

## Outline



#### Motivation

Research Questions & Research Approach

Findings

**Case Description** 

Adopted Agile Program

Identified Recurring Concerns

**Identified Best Practices** 

Lessons Learned of Implementing Pattern Catalogue Best Practice

Limitations

Conclusion & Future Work

## **Motivation**





#### Agile

... deals with changing requirements and promises higher customer satisfaction through continuous delivery and high customer involvement [3,6]



## Large-Scale

... companies are inspired by the success of agile practices and apply agile practices to large-scale projects[9].



## Challenges

... arise through the large-scale adaption. Geographically-distributed teams, the scaling of agile practices and adoption to the project team are just a few[2,3,7,9].



#### Research

... already identified 14
Stakeholder groups and
79 challenges[8]
... offers Patterns to be
applied for those
stakeholder groups [5]



#### **Patterns**

... as a structured solution to recurring problems [1,7]



#### **Evaluation**

... of observed Patterns in research according to concerns and stakeholder groups. [7]

## Outline



Motivation

## Research Questions & Research Approach

## Findings

**Case Description** 

Adopted Agile Program

Identified Recurring Concerns

**Identified Best Practices** 

Lessons Learned of Implementing Pattern Catalogue Best Practice

Limitations

Conclusion & Future Work

## Research Questions



RQ1

How has LeSS been adopted and applied at Robert Bosch?

RQ2

What are recurring concerns of stakeholders at the product organization of Robert Bosch?

RQ3

What are good practices for addressing recurring concerns of stakeholders of the product organization of Robert Bosch?

RQ4

Which bad practices should be avoided in the product organization of Robert Bosch?

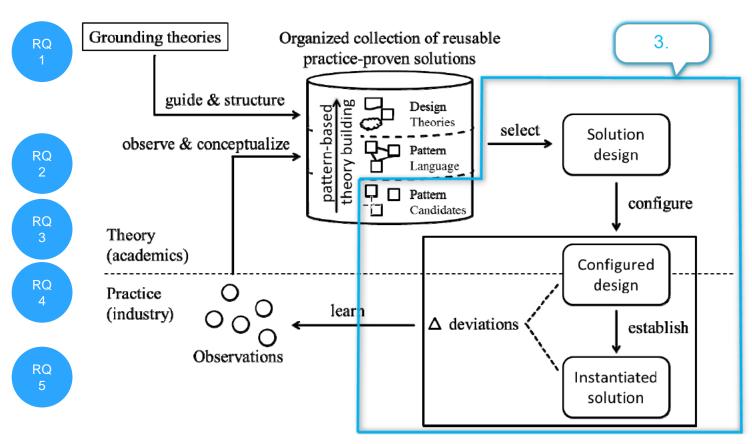
RQ5

What are the lessons learned of implementing already observed best practices in the product organization of Robert Bosch?

# Research Approach



- 1. Interviews for identifying adaptions of LeSS at agile program
- 2. Interviews identifying concerns and patterns with stakeholder groups (SM,Dev, PO) following Pattern-Based Research Design[1]
- 3. Select & instantiate solution according to Pattern-Based Research Design[1]



[1]

# Research Approach

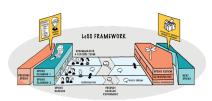


1. Interviews for identifying adaptions of LeSS at agile program



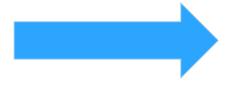
3 interviews with agileadoption questionnaire

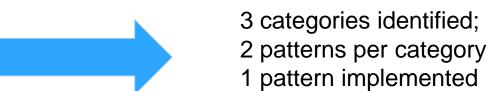
11 interviews with concerns



[6]

- 2. Interviews identifying concerns and patterns with stakeholder groups (SM,Dev, PO) following Pattern-Based Research Design[1]
- 3. Select & instantiate solution according to Pattern-Based Research Design[1]







[5]

