The Role of Architecture in a Scaled Agile Organization – A Case Study in the Insurance Industry

Christina Schimpfle, Master's Thesis – Final Presentation, 31.07.2017

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

Overview

1. Introduction
   - Digitization and agile Software Development
   - Agile Scaling Practices – Nexus, LeSS and SAFe
   - Domain-driven Design (DDD)

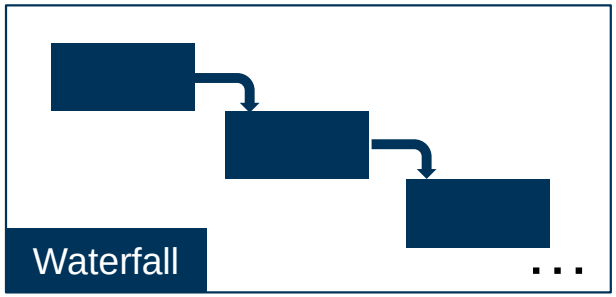
2. Framework for (Enterprise) Architecture in agile Teams
   - Overview
   - Development Process
   - Strategic DDD
   - Tactical DDD

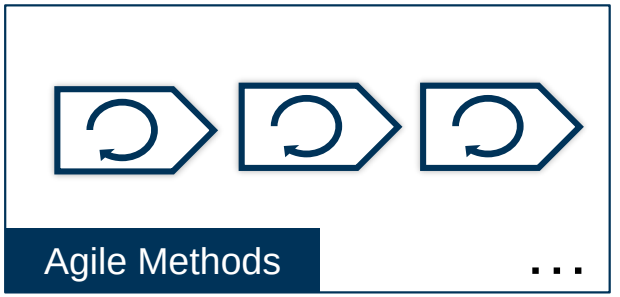
3. Evaluation
   - Pre-Study
   - Interviews

4. Discussion and Conclusion
   - Key Findings and Limitations
   - Outlook and future Work

Introduction
Digitization and agile Software Development

„Old World“ **&** **„New World“**





Scrum

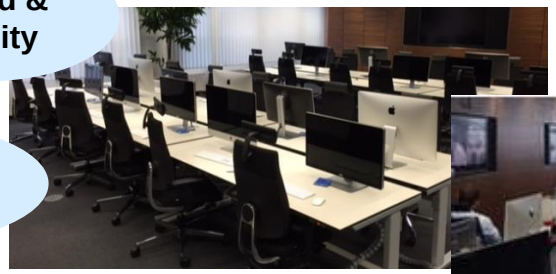
speed & quality

minimum viable product

customer feedback

co-located

transparency & business value



small teams

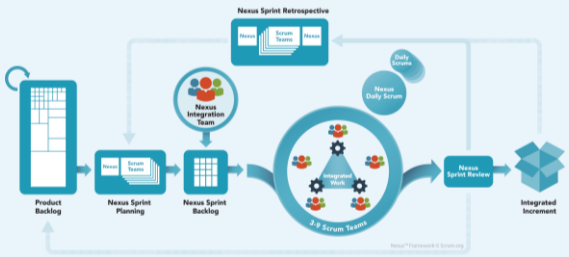
cross-functional

emergent architecture

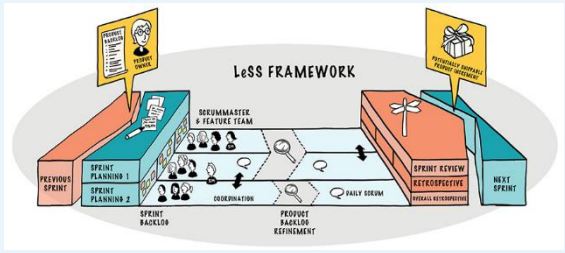
Introduction

Agile Scaling Practices – Nexus, LeSS and SAFe

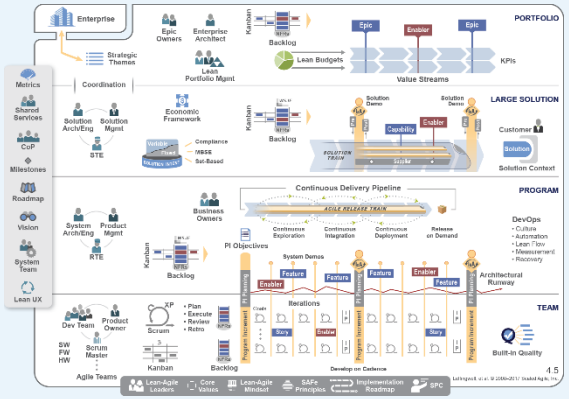
Nexus



Large-Scale Scrum (LeSS)



