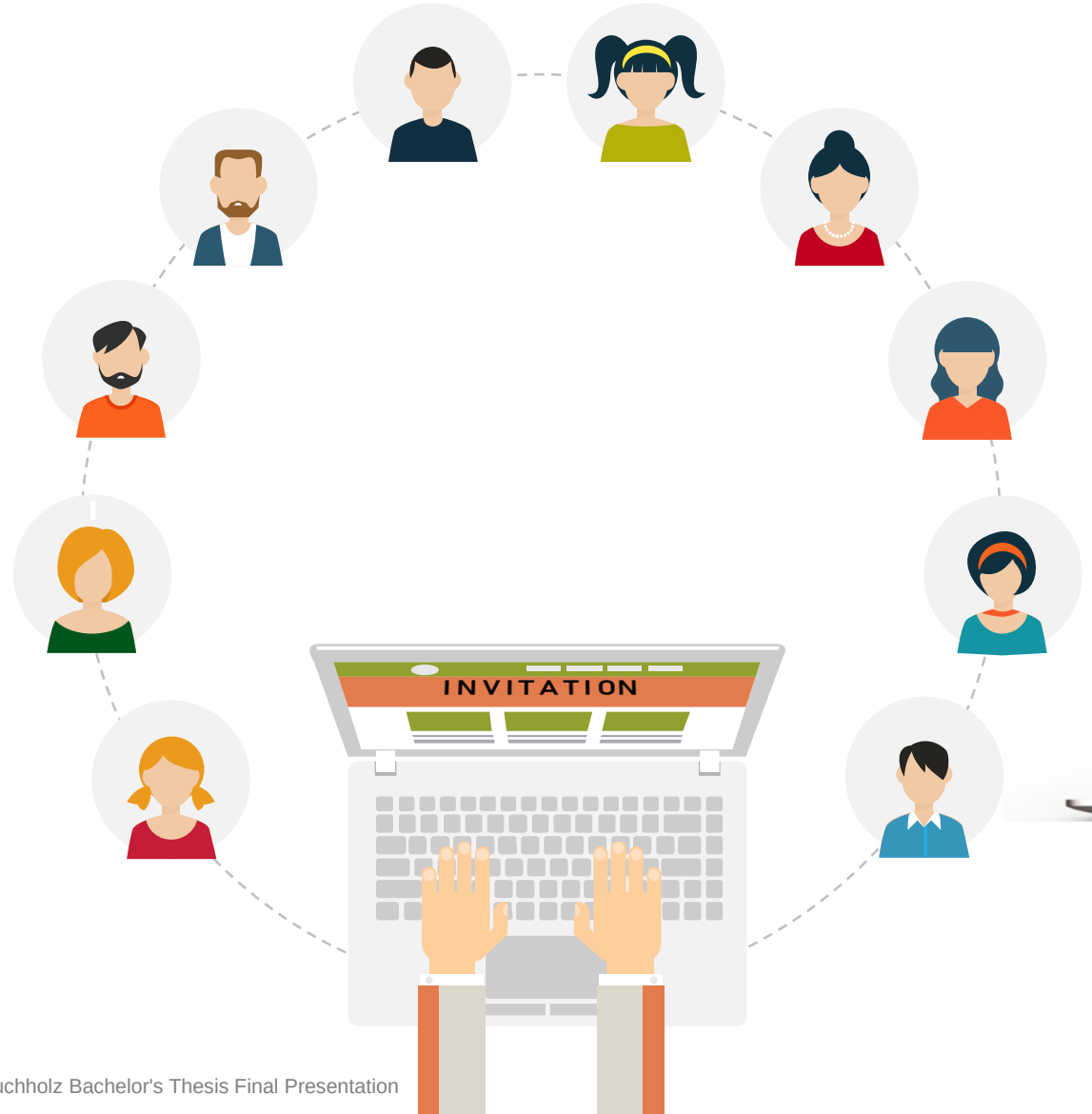
Empowered Communities of Practice have additional rights and regulations. They are obligatory for all selected development teams.

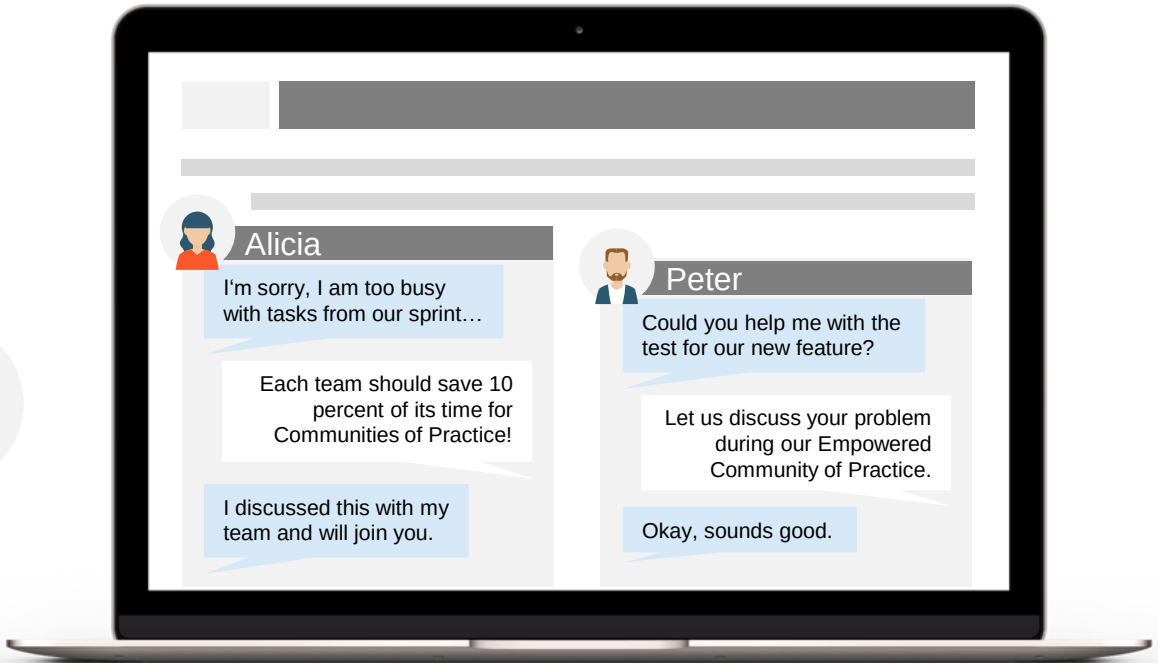
But how do we handle regulatory standards and cost intensive decisions?

What can we do, if nobody joins the community?

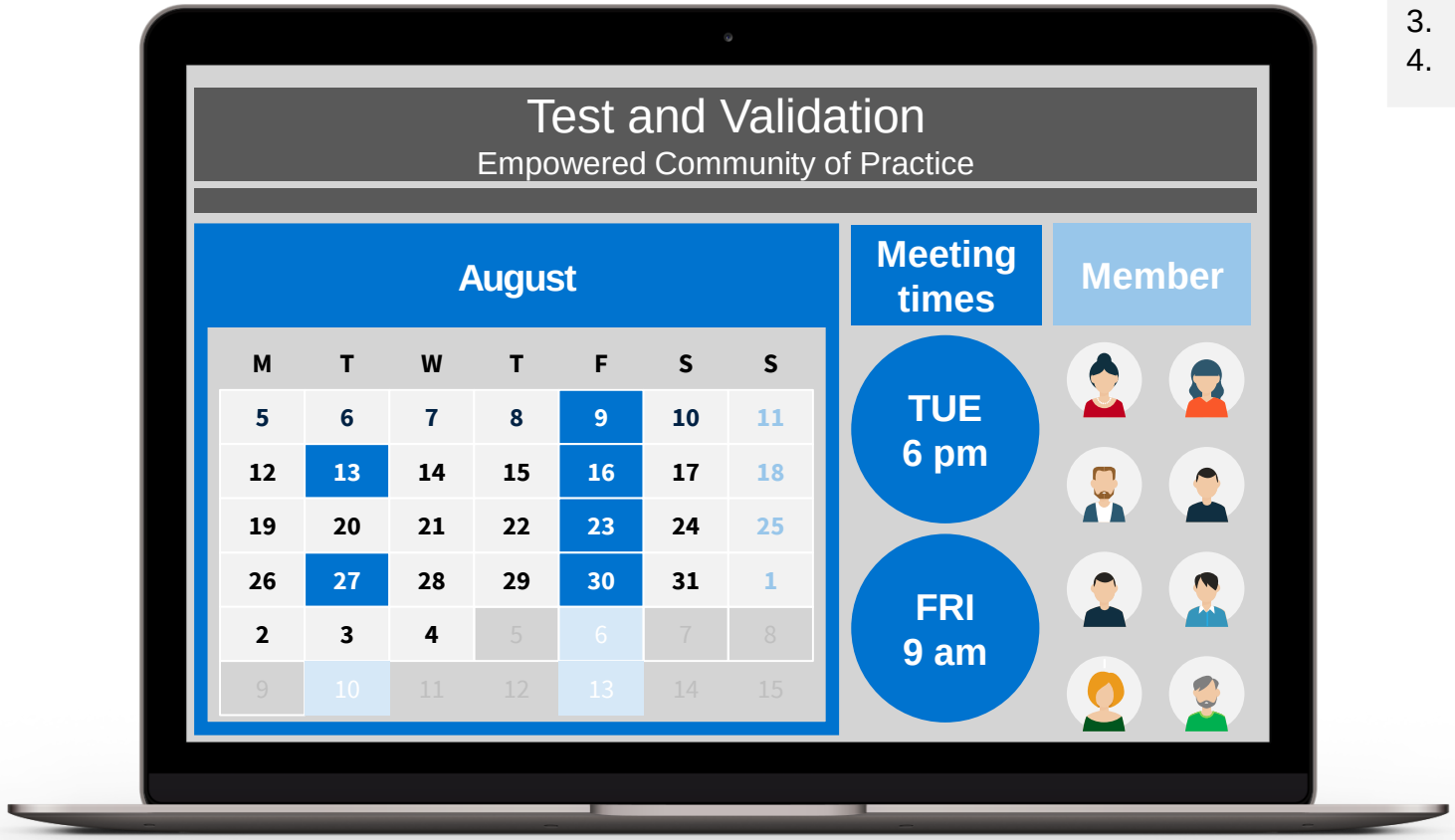
Management meeting with scrum masters

Good Practices
Exemplary Demonstration of Pattern Candidates





Good Practices
Exemplary Demonstration of Pattern Candidates



AGENDA

1. New regulations for brakes
2. Problems with technical maturity
3. On-board system change
4. Test for new feature (Peter)