## Outline



Motivation

Research Questions & Research Approach

## Findings

**Case Description** 

Adopted Agile Program

Identified Recurring Concerns

**Identified Best Practices** 

Lessons Learned of Implementing Pattern Catalogue Best Practice

Limitations

Conclusion & Future Work

# Adopted Agile Framework

## **Case Description**







**GS/POI21**Purchasing Integration, Lean and Unified Management











- 5 Purpose Teams
- total of 52 project members

## Interviews general Information

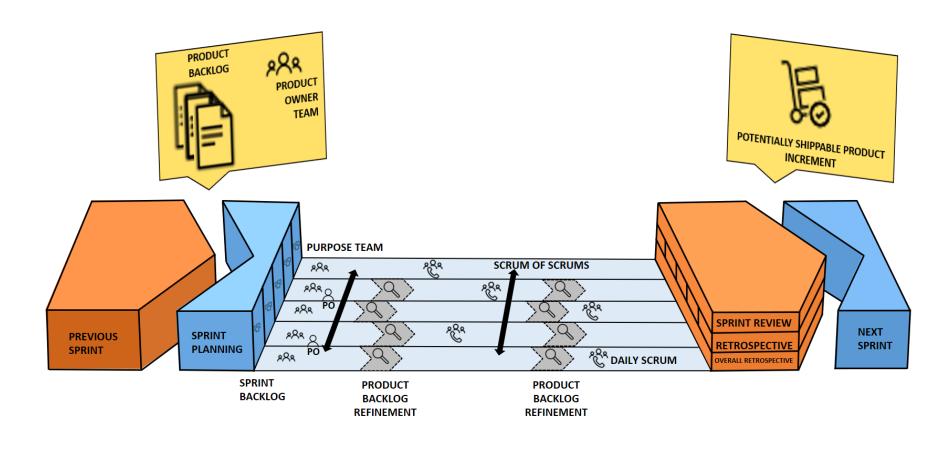


- 3 structured interviews using agile-adoption questionnaire
- 11 semi-structured interviews using concerns and patterns questionnaire
- Average time 1:10:09 h

| attorr |                  |           |                     |  |
|--------|------------------|-----------|---------------------|--|
| No.    | Role             | Duration  | Questionnaire       |  |
| 1      | Product Owner    | 1:06:21 h | Concerns & Patterns |  |
| 2      | Scrum Master     | 1:07:24 h | Concerns & Patterns |  |
| 3      | Product Owner    | 1:05:41 h | Concerns & Patterns |  |
| 4      | Scrum Master     | 1:23:33 h | Agile-Adoption      |  |
| 5      | Product Owner    | 1:23:54 h | Concerns & Patterns |  |
| 6      | Product Owner    | 1:17:55 h | Agile-Adoption      |  |
| 7      | Development Team | 1:12:48 h | Concerns & Patterns |  |
| 8      | Product Owner    | 1:13:40 h | Concerns & Patterns |  |
| 9      | Development Team | 1:02:52 h | Concerns & Patterns |  |
| 10     | Development Team | 0:50:42 h | Concerns & Patterns |  |
| 11     | Product Owner    | 1:13:52 h | Concerns & Patterns |  |
| 12     | Development Team | 1:18:29 h | Concerns & Patterns |  |
| 13     | Development Team | 0:54:52 h | Concerns & Patterns |  |
| 14     | Scrum Master     | 1:12:02 h | Agile-Adoption      |  |

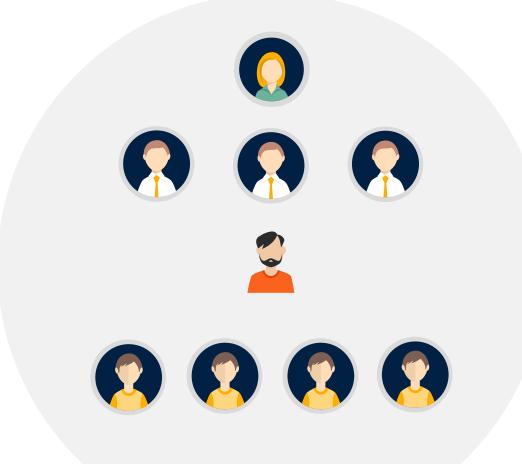
# Adopted Agile Program Case Study Partner





Purpose Team = Feature Team with Process Consultant and Product Owner Team





**Scrum Master** 

**Product Owner Team** 

**Process Consultant (50/50)** 

**Development & Test Team** 

# Newly identified Recurring Concerns



| Recurring Concern   | Stakeholder      | Category                     |
|---|------------------|------------------------------|
| C-79 How to balance amount and quality of delivered requirements                        | Product Owner    | Quality Assurance            |
| C-80 How to manage overarching backlog item prioritization with multiple product owners | Product Owner    | Communication & Coordination |
| C-81 How to understand all interfaces and dependencies of the system                    | Program Specific | Knowledge<br>Management      |
| C-82 How to support an onboarding approach for different stakeholder groups             | Program Specific | Knowledge<br>Management      |
| C-83 How to manage requirement development for multiple teams                           | Product Owner    | Project Management           |
| C-84 How to involve all team members in solution generation                             | Scrum Master     | Culture & Mindset            |
| C-85 How to share domain knowledge across agile teams                                   | Development Team | Knowledge<br>Management      |
| C-86 How to involve remotely working/external colleagues                                | Development Team | Tooling                      |
| C-87 How to clarify details outside of meetings in cross-shore agile teams              | Development Team | Communication & Coordination |