Scaled Agile Framework (SAFe)



Scrum in the teams

cross-team communication

overall meetings

inter-team coordination

Nexus: Architecture not addressed

LeSS: Emergent Architecture

LeSS & SAFe: Domain-driven Design

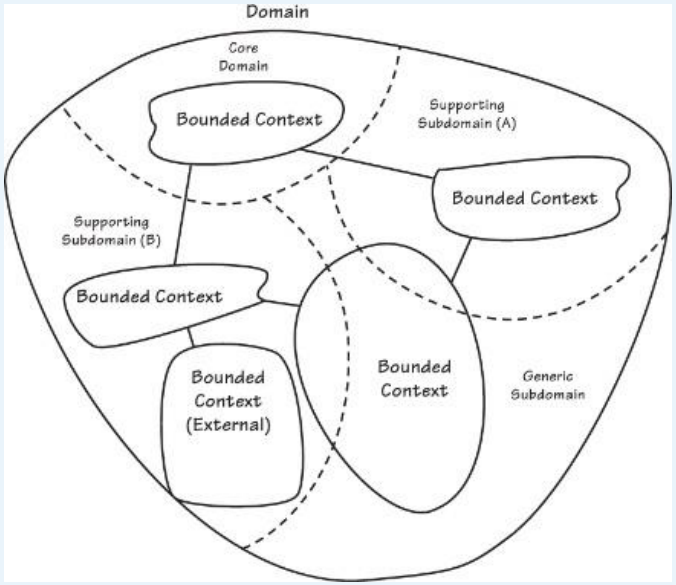
SAFe: Emergent Design vs. Intentional Architecture

Introduction
Domain-driven Design

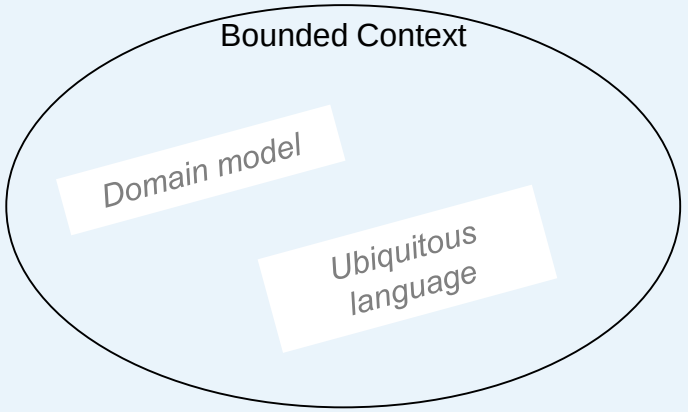
1. What is the role of architecture in scaled agile organizations?

Domain-driven Design (DDD)

Strategic DDD



Tactical DDD



One team works in one bounded context as independent as possible from other teams

2. How to adopt DDD in several agile teams in a large organization?

3. Which roles, processes, artifacts and tools are required for scaled DDD?

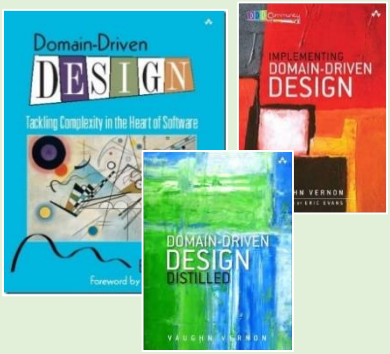
Framework for (Enterprise) Architecture in Agile Teams

Overview

Strategic DDD

Strategic Domain-driven Design

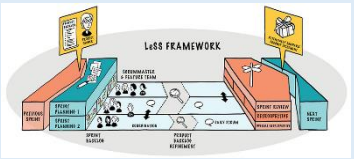
- Team structure and team responsibilities
- Matching business (sub-)domains and teams
- Overarching architectural questions addressed by Enterprise Architects
- Decision making by project managers



Development Process

Development Process for more than one agile team

- Synchronization of teams, cross-team coordination and communication
- Giving input for strategic DDD process
- Evolving and using domain models in the development process
- Enterprise Architects are facilitators for including new methods in the process