Outline

1

Introduction

1. Motivation
2. Research Questions
3. Research Approach

2

Case Study

1. Case Study Design
2. Transformation
3. Recurring Concerns
4. Good Practices

3

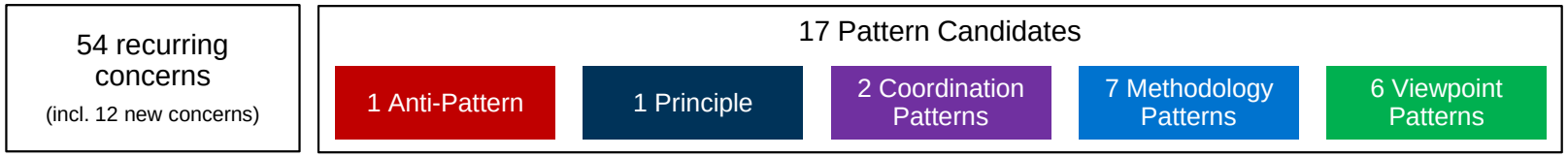
Conclusion

1. Key Findings
2. Future Work

Conclusion

Key Findings

- Main scaling practices: *1. work unit dedicated agility, 2. (Empowered) Communities of Practice, 3. Product Owner Board → Individual transformation*
- Role-oriented view on concerns and patterns



- Time and energy intensive transformation, but improvement in employee satisfaction

Future Work

- Evaluation of identified Pattern Candidates in the department for vehicle dynamics development
- Validation of identified Pattern Candidates in other organizations
- Long-time studies on transformations and the progress of concerns within transformations

References

[1] Papadopoulos, G. (2015). Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. *Procedia - Social and Behavioral Sciences*, *175*, 455–463.

[2] Highsmith, J. A. (2002). *Agile Software Development Ecosystems*.

[3] Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213–1221.

[4] Uludağ, Ö., Kleehaus, M., Caprano, C., & Matthes, F. (2018). Identifying and Structuring Challenges in Large-Scale Agile Development Based on a Structured Literature Review. *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, 191–197.

[5] Lethbridge TC, Sim SE, Singer J (2005). Studying software engineers: data collection techniques for software field studies. Empir Softw Eng 10(3):311–341.

[6] Robson, Colin. (2011) *Real world research*. Vol. 3. Chichester: Wiley.

[7] Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164.

[8] Uludağ, Ö., Harders, N.-M., & Matthes, F. (2019). Documenting Recurring Concerns and Patterns in Large-Scale Agile Development. *Association for Computing Machinery*, *1*(1), 15.

[9] Uludağ, Ö.; Reiter, N.; Matthes, F. (2020): Improving the Collaboration between Enterprise Architects and Agile Teams: A Multiple-Case Study. In: Zimmermann, A.; Schmidt, R.; Lakhmi, C. J. (eds) Architecting the Digital Transformation, Springer-Verlag.

Clara Lea Buchholz

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

clara.buchholz@tum.de
wwwmatthes.in.tum.de

Backup

Case Study - Transformation

Goal:

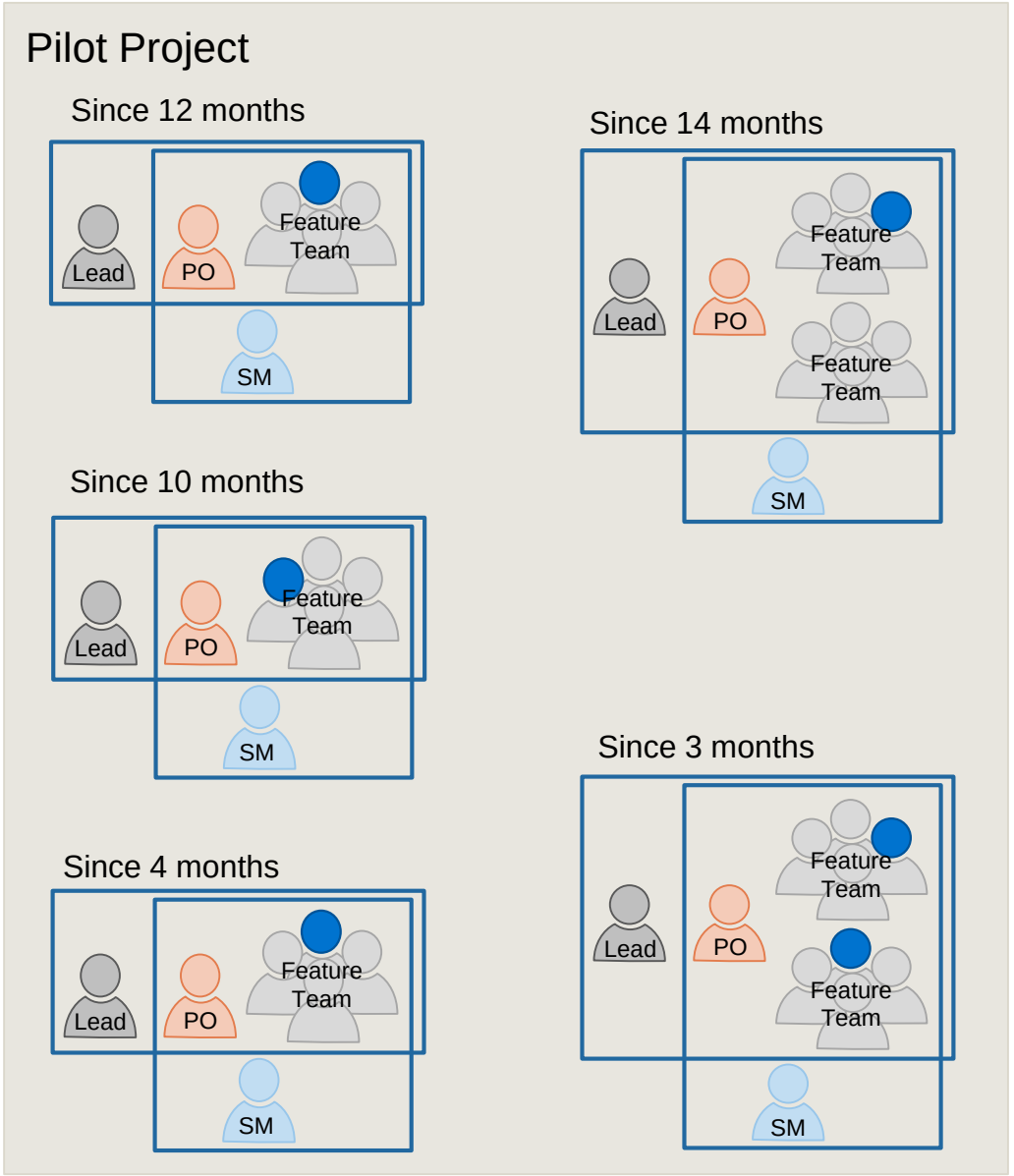
17 Feature Teams

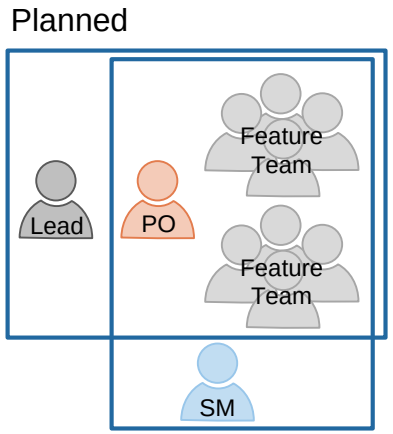
9 Leaders

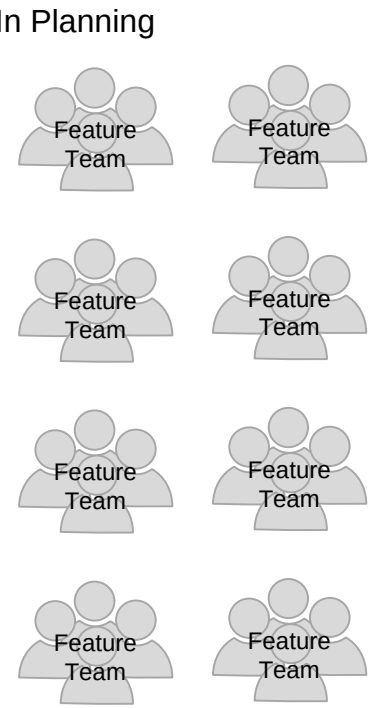
9 Product Owner

8 Scrum Master

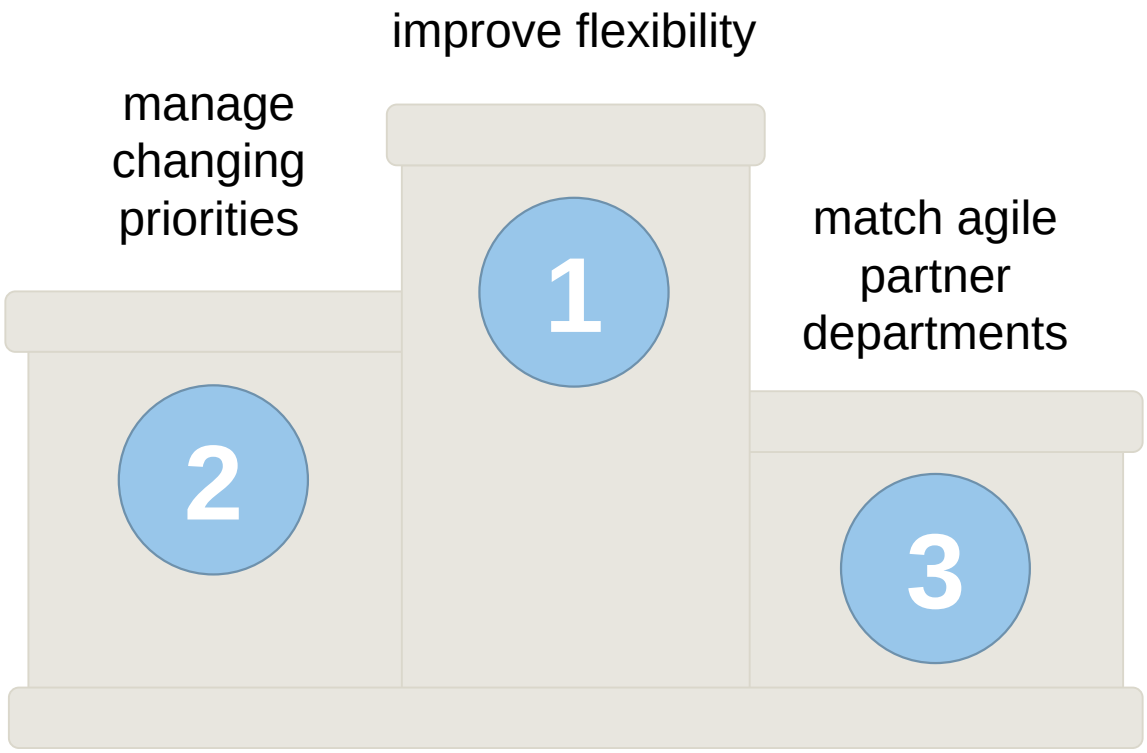
+ several Communities of Practice







Transformation Motivators



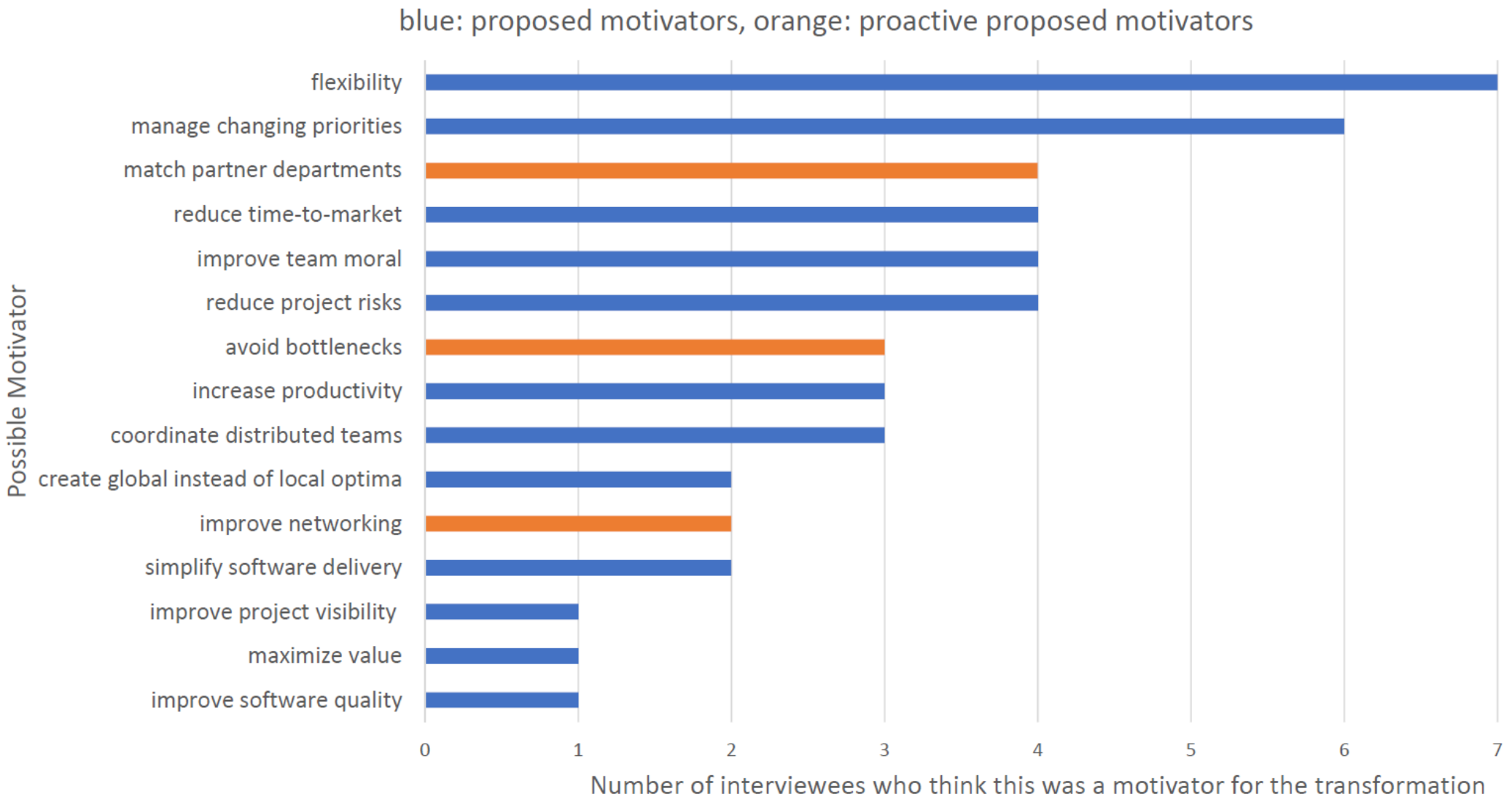
SCRUM MASTER
1. flexibilty
2. coordinate distributed teams
3. manage changing priorities

PRODUCT OWNER
1. manage changing priorities
2. match agile partner departments
3. improve team moral

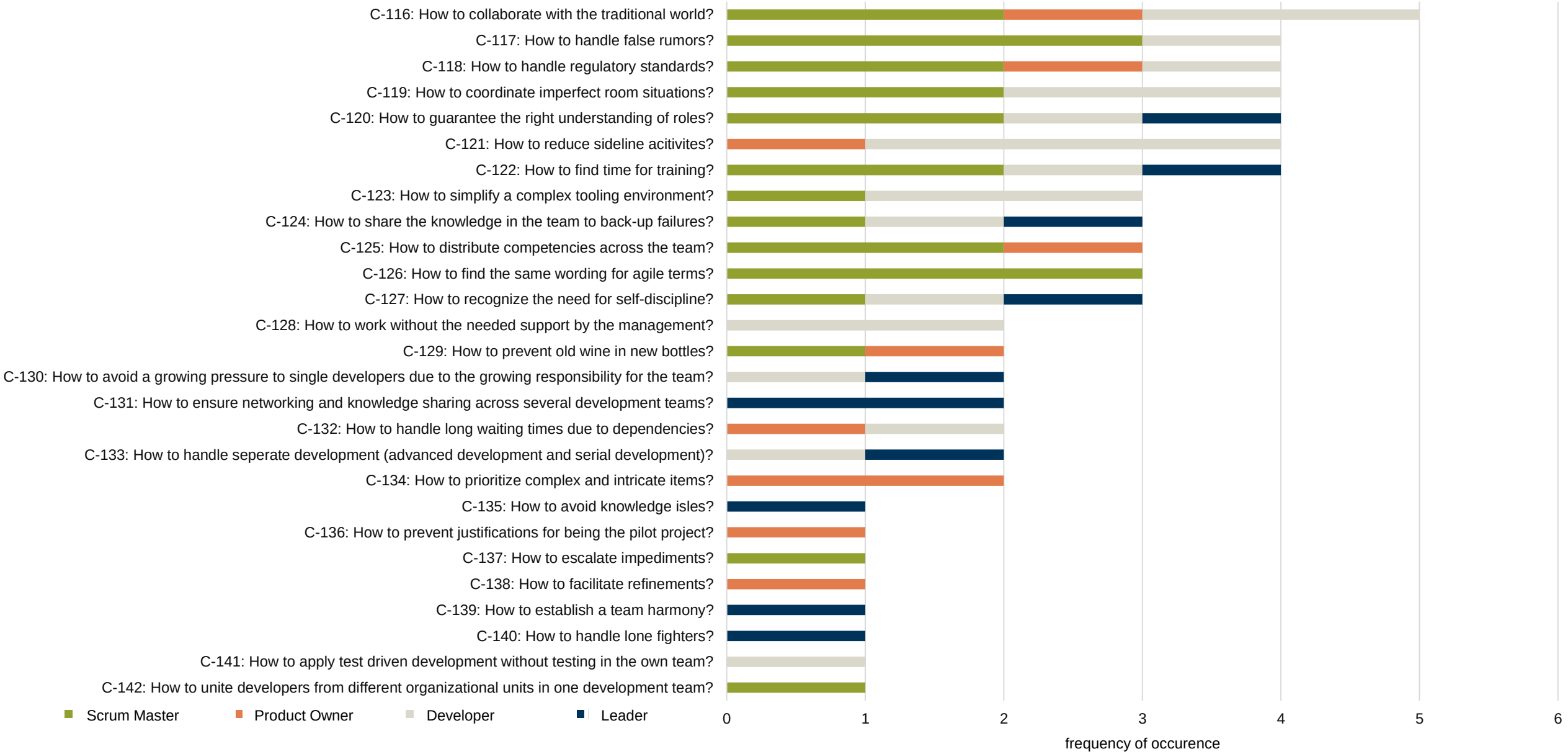
DEVELOPER
1. flexibilty
2. avoid bottlenecks
3. coordinate distributed teams
4. einfacher organisieren

MANAGER
1. flexibility
2. match agile partner departments
3. improve productivity