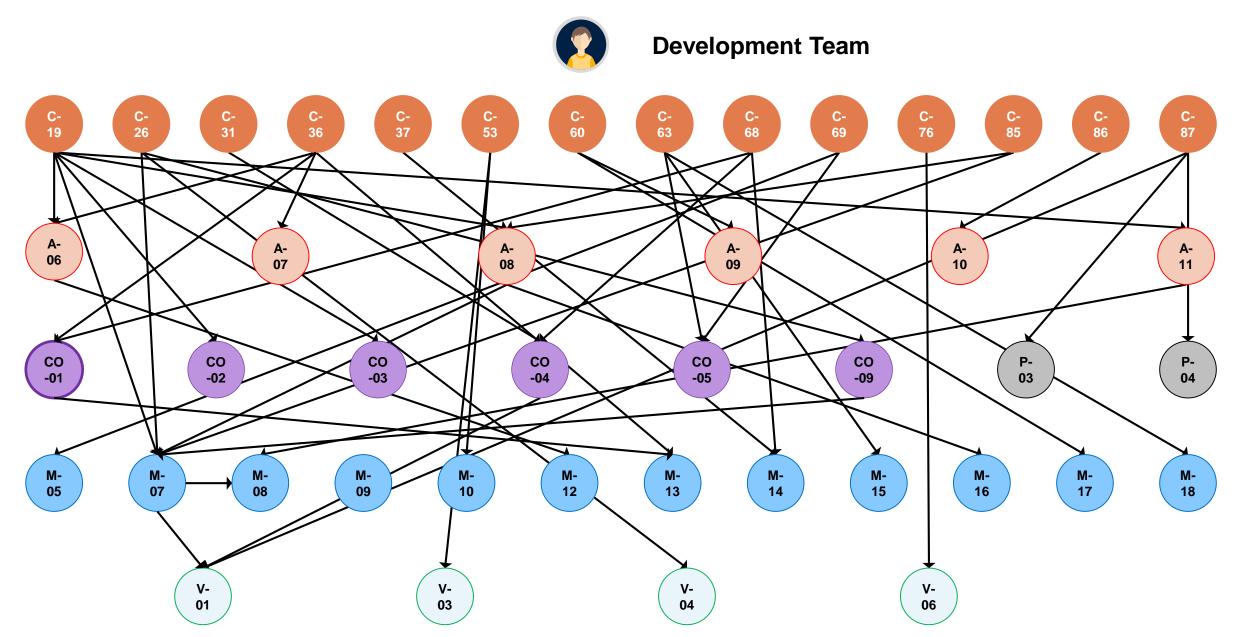
#### **Identified Best Practices**



P-03 M-01 M-07 M-11 A-02 P-01 A-07 M-03 M-15 **Avoid Extra Product Owner Don't Force Team** Reserved **Process Don't Have New** Geographically **Bug Prioritization Impact Analysis** Meetings Capacity Consultant Team Coherence Distributed Year Resolution **Meeting Hours** Dilemma M-12 M-16 A-03 P-04 M-02 M-04 M-08 Shifting Incremental Don't Assume **Purpose Teams** Semi Co-Location **Scope Change Acceptance** A-08 P-02 Responsibility **Onboarding Autonomous** Criteria **Don't Limit Velocity Sheet** Onboarding Knowledge Transfer to Knowledge M-17 M-05 CO-04 CO-08 M-09 M-13 Transfer-**Planning Poker Definition of** CO-1 **Kickoff** Newsflash **Story Points Automation Lead** A-04 Workshops light Ready and Community of Don't Assume **Definition of Done Practice** Mutual Terminology CO-05 CO-09 M-14 A-09 M-10 M-18 **Product Backlog Process** Docupedia for **Don't Misuse** M-06 CO-01 Subtask-Testing **Proof of Concept** Refinement Consultant Architecture Estimation **Pre-Planning Functional** A-05 Meeting **Documentation** Creation Coordination **Splitting** Don't Forward + Requirements CO-06 A-1 V-03 **Direct Customer** Don't Use CO-10 V-01 JIRA-Board CO-02 A-10 V-05 Communication Frameworks as Go-Live Dependency Face-to-Face **Don't Limit** Power BI A-06 Recipes Celebration Matrix **External** Knowledge \*\*\*\* **Don't Overshoot** Transfer Colleagues Coordination V-04 Meetings CO-07 Interface \*\* A-01 Regelrunden V-06 CO-11 V-02 **Architecture** Don't Have **Epic-Plan Game Third Party** CO-03 A-11 **Burn-Down Chart Board** Blurred Interface -Exemplary Don't Capsulate **Boundaries** Knowledge Planning Meeting Teams too much Transfer ID Name Pattern Coordination Methodology Principle Anti-Pattern Viewpoint Occurence Pattern Pattern Pattern

Mapping Concerns to Patterns for Development Team





# **Focus Concern Categories**



# Knowledge Management

- (1) 8 Concerns (3 new)
- 1 5 Anti-Pattern Candidates



(1) 13 Concerns (1 new)

1 Anti-Patterns1 Anti Pattern Candidate



- (1) 5 Concerns (1 new)
- 1 3 Anti-Pattern Candidates

## Lessons Learned Community of Practice for PO





## **Community of Practice for POs**

Why?