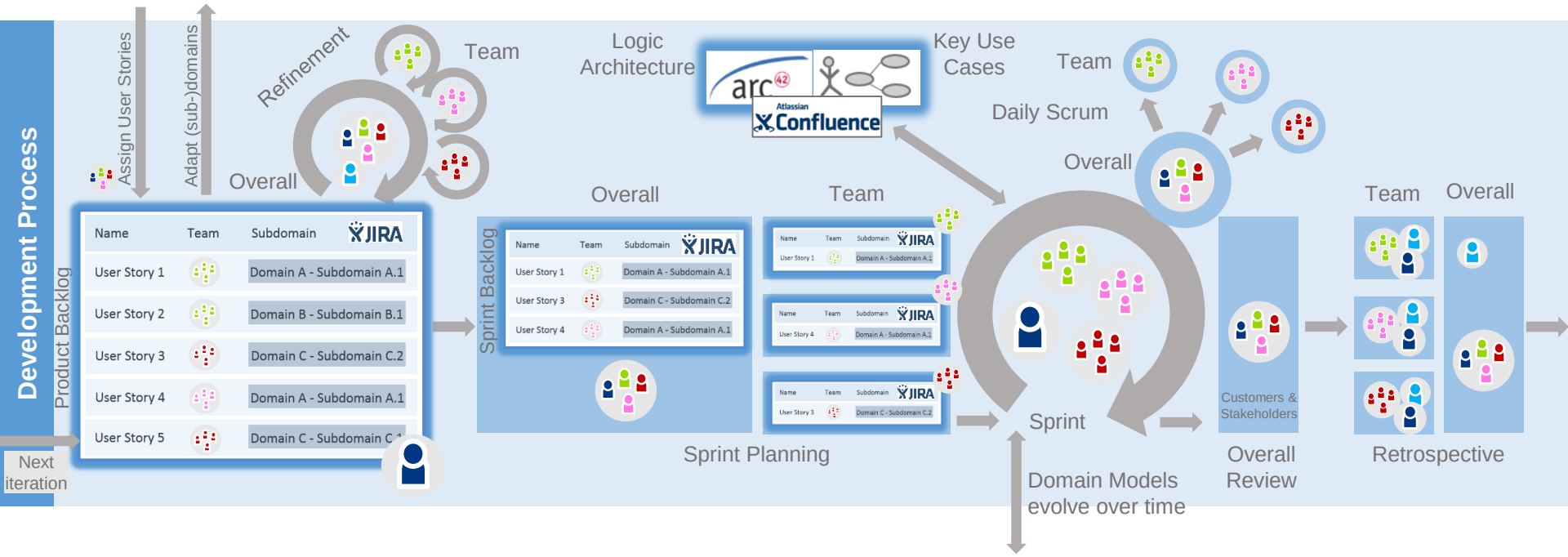


Tactical DDD

Tactical Domain-driven Design

- Formulation of domain models to gain a shared understanding for each agile team and the respective business experts
- Architectural and technical questions are addressed by the teams
- Enterprise Architects provide methodological guidance

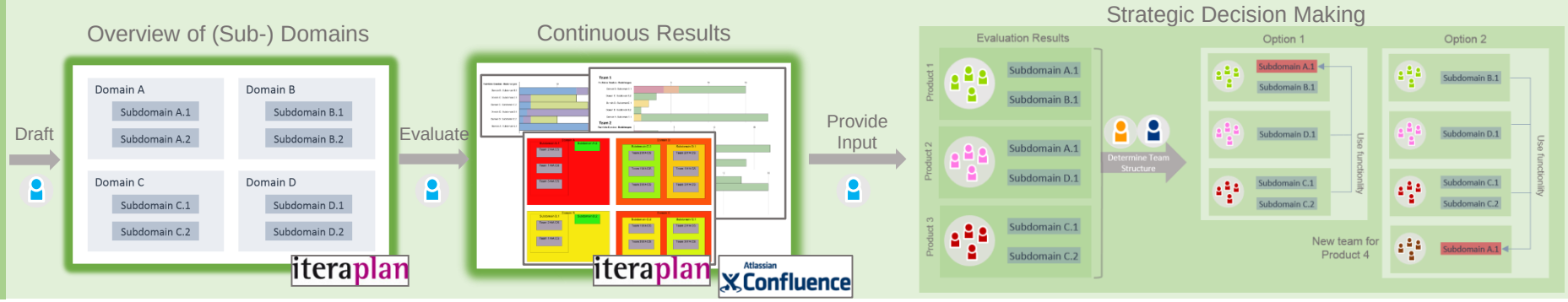
- Largely based on Large-Scale Scrum
- Scrum events plus some additional overarching events with representatives from each team
- Single product owner is responsible for single product backlog