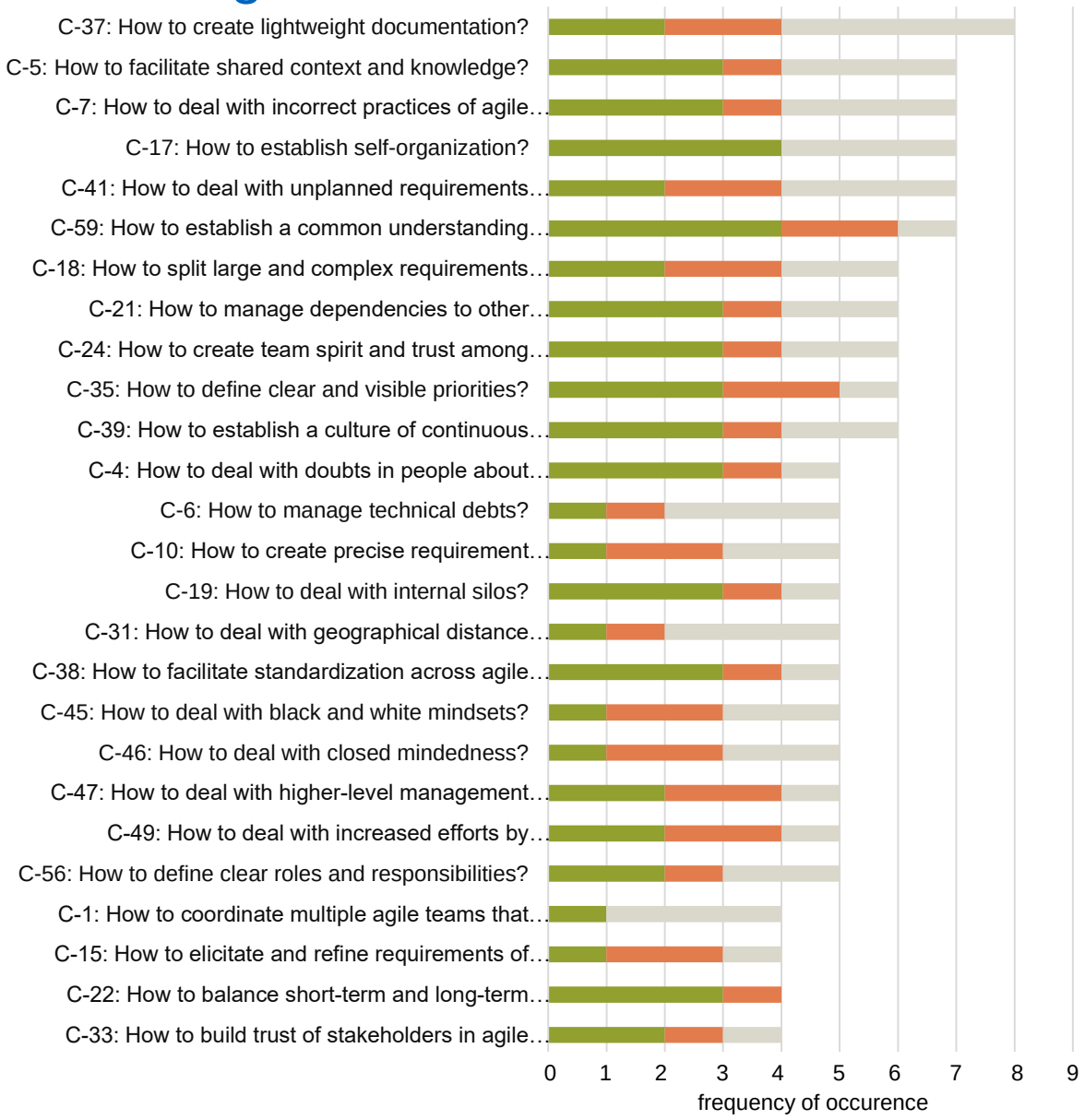
Transformation Motivation

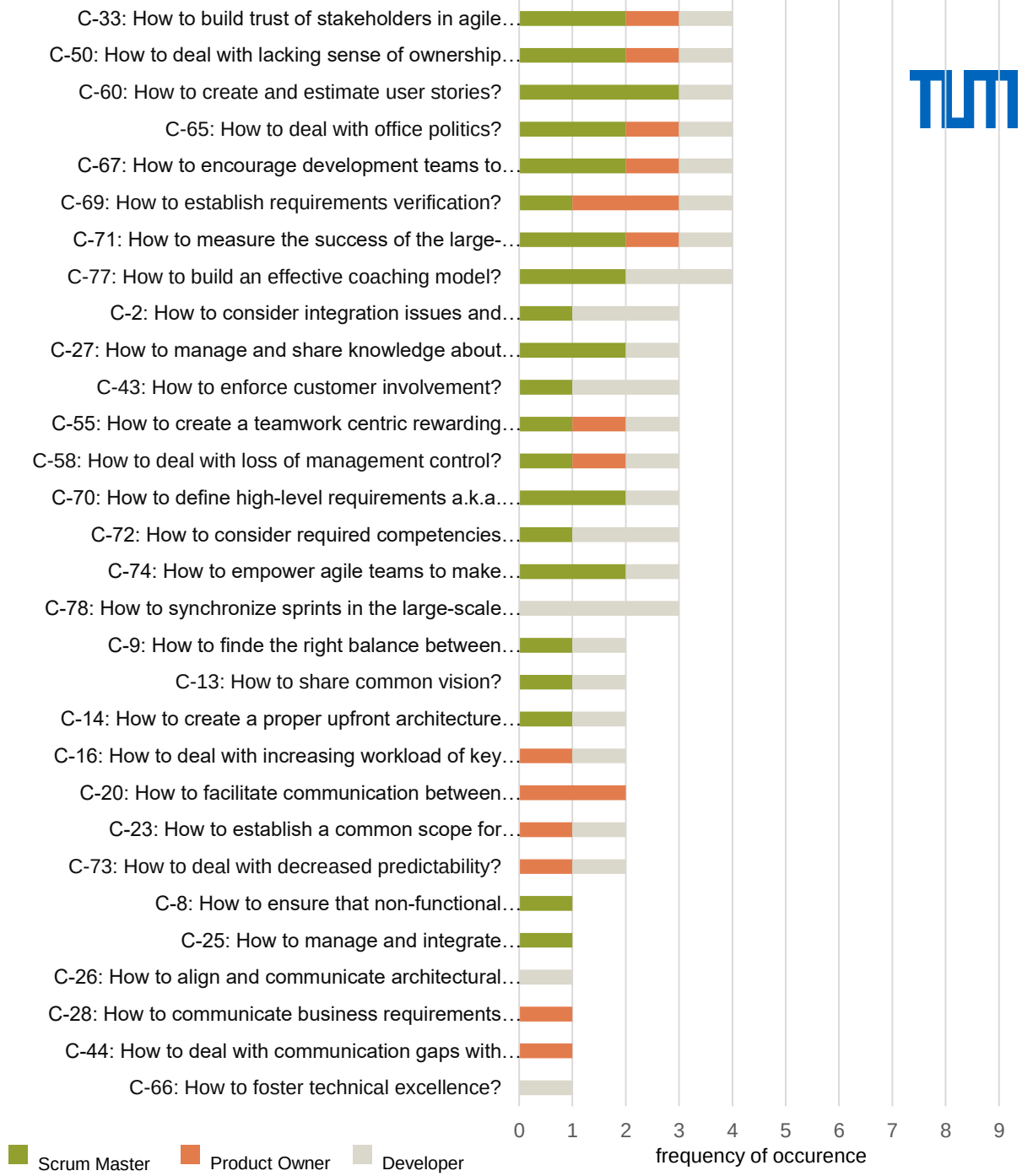


Concerns
New Findings



Recurring Concerns





Conclusion

1 How does the large-scale agile transformation take place at the case organization?

Main scaling practices: 1. special work unit for agility, 2. (Empowered) Comunities of Practice, 3. Product Owner Board
No common framework was used

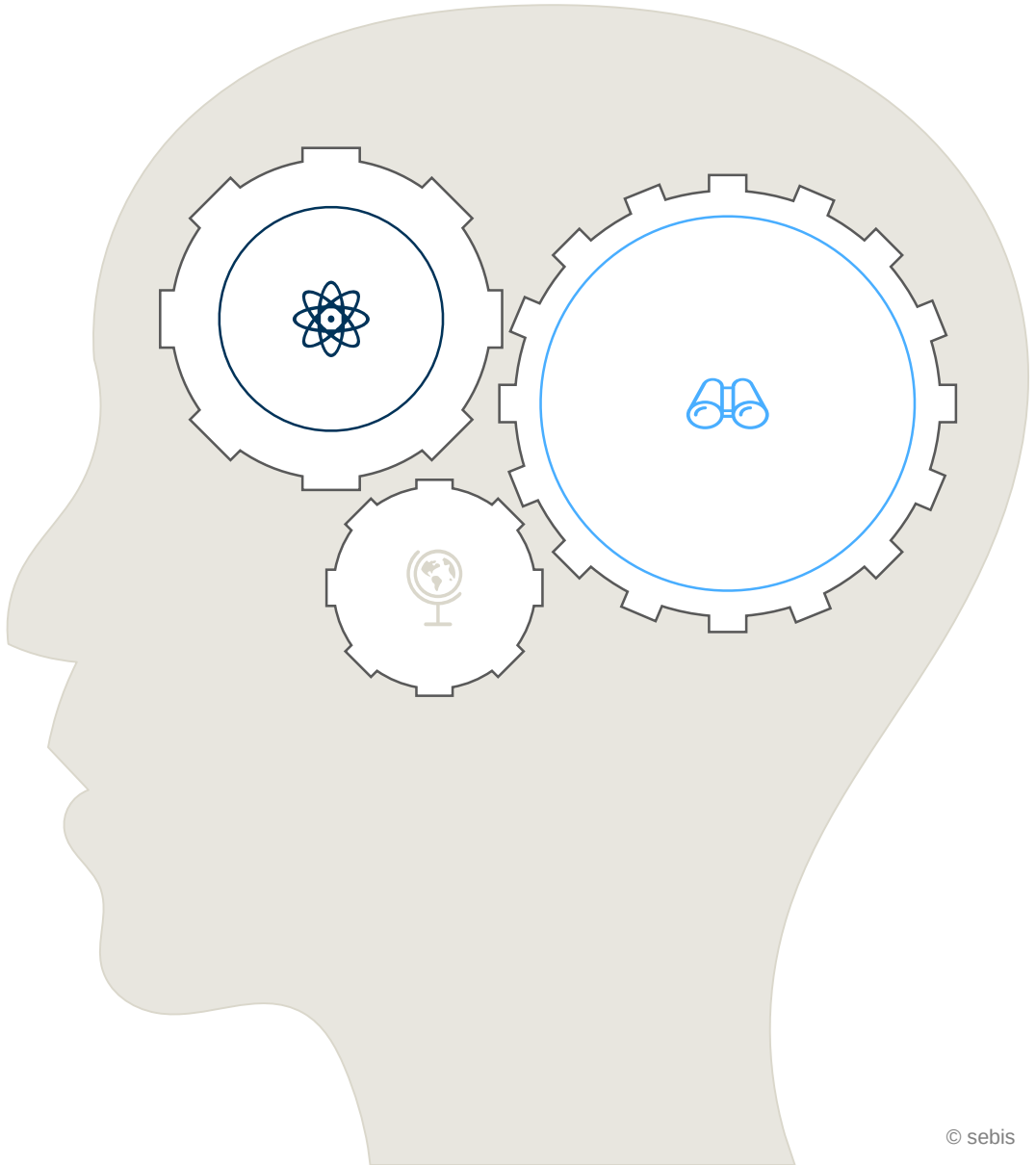
2 What are challenges and success factors within the large scale agile transformation at the OEM?