Each agile team has several Product Owners



8 Concerns (3 newly identified)



5 Anti-Patterns

How?



Following CoP pattern identified by sebis-chair[7]



Addressing characteristics of a successful CoP by Paasivaara [4]



Comparing to interview information from CoPs at case study partner

## Lessons Learned Community of Practice for PO



#### 21.02.2020 Kickoff 2020

Mittwoch, 19. Februar 2020

10:39

#### Agenda 1. What makes a successful CoP a. 8 characteristics by Paasivara (see Sharepoint) (presentation me 5 mins) 2. Passionate leader Interesting topic with concrete 3. Proper agenda benefits to participants 8. Cross-site 4. Decision making b. participation when Successful CoP authority needed 7. Suitable rhythm 5. Open community 6. Supporting tools to create transparency

+ Add Page CoP Agile Master 26.01.2020 07.02.2020 27.02.2020 Backlog **Culture things** CoD Development Misc 2020 01 31 - KickOff 2020 2020 02 21 - Session II 2020 03 13 - Session III CoP UI Umbrella Retro - CoP UI Works CoP PO **Topic Collection** 21.02.2020 Kickoff 2020 12.03.2020 Session II

## Lessons Learned Community of Practice for PO contd.



- b. Who will be moderating the CoP?
  - a. Rotating moderation deciding at the end of a CoP meeting, who will moderate the next CoP (5 min timeslot for planning next CoP and Moderation)
  - b. Agenda is generated by moderator of next CoP
- c. How often will we meet?
  - i. Last Thursday of each month. 1h Blocker from 2-3 pm
- d. How will we decide on which Topics to discuss?
  - a. When topics are already present, decision at the end of CoP which topics will be discussed.
  - b. OneNote for all topics and documentation of meetings. Startpage for Topic collection and general information
- e. For now only Purchasing Galaxy PO as members, invitation via mail. Beas initiator for creating meeting in calendar
- f. Meeting participation
  - a. Skype Meeting and meeting rooms as available
- g. Decision making enabled by default
- 3. We will discuss our First Topic:
  - a. How to create precise requirement specifications for the development team?
    - a. CPREQ-25606
      - i. Split requirements if possible, try not to have more than 5 d development days in one ticket.
      - ii. Use subtasks and encourage developers to tell you when requirement is too large or complex.
      - iii. When having complex requirements have detailed discussion beforehand
      - iv. Topic pushed to next CoP
- 4. Topic Collection
  - a. Large or complex Requirement Splitting
  - b. Estimation of user stories and actual effort
  - c. Conflict in Roadmaps (how to deal with conflict in plans)
  - d. Motivating developers
- 5. Next Moderator
  - a. D
  - b. Date 26.03
- We'll have a short Feedback session (10 mins)