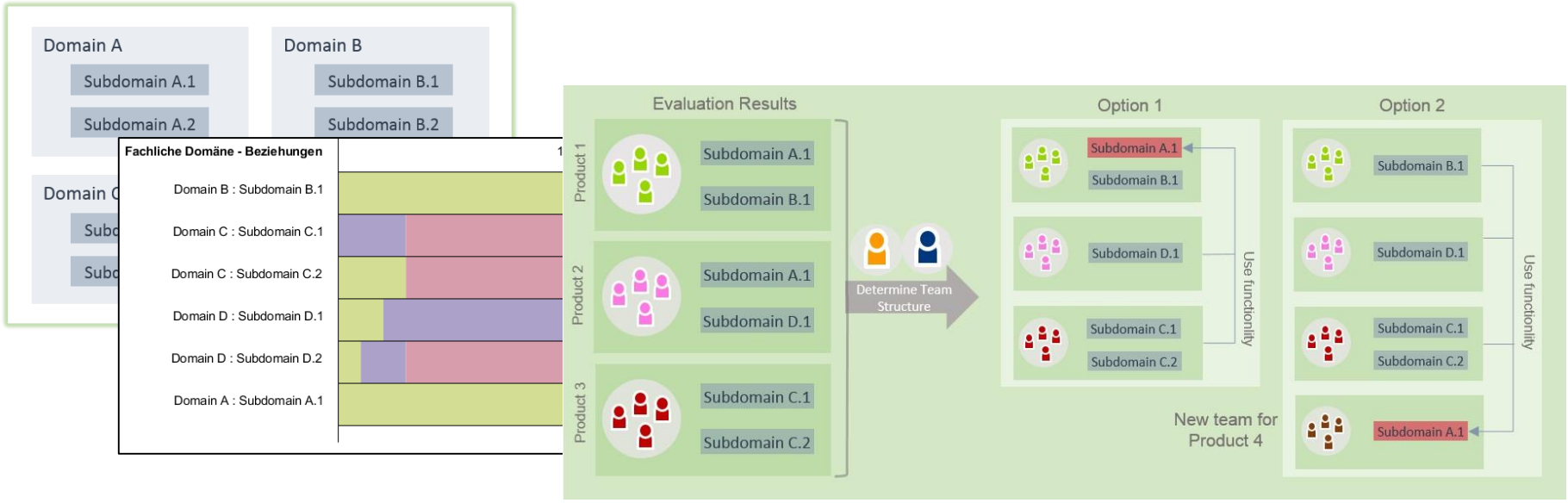


- Teams give input to the strategic DDD process
- Teams evolve and refine their domain models continuously
- Teams use their domain models for writing and refining user stories

➔ **Solving issues many teams face, while integrating the DDD components**

Strategic DDD





- Architects provide an architectural overview of (sub-) domains which evolves over time
- Teams assign their user stories to the architectural (sub-)domains
- Enterprise Architects evaluate the user story assignment continuously in an EAM Tool
- Decision makers use the results provided in the Wiki to decide about team structure

➔ **Goal is to structure teams dependent on the subdomains to gain more independency**
➔ **Responsibility for overarching functions should be identified**

- First step towards a domain model for each team with focus on the most essential events
- Domain models are evolved over time to include entities, value objects and further building blocks
  Teams are involved and profit, while architects provide method