26 new concerns
+ 55 concerns found again

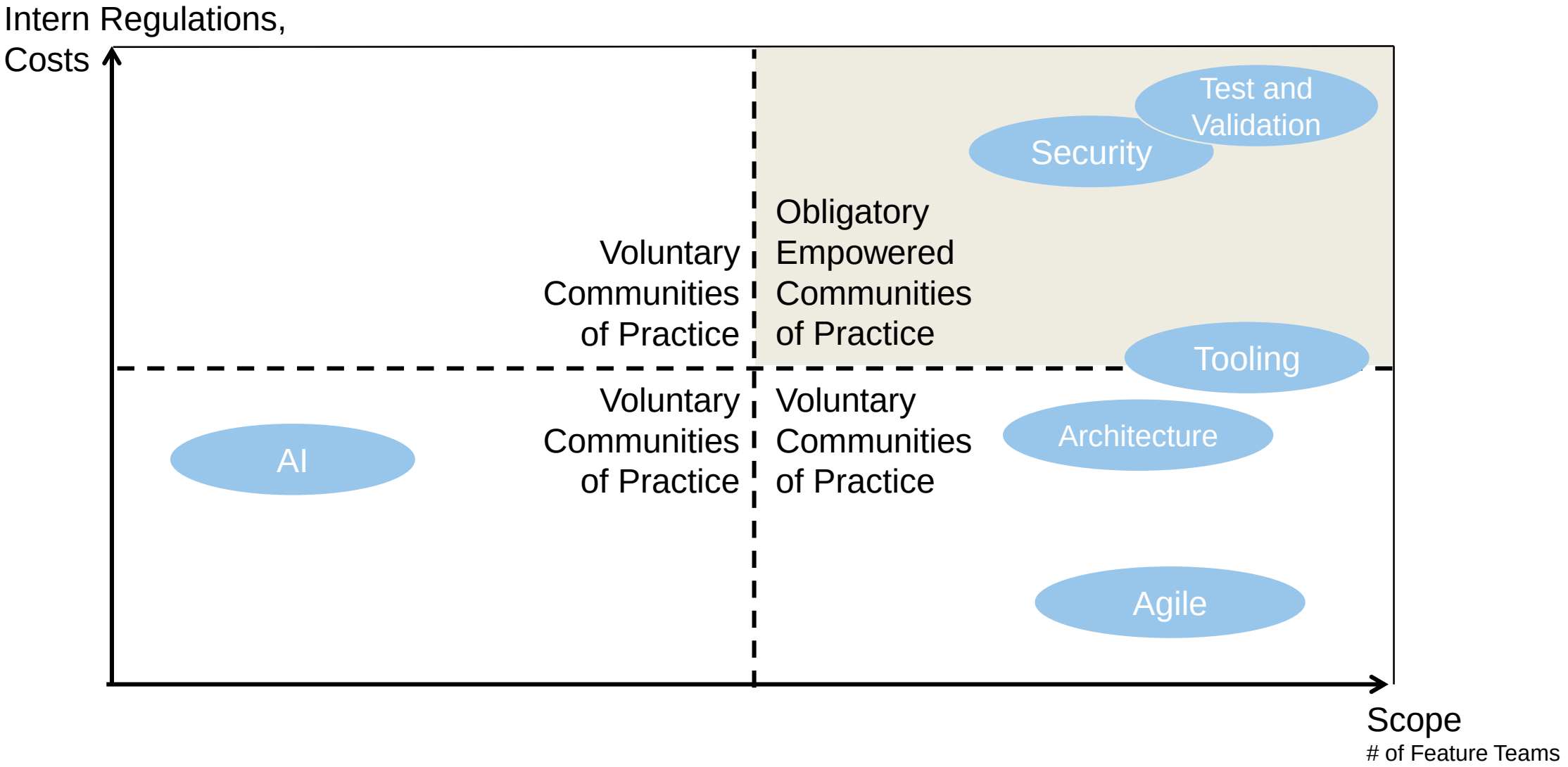
3 What are good practices to address the observed challenges within the large scale agile transformation at the OEM?

17 Pattern Candidates:

- 1 Anti-Pattern
- 1 Principle
- 2 Coordination Patterns
- 7 Methodology Patterns
- 6 Viewpoint Patterns



Communities of Practice
in the Agile Architecture Decision-Making Model [9]



[9] Uludag, Ö.; Reiter, N.; Matthes, F. (2020)

CO-01
Community of Practice

Pattern Overview	
ID	CO-01
Name	COMMUNITY OF PRACTICE
Alias	*none*
Summary	A COMMUNITY OF PRACTICE is an informal network between development teams regarding a special domain. The domain can be technical, agile and more.

Context

Agile working employees may seek for interaction and discussion with like-minded people (cf. [50]). Knowledge management across different development teams must be guaranteed: Knowledge sharing in product-oriented development teams differs from the knowledge sharing in skill-oriented development teams.

Problem

- **C-5:** *How to facilitate shared context and knowledge?*
- **C-26:** *How to align and communicate architectural decisions?*
- **C-27:** *How to manage and share knowledge about system components and their dependencies?*
- **C-38:** *How to facilitate standardization across agile teams?*

Forces

Colleagues must be motivated to work for COMMUNITIES OF PRACTICE. If the participants do not use the COMMUNITY OF PRACTICE or not enough people participate at all the COMMUNITY OF PRACTICE makes no sense.

Solution

A COMMUNITY OF PRACTICE can be founded by every interested person regarding a specific topic. Everyone can be part of a community, the participation is voluntary. Usually the members of a COMMUNITY OF PRACTICE are from different development teams. A community is no team and consequently doesn't work with an own backlog. Tasks are solved directly together or handed over to development teams for the solving. The participants meet frequently and can have different kinds of meetings: workshops, work meetings, stand ups, pitches, and more. Also informal meetings off the job are possible, especially when the domain is not directly regarding the job. The COMMUNITY OF PRACTICE Coordinator organizes the meetings. A COMMUNITY OF PRACTICE works well for knowledge-management and networking across feature teams. But COMMUNITIES OF PRACTICE are not authorized to reach decisions automatically effective for all development teams (cf. [50]).

Consequences

Benefits:

- Given foundation for networking.
- Higher transparency, because topics are discussed in the open COMMUNITY OF PRACTICE, instead of closed circles.
- Facilitates knowledge management, because all interested colleagues come together.
- Solve problems with the help of others.
- Broad knowledge base open for everyone.

Liabilities:

- Time intensive, because of additional meetings (Solution: Save time in every sprint for Community-Work).

See also

M-01 EMPOWERED COMMUNITY OF PRACTICE

M-01
Empowered Community of Practice

Pattern Overview	
ID	M-01
Name	EMPOWERED COMMUNITY OF PRACTICE
Alias	Core Community of Practice
Summary	Additional to COMMUNITIES OF PRACTICE the EMPOWERED COMMUNITY OF PRACTICE is obligatory for all selected development teams and enjoys additional rights and regulations.

Context

COMMUNITIES OF PRACTICE may not have enough power to justify far-reaching decisions. Regulatory Standards suffer for consistent and powerful instruments to be enforced. Also COMMUNITIES OF PRACTICE with required but tedious topics may have not enough participants.

Problem

- **C-5:** *How to facilitate shared context and knowledge?*
- **C-26:** *How to align and communicate architectural decisions?*
- **C-27:** *How to manage and share knowledge about system components and their dependencies?*
- **C-38:** *How to facilitate standardization across agile teams?*
- **C-118:** *How to handle regulatory standards?*

Forces

Usually COMMUNITIES OF PRACTICE work democratic, this is not possible when regulatory and cost intensive decisions are needed. These require responsibility of executives. Especially the role of the Empowered Community leader must be picked wisely.

Solution

To decide whether a community is a COMMUNITIES OF PRACTICE or an EMPOWERED COMMUNITIES OF PRACTICE the agile architecture decision-making model by Uludağ et al. [70] (see Figure 5.6) can be used. Topics with a wide scope (number of development teams influenced by the topic) and high costs or regulations are Topics for a Empowered

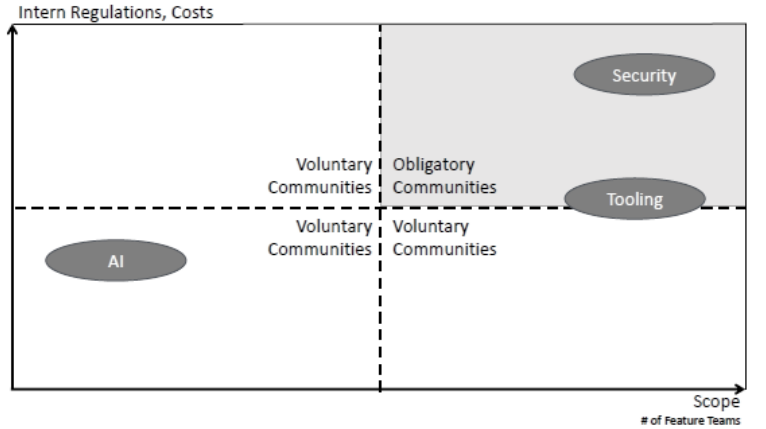


Figure 5.6.: Decision making model process with example domains [70]

Community of Practice, while all others are for usual Communities of Practice.

Usually EMPOWERED COMMUNITIES OF PRACTICE work democratic. Some regulatory and cost intensive decisions require responsibility of executives and therefore work not democratic. The role of the Empowered Community leader must be picked wisely. The leader is an executive. He coordinates the Community and is empowered to take decisions valid for all development teams.

Every selected development teams sends at least one developer to the EMPOWERED COMMUNITIES OF PRACTICE. The participation is obligatory. EMPOWERED COMMUNITIES OF PRACTICE can help to guarantee the compliance of the intern regulations and standards as well as knowledge management in a feature oriented structure.

Consequences

Benefits:

- Participation of all development teams in the decision making process is guaranteed.
- Guarantee networking through all feature teams.
- Guarantee base for knowledge management for specific domains.
- Improves quality of valid standards (experts are included in the decision making and standards are valid for the whole department).

Liabilities:

- Obligatory activities mostly have a lower motivation rate.
- Time intensive because more participants are required and long discussions can appear.
- The leader has more power, so participants can feel unequal.

See also

CO-01 COMMUNITY OF PRACTICE

M-02
Piloting

Pattern Overview	
ID	M-02
Name	PILOTING
Alias	*none*
Summary	To ascertain the environment for agile development, gain first experience and experiment a new idea a pilot project is started.
Type	Communication
Binding Nature	Mandatory

Context

To test how agile methods can be adapted at the case organization sample teams start to work agile.

Problem

C-81: *How to enable change from process to product orientation?* (found by Uludağ and Matthes [69])

Forces

If the experiment fails, new pilots need to start. Additional pilots often need to justify in their department, for working different and creating additional work to others without knowing whether it will improve their development.