+ Add Page

CoP Agile Master

26.01.2020

07.02.2020

27.02.2020

Backlog

Culture things

CoD Development

Misc

2020 01 31 - KickOff 2020

2020 02 21 - Session II

2020 03 13 - Session III

CoP UI

Umbrella Retro - CoP UI Works

CoP PO

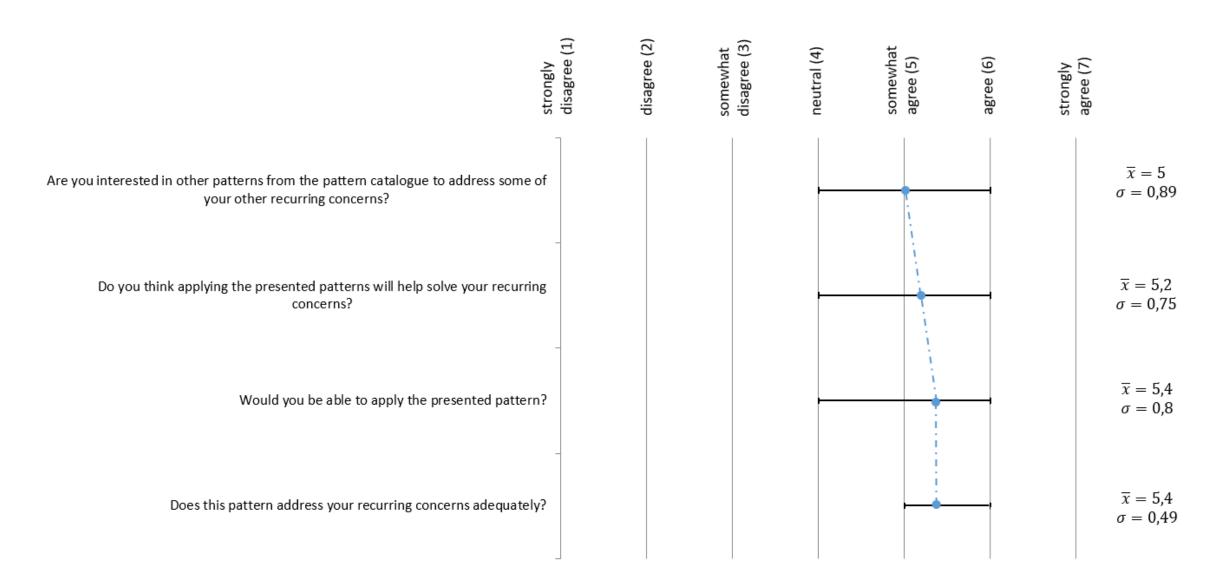
**Topic Collection** 

21.02.2020 Kickoff 2020

12.03.2020 Session II

## Lessons Learned contd.





## Outline



Motivation

Research Questions & Research Approach

Findings

**Case Description** 

Adopted Agile Program

Identified Recurring Concerns

**Identified Best Practices** 

Lessons Learned of Implementing Pattern Catalogue Best Practice

#### Limitations

Conclusion & Future Work

## Limitations





Only two patterns identified



Several new best practices identified



Longer time horizon needed for impactful observation and adaption of patterns of the pattern catalogue



Only action plan observed

→ No certainty whether the implemented pattern and the planned to be implementend patterns will help solve the concerns or are continually applied

## Outline



Motivation

Research Questions & Research Approach

Findings

**Case Description** 

Adopted Agile Program

Identified Recurring Concerns

**Identified Best Practices** 

Lessons Learned of Implementing Pattern Catalogue Best Practice

Limitations

Conclusion & Future Work

## Conclusion & Future Work



#### **Key Artifacts**

LSAD framework for case study partner, 54 Concerns identified, +9 new Concerns identified, 1 new role (PC), 1 Anti-Pattern, 1 CO-Pattern, 50 Pattern-Candidates identified(4 Principle, 11 Anti-Patterns, 11 CO-, 18 M- and 6 V-Pattern-Candidates), 1 Pattern successfully validated

#### **Key Findings**

- Introduction of new role Process Consultant creates an openly admitted internal silo, however, the role helps addressing concerns in Knowledge Management.
  - Domain knowledge with single point of contact
  - Mini-Architects as part of team
- Kickoffs and similar events, bringing together geographically distributed teams, are vital
- Patterns of the pattern catalogue address concerns associated with them
- Willingness for adopting patterns from other organizations is high
- Out of the six presented patterns, one was implemented and three more are planned to be implemented in the future

#### **Future Work**

- Actively implement several patterns at case study partner and observe for a longer time horizon
- Observing the lessons learned for multiple patterns
- Application of more patterns addressing concerns of other stakeholder groups (Agile Coach, ...)
- Further validation of patterns identified in practice at case study partner

## References



- [1] Buckl, S., Matthes, F., Schneider, A. W., & Schweda, C. M. (2013). Pattern-based design research—an iterative research method balancing rigor and relevance. In International Conference on Design Science Research in Information Systems (pp. 73-87). Springer, Berlin, Heidelberg.
- [2] Dingsøyr, T., & Moe, N. B. (2014). Towards principles of large-scale agile development. In International Conference on Agile Software Development (pp. 1-8). Springer, Cham.
- [3] Larman, C., & Vodde, B. (2009). Scaling lean & agile development. Organization, 230(11).
- [4] Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization—Case Ericsson. Information and Software Technology, 56(12), 1556-1577.
- [5] Scaling-Agile Hub. Patterns. <a href="https://scaling-agile-hub.sebis.in.tum.de/#/patterns">https://scaling-agile-hub.sebis.in.tum.de/#/patterns</a>.
- [6] The LeSS Company B.V. (2014). Overview Large Scale Scrum(LeSS). <a href="https://less.works/">https://less.works/</a>.
- [7] Uludag, Ö., Harders, N. M., & Matthes, F. (2019, July). Documenting recurring concerns and patterns in large-scale agile development. In Proceedings of the 24th European Conference on Pattern Languages of Programs (pp. 1-17).
- [8] Uludag, Ö., Kleehaus, M., Caprano, C., & Matthes, F. (2018). Identifying and structuring challenges in large-scale agile development based on a structured literature review. In 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC) (pp. 191-197). IEEE.
- [9] VersionOne, C. 13th Annual State of Agile Report. (2019).