➔ **Shared understanding and "ubiquitous" language in each team**

➔ **Identify elements to be discussed more in detail**

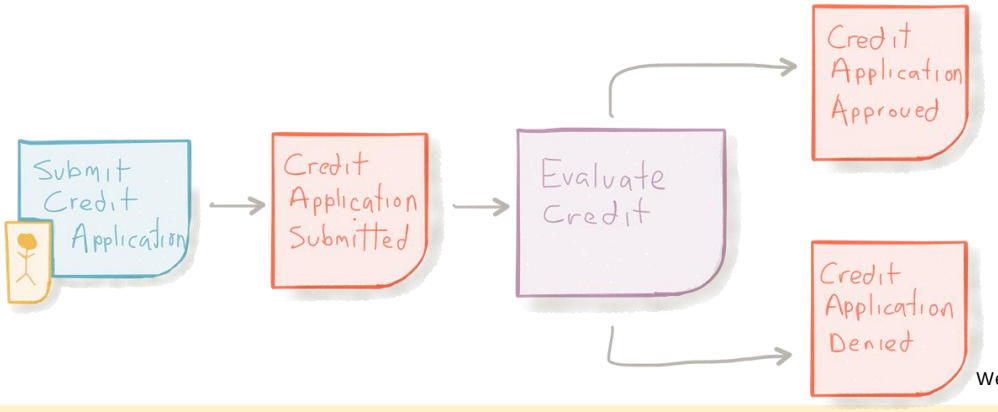
➔ **Identify (sub-) domain boundaries**

*"A **domain event** is anything that happens that is of interest to a domain expert."*

Steven A. Lowe, 2017

*"**Event Storming** is a meeting where all the key stakeholders get together in a room with a lot of modelling space and use stickies to describe what happens in a system."*

Jan Stenberg, 2016



Webber, 2017

Tactical DDD

Domain Models

Event Storming

Event Storming

Event Storming

Atlassian Confluence Team Room

Atlassian Confluence Team Room

Atlassian Confluence Team Room

Event Storming

Use and provide information

Domain Models evolve over time

Roles

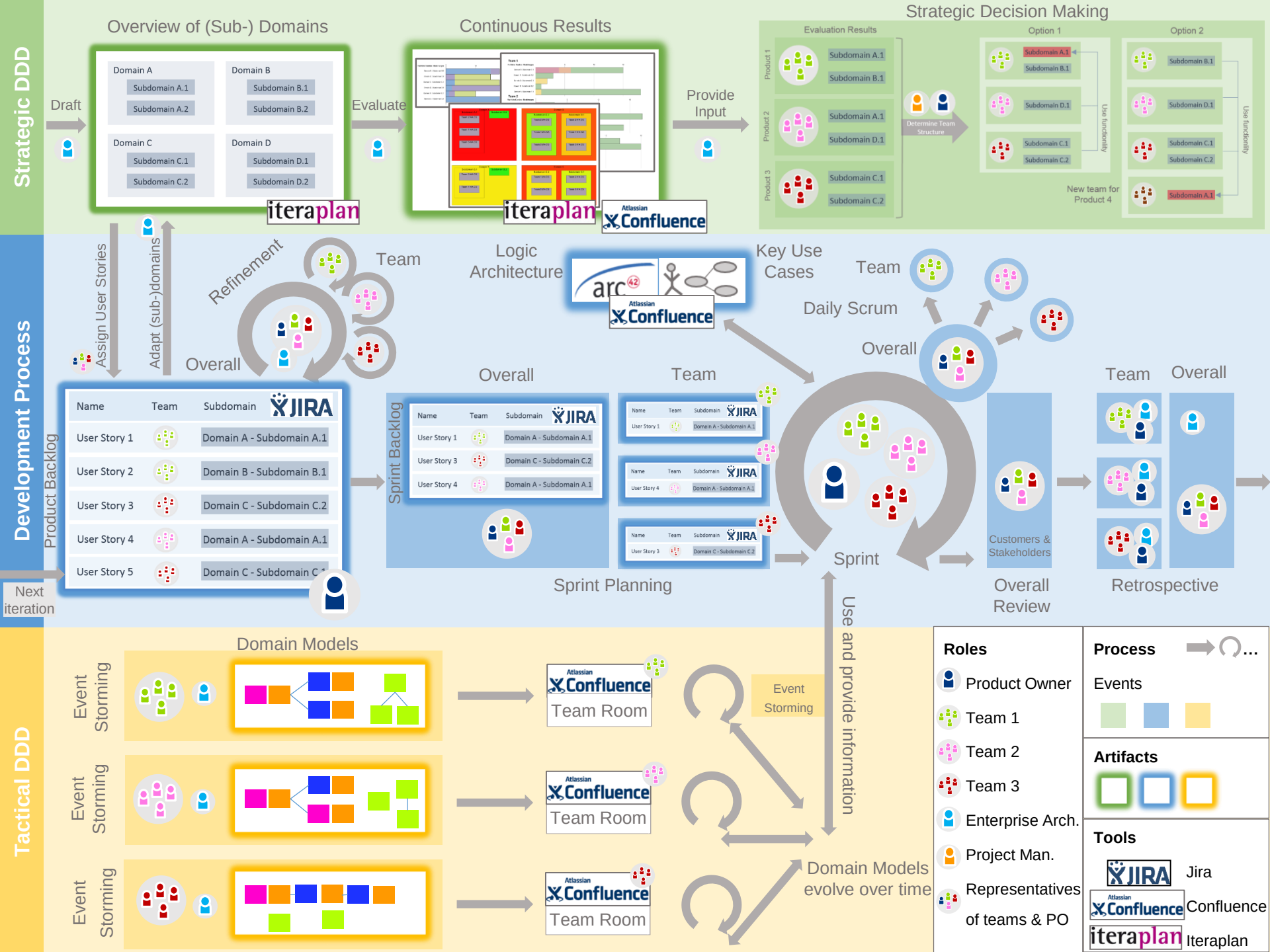
- Product Owner
- Team 1
- Team 2
- Team 3
- Enterprise Arch.
- Project Man.
- Representatives of teams & PO