Consequences

Benefits:

- Early experience by pilots can be used for the transformation
- Pilot can be used as a training by using SHADOWING (**M-06**)
- Fail fast, fail cheap (cf. [33])
- Successful experiments motivate developer

Liabilities:

- Time intensive
- Failing experiments can demotivate developer
- **C-136:** *How to prevent justifications for being the pilot project?*

M-03
Travelling

Pattern Overview	
ID	M-03
Name	TRAVELLING
Alias	Personnel Exchange
Summary	Developer change their team for one sprint length to get deep insights of the technical and agile work of another development team.

Context

Single developers want to gain knowledge about the technical work of another team and to get insights in their agile work.

Problem

- **C-7:** *How to deal with incorrect practices of agile development?*
- **C-38:** *How to facilitate standardization across agile teams?*
- **C-59:** *How to establish a common understanding of agile thinking and practices?*
- **C-122:** *How to find time for training?*

Forces

If the regular development team and the visited development team do not have the same sprint length and the time of the sprint end is not the same, the traveler has an overload of work in between the times. Both teams may recognize a lack of competencies in their team during this sprint. On the other hand especially for the planned tasks in the sprint experts can travel to support the team.

Solution

Two developer can change positions for one or more sprints. The travelling is usually organized by the developer itself or the scrum master.
During the sprint developer A and B do not work for their original teams, but just for their travelling-team. After the sprint they organize one meeting together with all members of both development teams (in the example above all members from development team alpha and beta) and present their experience. Especially the experienced differences between both teams regarding the agile work, advantages and disadvantages they discovered and their lessons learned.

Example: Developer A from team Alpha and developer B from team Beta do not work directly together normally. Their scrum master propose them to change teams for one sprint length to become more familiar with the contents of the other one. They decide to take the chance and in one sprint they change teams to get insights in the work of the other one. That means for the duration one sprint A develops in team Beta and B works in team Alpha. A is part of all meetings of the Beta team, while B is part of all meetings of the Alpha team during this time. After the sprint they write down their experience and organize a meeting with the team members of the Alpha as well as of the Beta team. Also the product owners and the managers are invited. In the meeting they present their experience and give advice how the teams could change parts of their work and try learning from the other team.

Variants

- Developer can particular be sent to another team to support during hard times without having an exchange partner.
- Travelling can also be for longer, for example if developer are bored by their own work and want to get insights into another department.

Consequences

Benefits:

- Gain of knowledge about other teams technical work.
- Discover agile work of other development teams.
- Understand how different work methods can be adapted.

Liabilities: *None known.*

See also

M-06 SHADOWING

M-04

Pattern Overview	
ID	M-04
Name	SHARE THE CHANGE
Alias	*none*
Summary	Involve all employees and stakeholder and communicate the process, goal and motivation of the change during a workshop

Context

During a transformation work routine for the employees and stakeholder change. To include all employees in the change a well-organized workshop with time for discussion helps the employees to get into the transformation.

Problem

- **C-4:** *How to deal with doubts in people about changes?*
- **C-59:** *How to establish a common understanding of agile thinking and practices?*
- **C-116:** *How to collaborate with the traditional world?*
- **C-117:** *How to handle false rumors?*
- **C-128:** *How to work without the needed support by the management?*

Forces

It takes time for stakeholders to understand the change and to find their new place in the organization (contact person, ...). Also it is cost and time intensive to inform and include all employees in the change in a large organization. The performance of all workshops takes time and needs personal capacities.

Solution

Workshop series organized by a team of employees involved in the team with the direct assistance of the (higher) management. The workshop should collect all stakeholders and motivate them to be part of the transformation. Even stakeholders which are not directly involved in the change should be informed at the workshop to get a higher transparency. Possible content for the workshop:

- Show motivation for the change (why do we need a transformation?);
- who organized the transformation;
- what will change;
- what already changed;
- what will not change;
- vision;
- explain new work methods simply to the participants (to all participants, also the ones which will not work with the new work methods);
- deviate a common roadmap.

Consequences

Benefits:

- Everyone is involved
- Higher motivation
- Transparency
- Promote dialog

Liabilities:

- Cost intensive.
- Time intensive.

M-06

Pattern Overview	
ID	M-06
Name	SHADOWING
Alias	Following, See and Go
Summary	To get insights in the work of others, one colleague follows another one to meetings like a shadow and observes the colleagues work.

Problem

- **C-5:** *How to facilitate shared context and knowledge?*
- **C-47:** *How to deal with higher-level management interference?*

Solution

Interested colleagues can follow special related roles in the agile work to understand their way of working. For example managers follow the scrum master or product owner. Inexperienced developer follow the members of a development Team. Stakeholder follow their partner. The person following acts like a shadow and observes meetings, without interrupting the meeting. Teams should not feel disturbed so it is important, that the "Shadows" doesn't interact with the teams during the meeting. The person which is followed can explain open topics to the "Shadow" if needed after the meetings.

Variants

Experienced colleagues (especially scrum master and product owner) can use shadowing to help inexperienced scrum masters and product owner to improve their work in the initial phase of the transformation.

Consequences

Benefits:

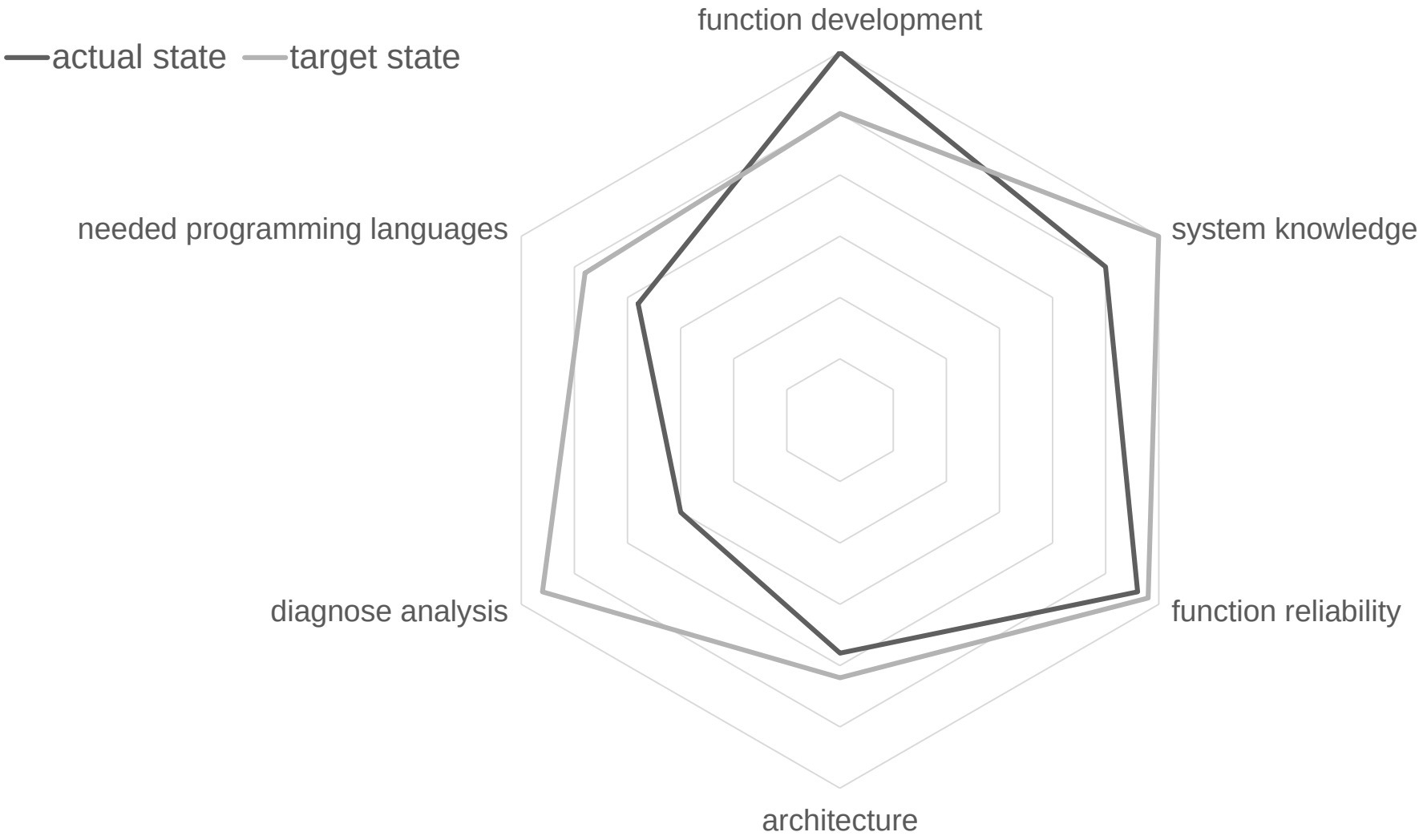
- Insights in the work without disturbing a team