B.Sc.

# **BACKUP SLIDES**

# Responsibilities and Challenges Scrum Master, Development Team and Product Owner (1/2)





#### **Scrum Master**

#### Responsibilities

- Part of agile team
- Enable development process
- Remove impediments
- Promote agile program
- Manage meetings

#### Challenges

- 1. How to deal with incorrect agile practices
- 2. How to provide sufficient tools and infrastructure for remote communications
- 3. How to deal with increasing workload of key stakeholders
- 4. How to synchronize working hours of cross-shore meetings
- 5. How to create a culture of continuous improvement
- 6. How to rearrange physical spaces
- 7. How to deal with closed mindedness
- 8. How to deal with increased efforts by establishing inter-team communication
- 9. How to deal with lacking sense of ownership responsibilities for developed services
- 10. How to define clear roles and responsibilities
- 11. How to establish a common understanding of agile software development
- 12. How to deal with cultural differences between cross-shore agile teams
- 13. How to encourage development teams to talk about tasks and impediments
- 14. How to empower agile teams to make decisions
- 15. How to form and manage autonomous teams
- 16. How to involve all team members in solution generation\*\*



## **Development Team**

- · Part of agile team
- Clarification non-functional requirements\*
- Estimating user stories
- · Architecture and design
- Implementation
- Sprint Review
- 1. How to deal with increasing workload of key stakeholders
- How to deal with internal silos.
- B. How to align and communicate architectural decisions
- 4. How to deal with geographical distance between agile teams
- How to establish automated testing
- 6. How to create lightweight documentation
- 7. How to apply agile practices for developing or maintaining legacy systems
- 8. How to ensure traceability of tests and requirements
- 9. How to create and estimate user stories
- 10. How to explain requirements to stakeholders
- 11. How to write understandable automated tests
- 12. How to establish test verification
- 13. How to coordinate test and deployment with external parties
- 14. How to share domain knowledge across agile teams\*\*
- 15. How to involve remotely working/external colleagues\*\*
- 16. How to clarify details outside of meetings in cross-shore agile teams\*\*

<sup>\*</sup> Process Consultant as part of Development Team

<sup>\*\*</sup>Counting from 78 identified by [4]

# Responsibilities and Challenges Scrum Master, Development Team and Product Owner (2/2)





#### **Product Owner**

#### Responsibilities

- Part of agile team
- Clarification of functional requirements
- User Story description
- Customer communication
- Manage epic plan

#### Challenges

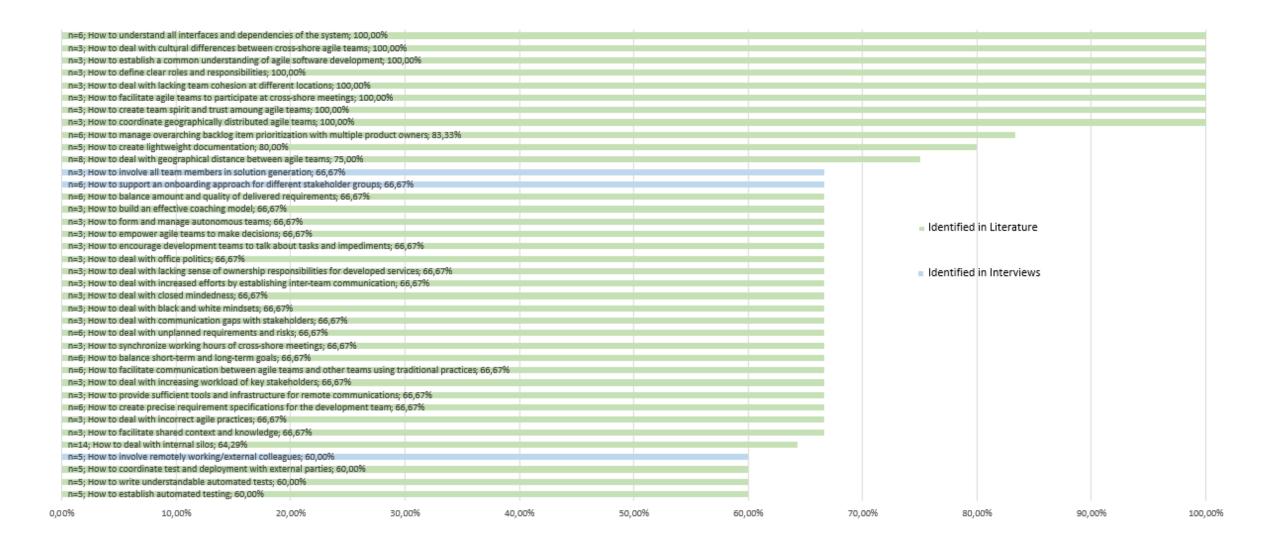
- 1. How to create precise requirement specifications for the development team
- 2. How to elicit and refine requirements of end users
- 3. How to split large and complex requirements into smaller requirements
- 4. How to deal with internal silos
- 5. How to facilitate communication between agile teams and other teams using traditional practices
- 6. How to balance short-term and long-term goals
- 7. How to communicate business requirements to development teams
- 8. How to define clear and visible priorities
- 9. How to deal with unplanned requirements and risks
- 10. How to enforce customer involvement
- 11. How to ensure traceability of tests and requirements
- 12. How to make a cost and schedule estimation
- 13. How to create and estimate user stories
- 14. How to deal with fixed price contracts in agile software development
- 15. How to establish requirements verification
- 16. How to define high-level requirements a.k.a. epics