Process ... Events

Artifacts

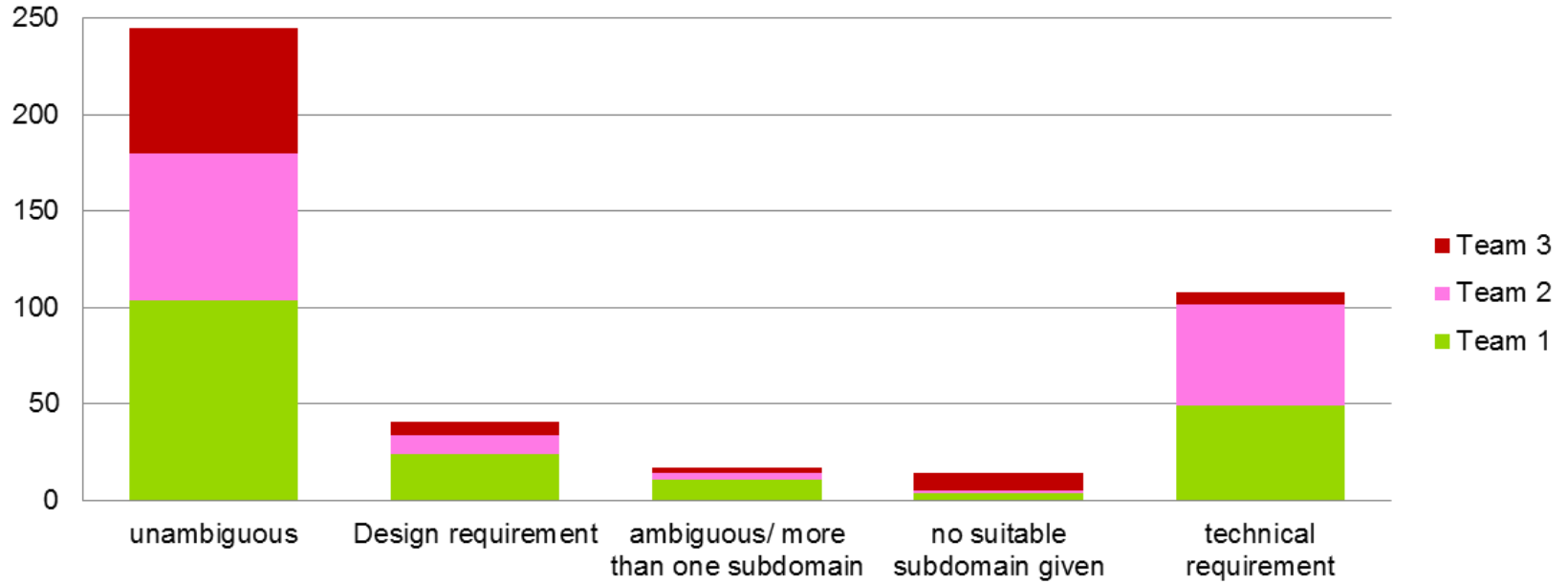
Tools

- Jira
- Confluence
- Iteraplan


Strategic DDD
Development Process
Tactical DDD
Overview of (Sub-) Domains
Domain A
Subdomain A.1
Subdomain A.2
Domain B
Subdomain B.1
Subdomain B.2
Domain C
Subdomain C.1
Subdomain C.2
Domain D
Subdomain D.1
Subdomain D.2
Draft
Evaluate
Continuous Results
Provide Input
Strategic Decision Making
Evaluation Results
Option 1
Option 2
Product 1
Product 2
Product 3
Determine Team Structure
Use functionality
New team for Product 4
Assign User Stories
Adapt (sub-)domains
Refinement
Team
Overall
Logic Architecture
Key Use Cases
Daily Scrum
Product Backlog
Name
Team
Subdomain
User Story 1
Domain A - Subdomain A.1
User Story 2
Domain B - Subdomain B.1
User Story 3
Domain C - Subdomain C.2
User Story 4
Domain A - Subdomain A.1
User Story 5
Sprint Backlog
Sprint Planning
Sprint
Customers & Stakeholders
Overall Review
Retrospective
Next iteration
Domain Models
Event Storming
Team Room
Use and provide information
Domain Models evolve over time
Roles
Product Owner
Team 1
Team 2
Team 3
Enterprise Arch.
Project Man.
Representatives of teams & PO
Process
Events
Artifacts
Tools
Jira
Confluence
Iteraplan

Evaluation

Pre-Study



- Design and technical user stories are not included in evaluations
- Adaption to the overview of (sub-)domains necessary