Liabilities: *None known.*

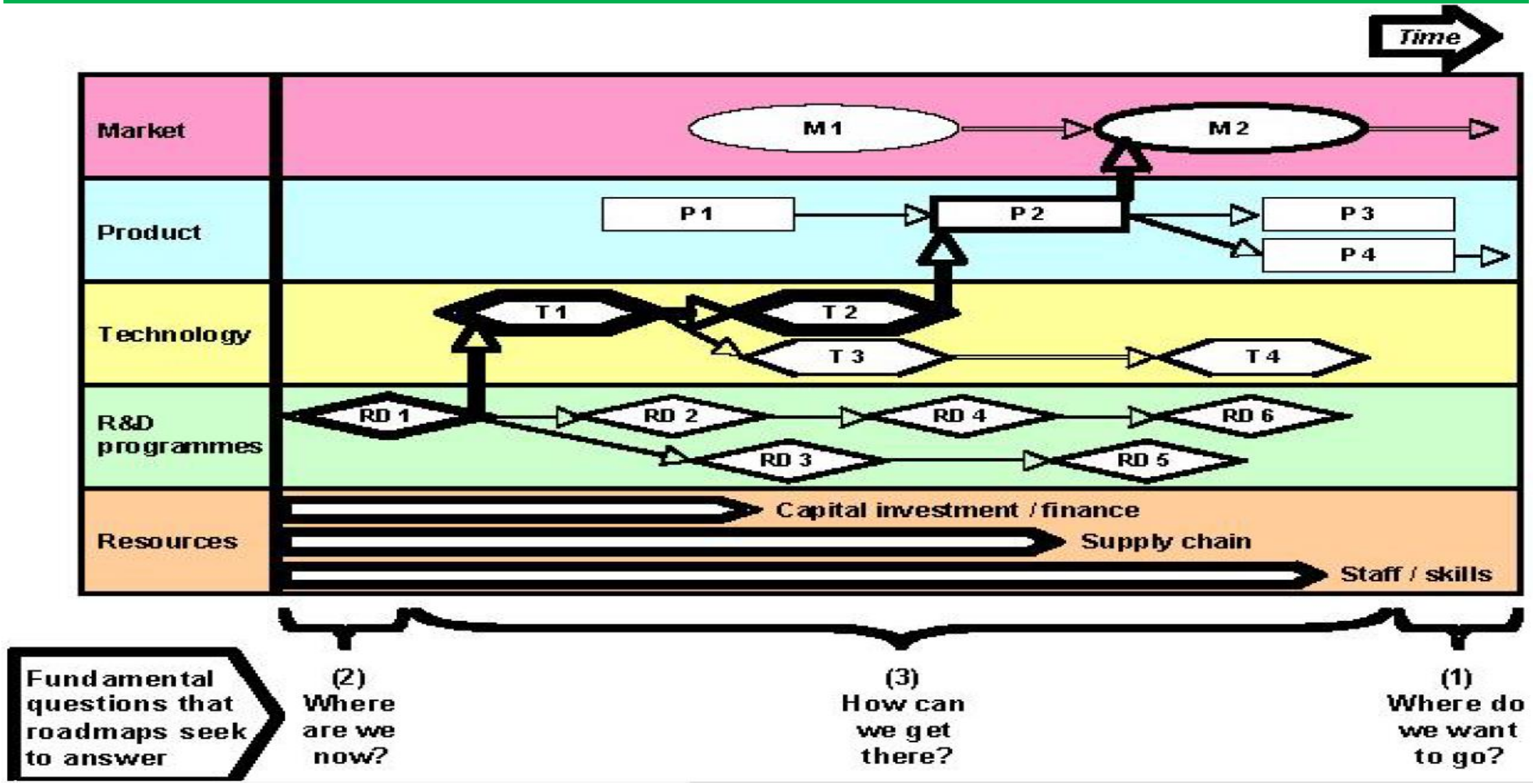
See also

M-05 COME TO OUR DEMOS

V-01 Radar chart

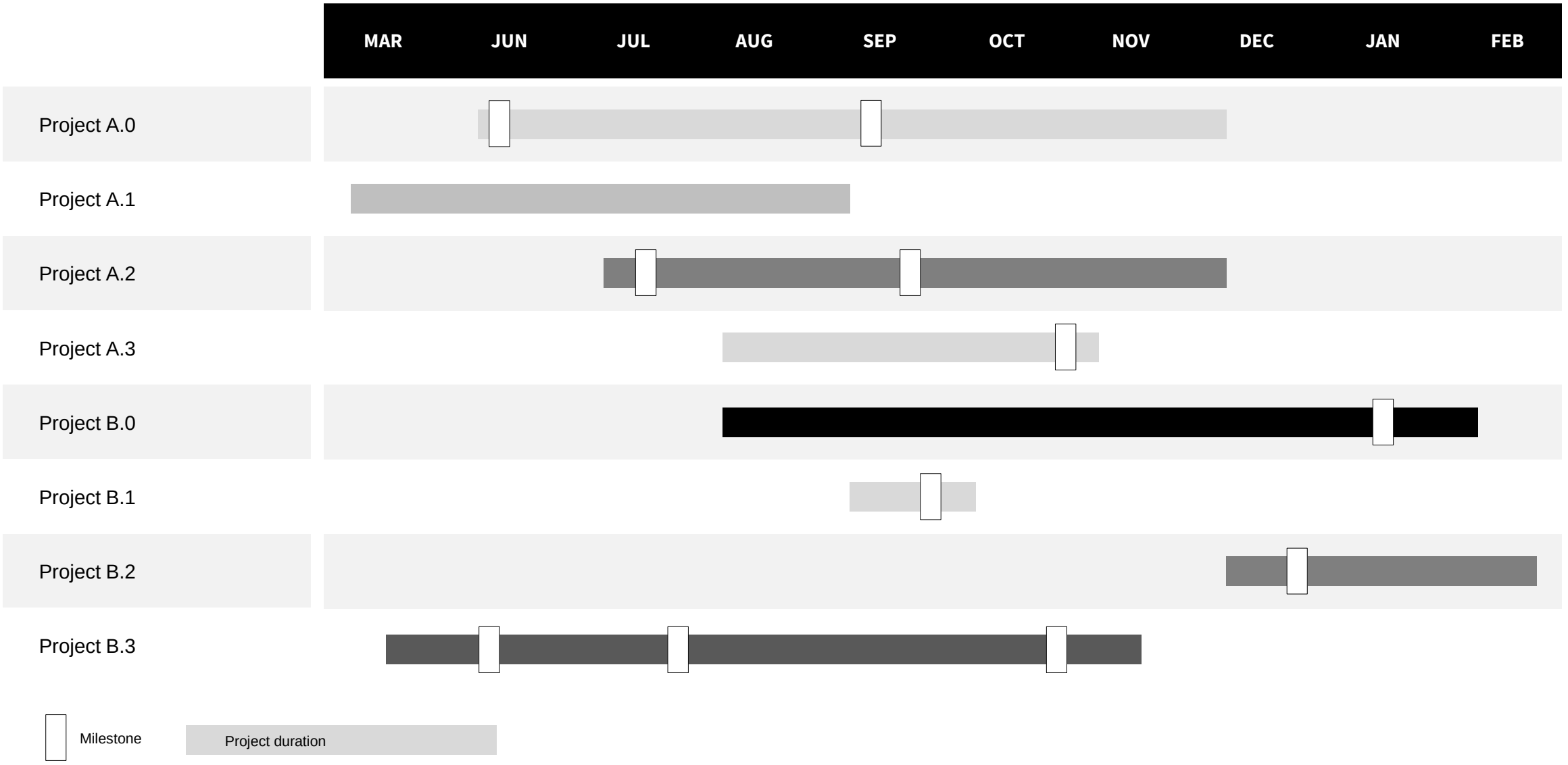


V-02 Roadmap

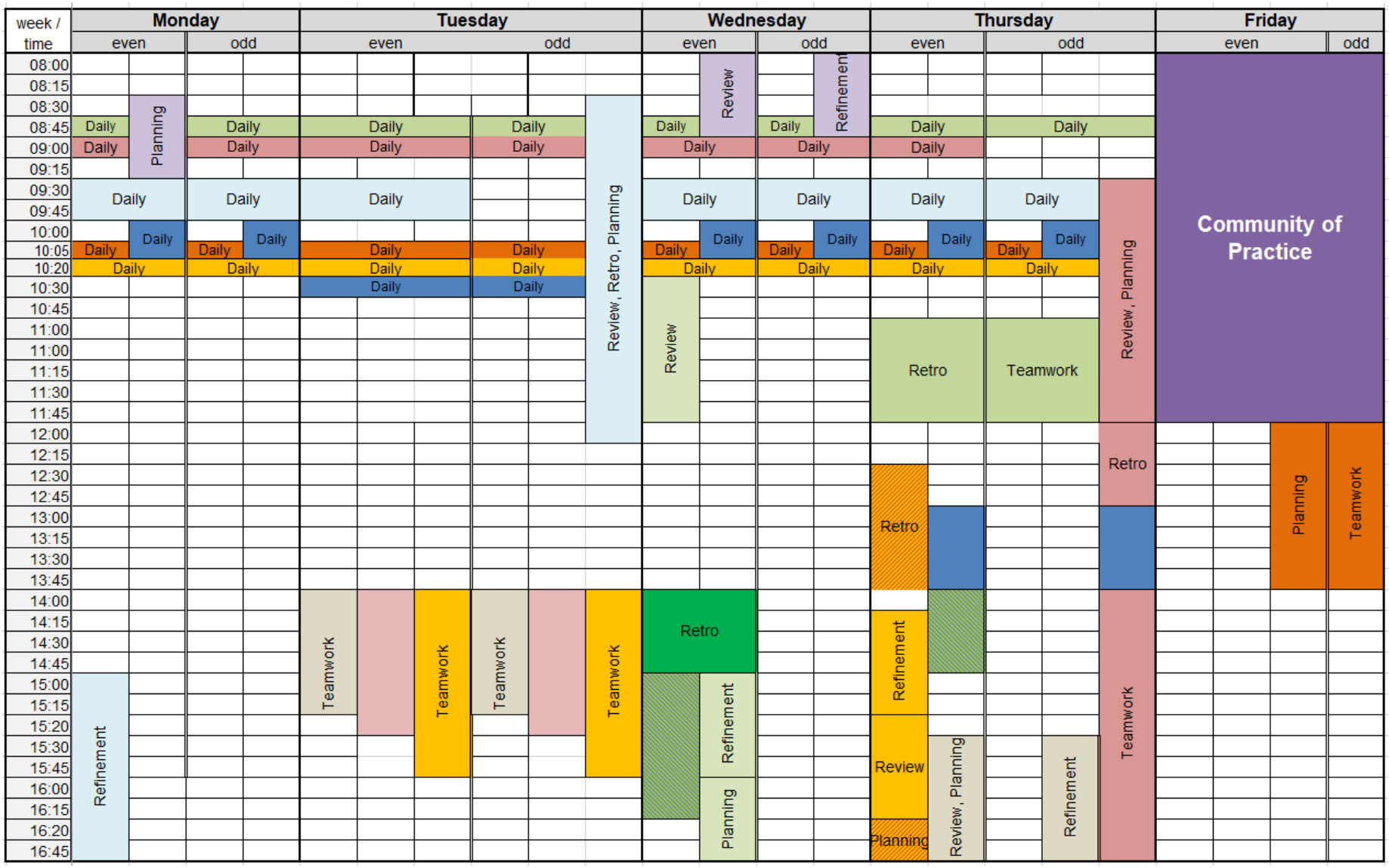


Phaal, R., & Muller, G. (2007). Towards Visual Strategy: An Architectural Framework for Roadmapping. *PICMET '07 - 2007 Portland International Conference on Management of Engineering & Technology*, 1584–1592. https://doi.org/10.1109/PICMET.2007.4349483

V-03 Project plan



V-04 Synchronized calendar



V-04
Synchronized Calendar

Pattern Overview	
ID	V-04
Name	SYNCHRONIZED CALENDAR
Alias	One Calendar Approach, Synchronized Sprints
Summary	Several development teams share their frequent meeting times and locations to facilitate coordination with other development teams.