- 17. How to measure the success of the large-scale agile development program
- 18. How to balance amount and quality of delivered requirements\*\*
- 19. How to manage overarching backlog item prioritization with multiple product owners\*\*
- 20. How to understand all interfaces and dependencies of the system\*\*
- 21. How to support an onboarding approach for different stakeholder groups\*\*
- 22. How to manage requirement development for multiple teams\*\*

<sup>\*\*</sup>Counting from 78 identified by [4]

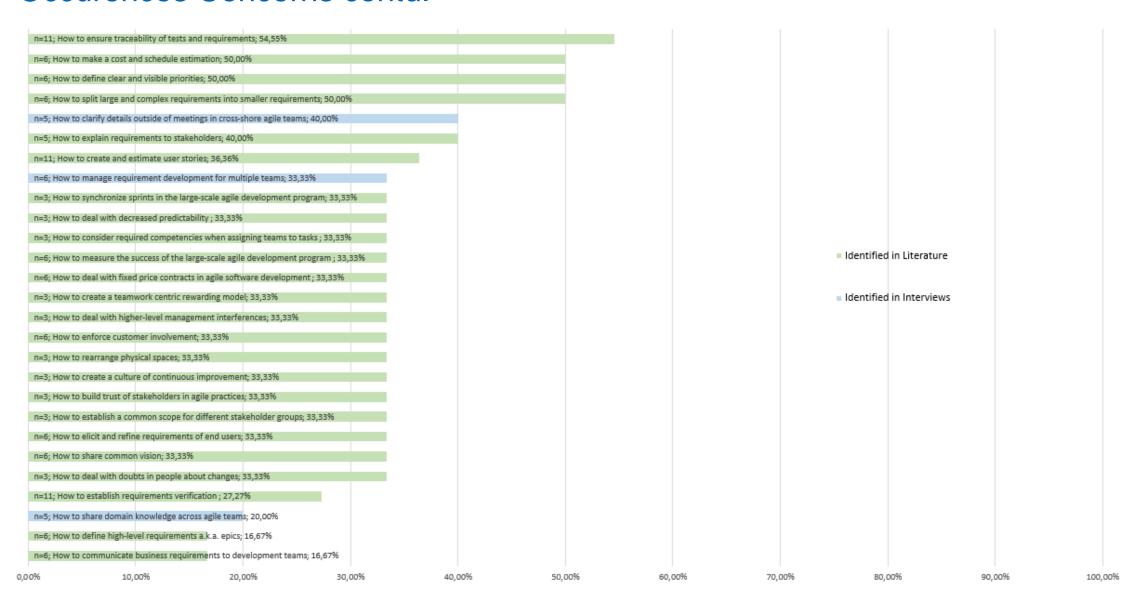
## Occurences Concerns





## Occurences Concerns contd.





# **Exemplary Pattern Candidate**



| Process Consultant |  |  |
|--------------------|--|--|
| ID                 | M-07   |  |
| Name               | Process Consultant   |  |
| Alias              | Lead Developer   |  |
| Summary            | Process Consultants combine responsibilities of the roles of a developer, architect and consultant. The role finds application in LSAD with domain-specific knowledge. It can be compared to the concept of an Lead Developer. The Process Consultant aids in architecture decision making, knowledge sharing with less experienced developers and Product Owners and more. The Process Consultant is the most experienced developer of a domain responsible for the support in development of requirements as well as development itself. |  |