- Most stories could be assigned unambiguously

Assignment	User Stories	in %
In a meeting	78	18,35%
By a team member	113	26,59%
By an Enterprise Architect	234	55,06%
Sum	425	100,00%

➔ **Core subdomains are identified**

➔ **Overarching functions are identified**

➔ **Decision for a new team for one overarching subdomain based on the results**

Evaluation
Interviews

Nine Interviews with 20 questions each with following interview partners:

- Agile Master
- Two Business Analysts
- Product Owner

- Two Project Managers
- Department Head Sales Processes and Applications (Operational Organization)
- Enterprise Architect
- Domain Architect (Sales Processes and Applications)

➔ **General Questions:** Roles and major challenges

➔ **Strategic DDD Process:** User Story Assignment

➔ **Tactical DDD Process:** Event Storming and domain modeling

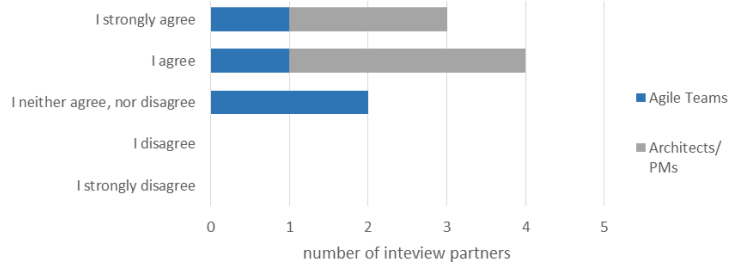
➔ **Development Process:** Involvement of architects, effort to integrate the DDD components, benefits and drawbacks of single backlog & Product Owner, synchronizing sprints

Evaluation
Interviews

Opinions: *positive* | *mixed* | *negative*

Strategic DDD

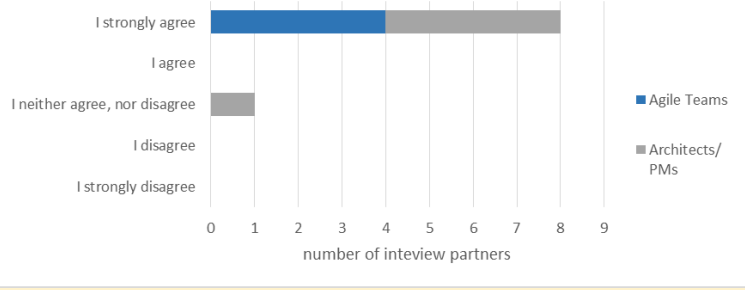
It is beneficial to further assign user stories and to evaluate the assignment continuously.



Development Process

Tactical DDD

It is beneficial if each teams creates and evolves its own domain model.