Context

To facilitate SHADOWING (**M-06**), coordinate COMMUNITIES OF PRACTICE (**CO-01**) or EMPOWERED COMMUNITIES OF PRACTICE (**M-01**), share meeting rooms, allow TRAVELLING (**M-06**), and evolve transparency a shared calendar can be used.

Problem

- **C-78:** *How to synchronize sprints in the large-scale agile development program?*

Forces

The calendar should be always up to date, so every team is responsible to update the calendar. Responsibility for the sharing in the different development teams, but the teams may not be prioritize the updating of the calendar so the coordination can be hard. The room management may fail if not enough conference rooms are available and less time slots for meetings, especially longer ones can be offered.

Solution

All development teams (working with one product) share their meeting times (including locations) in an accessible calendar and update the calendar if data changes. Every color represents one team. All teams work in sprints with a duration of two weeks, so meeting dates to sprint change times are shown by dividing the table in even and odd calendar weeks.

Variants

- If every development team has its own mailbox, they can also share their calendar through outlook instead of a table.
- The synchronizing of sprints through a shared calendar adds more value to the communication through the sprint. Several teams can synchronize their sprint length and sprint change times to increase their cooperation. The organization can be done central or decentral, depending on the preferences of development teams and the management. Central organization facilitates the room management and the time management for corporations, as Communities of Practice, Travelling, and more.
- Additional match sprint times (beginning, end, and duration) with supplier.

Consequences

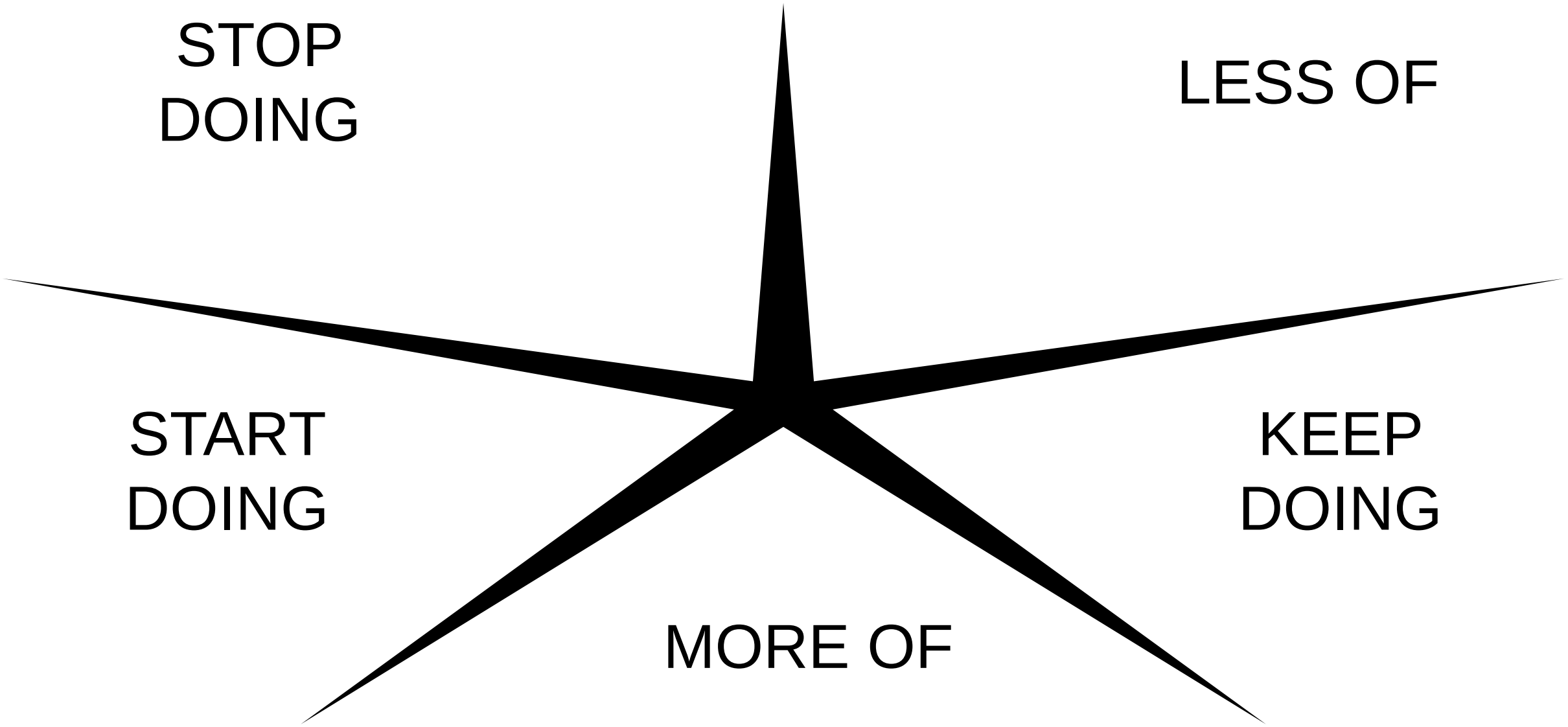
Benefits:

- Transparency
- Fast information for stakeholder and interested members

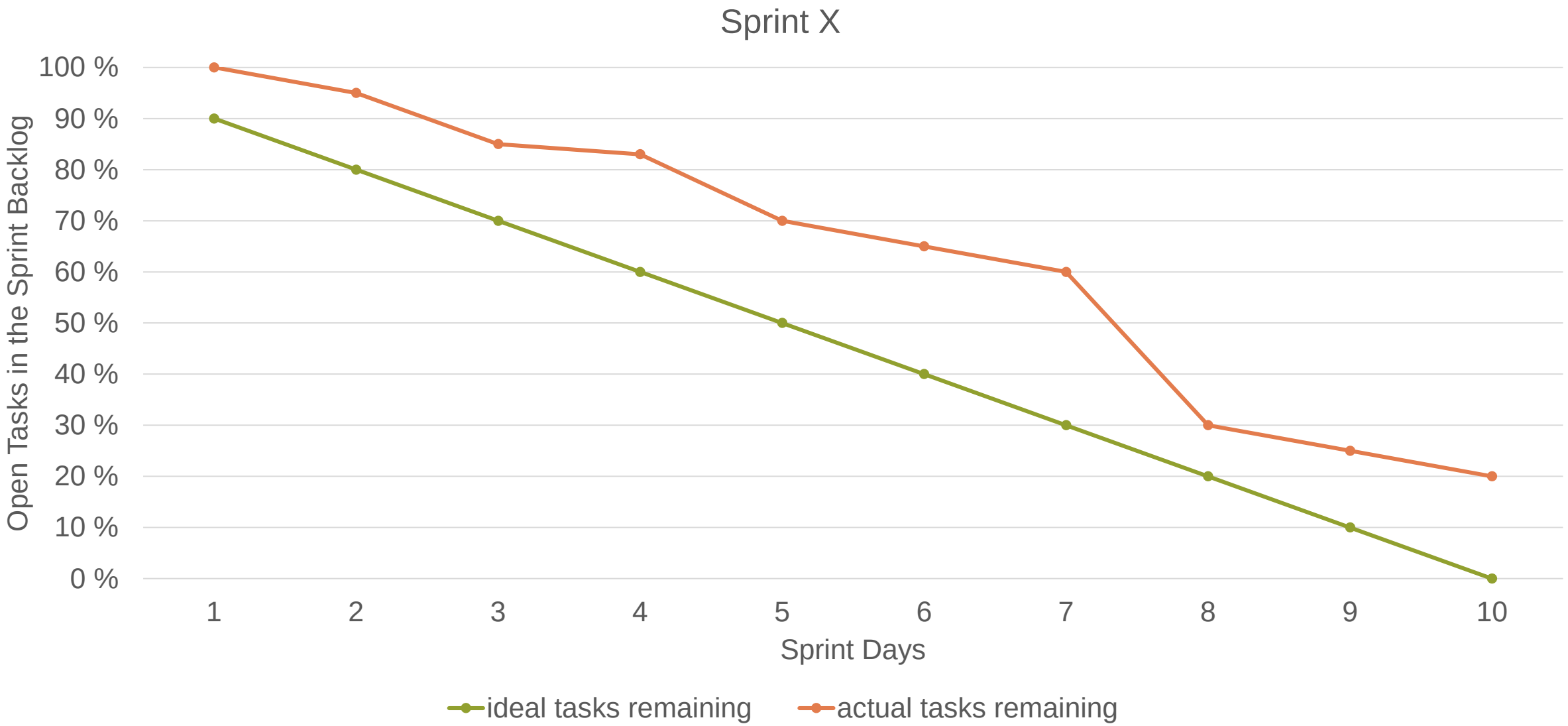
Liabilities:

- Responsibility for the care of the calendar.

V-05 Starfish



V-06 Burndown chart



Overview Related Work

Author	Publication date	Type	Deliverables
Papadopoulos [1]	2015	case study	practices and challenges especially during the adoption
Paasivaara and Lassenius[1*]	2016	case study	scaling practices
Dikert, Paasivaara and Lassenius [2*]	2016	literature review	35 challenges and 29 success factors
Kalenda, Hyna and Rossi [3*]	2018	literature review followed by a case study	several practices, 9 success factors and 10 challenges
Uludağ, Kleehaus, Capano and Matthes [4]	2018	literature review	14 stakeholder roles and 79 challenges (for their pattern language)
Fuchs and Hess [4*]	2018	case study	evolution of challenges and barriers through several phases of the transformation
Uludağ, Harders and Matthes [8]	2019	empirical research	concerns and patterns which extend their pattern language
Uludağ and Matthes [5*]	2019	pattern-based design research	concerns and pattern (related to agile coaches and scrum masters) which extend their pattern language

References

[1*] Paasivaara, M., & Lassenius, C. (2016). Scaling Scrum in a Large Globally Distributed Organization: A Case Study. *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, 74–83.

[2*] Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, *119*, 87–108.

[3*] Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*.

[4*] Fuchs, C., & Hess, T. (2018). Becoming Agile in the Digital Transformation: The Process of a Large-Scale Agile Transformation. *Thirty Ninth International Conference on Information Systems*, *39*, 18. San Francisco, California: Association for Information Systems.

[5*] Uludag, Ö., & Matthes, F. (2019). Identifying and Documenting Recurring Concerns and Best Practices of Agile Coaches and Scrum Masters in Large-Scale Agile Development. *HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. 26*, 22.