# Exemplary Pattern Candidate contd.



| Process Consultant |   |  |  |  |
|--------------------|---|--|--|--|
| Example            | Technical LLC identified during the adoption process of a LSAD framework, the need for a Process Consultant. Process Consultants (Lead Developer) are domain experts supporting the development process of an system consisting of migrated legacy systems with inexperienced developers working on it.   |  |  |  |
| Context            | Complex and historically grown system. Migrating several legacy systems and providing complex functions. Project growth leads to inexperienced developers working on the system.  |  |  |  |
| Problem            | C-19 How to deal with internal silos<br>C-85 How to share domain knowledge across teams   |  |  |  |
| Forces             | High pace feature-driven development with newly created teams. Several bottlenecks in respect to knowledge as the teams work on legacy systems with complex functions and dependencies.   |  |  |  |
| Solution           | Process Consultant is like a technical lead for a specific domain and product.  Creating an access point to knowledge by assigning a experienced developer, who can help less experienced developers and Product Owners understand system specifications and support creation process of new features and feature change.  The Process Consultant is a 50/50 role, half of his time is allocated in supporting other developers and the PO work with the process.  The other half is allocated for implementations and supporting architectural conception, alignment and implementation. |  |  |  |

# Exemplary Pattern Candidate contd.

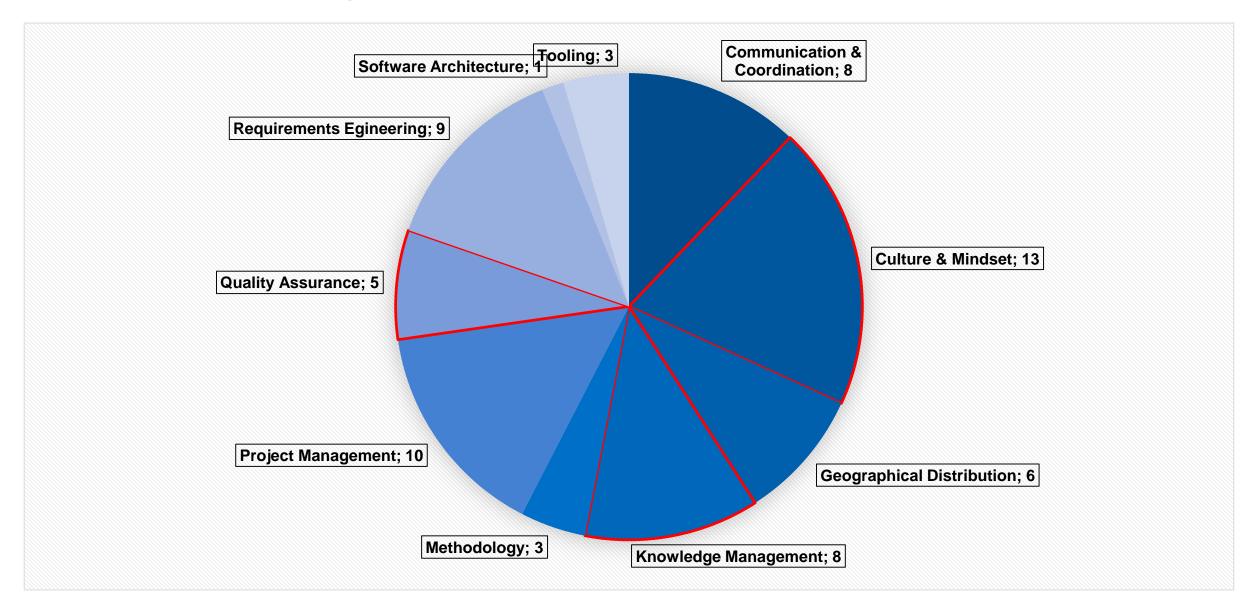


|          | Process Consultant   |  |  |
|----------|--|--|--|
| Variants | The Program uses experienced developers for multiple topics( architecture and team support, coaching for specific architecture topics, like UI/UX and Testing). Knowledge Transfer Workshops offer a variation.  |  |  |
| Conseque | <ul> <li>Advantages</li> <li>Offers solution for knowledge holes by forcefully creating internal silos</li> <li>Clear Contact Point for developers and PO alike through technical responsible person for domain</li> <li>Accountability for product and domain from technical aspect</li> <li>Efficiency Improvement with inexperienced developers</li> <li>No stoppage during development process</li> <li>introduction reduces bottlenecks by freeing up the day of these bottleneck to play a supportive knowledge transfer role</li> <li>Limits the functional knowledge and process functional knowledge of developers</li> <li>Intentional Internal silos</li> </ul> |  |  |
| See also | M-08 Purpose Teams, C-09 Process Consultant Meeting  |  |  |

200302 Holz MT Final

# **Overall Concern Categories**





## **Introduced Patterns**



- Knowledge Management
  - Supervision
  - CoP for POs
- Culture & Mindset
  - Event-Storming Workshops (DDD)
  - Celebrate and Publish Success
- Quality Assurance
  - Quality Gates
  - Communicate Architecture

# Instantiation Process Proposed