Evaluation
Interviews

Opinions

positive | mixed | negative

Strategic DDD

- "No real benefits for teams so far"
- "Maybe useful for teams in the future"
- "Helps to identify which subdomains require discussion/ action"
- "Helps to identify overarching functions/ domains"
- "Helps to determine team structure and responsibilities"
- "Beneficial for decisions makers"
- "Wiki is the best medium to present results"

Development Process

- "Synchronizing sprints: opinions differ"
- "Single backlog: opinions differ"
- "Different definitions of Product Owner"
- "Artifacts can be helpful in the meetings"

Tactical DDD

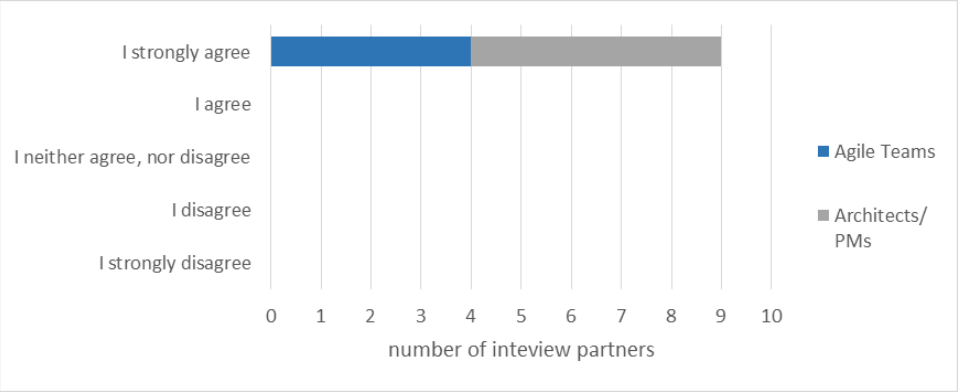
- "Event Storming is a simple method and creates a good atmosphere"
- "Useful for teams"
- "Focus on business logic"
- "Helpful for finding aggregates/ subdomains/ contexts"
- "Helpful for shared understanding"
- "Comparison of domains models can be helpful"
- "Helpful for scoping the next MVP"
- "Good method especially in the beginning of the development process"

Evaluation
Interviews

Opinions

positive *mixed* *negative*

It makes sense that architects support agile teams as described in the framework (from 1 – I strongly disagree to 5 – I strongly agree).



Evaluation
Interviews

Opinions

positive | mixed | negative

It makes sense that architects support agile teams as described in the framework .
➔ **All respondents agreed strongly (Average of 5)**

➔ Overall feedback about the framework was mostly positive

➔ Involvement of architects is considered as necessary

➔ Efforts to operationalize, scale and improve the framework further must be taken

➔ Benefits and results of the strategic DDD must be made available for the agile teams

Discussion and Conclusion

Key Findings and Limitations

Key Findings

- ➔ Architects coach new methodologies to the teams
- ➔ Architects as objective moderators in workshops
- ➔ Architects address overarching architectural and organizational questions

Limitations

- ➔ Only one case in one company
- ➔ No “cookbook” for the combination of scaling agile practices and DDD
- ➔ Architects’ role might change over time

Outlook and future Work

- ➔ Further operationalization to test framework in the long run at the large insurance company
- ➔ Analyze how the strategic and tactical DDD process can be connected in practice
- ➔ Additional cases in other companies to test the framework

Any Questions?

Sources

Brandolini, A. Introducing event storming. http://ziobrando.blogspot.de/2013/11/introducing-event-storming.html, 2013.

Stenberg, J. Experiences Using Event Storming. https://www.infoq.com/news/2016/06/event-storming-ddd, 2016.

Lowe, S. A. An introduction to event storming: The easy Way to achieve domain-driven design. https://techbeacon.com/introduction-event-storming-easy-way-achieve-domain-driven-design, 2017.

Scrum Alliance. The 2017 state of scrum report, 201

Webber, K. Modelling Reactive Systems with Event Storming and Domain-Driven Design. https://blog.redelastic.com/corporate-arts-crafts-modelling-reactive-systems-with-event-storming-73c6236f5dd7, 2017.

Evans, E. (2003). . *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley.

Evans, E. (2015).*Domain-Driven Design Reference – Definitions and Pattern Summaries*.

Larman;, C. & Vodde, B. (2016). *Large-Scale Scrum: More with LeSS*. Addison-Wesley Professional.

Millett, S. (2015). *Patterns, Principles and Practices of Domain-Driven Design.* John Wiley & Sons.

Scaled Agile Inc., (2016). *SAFe® 4.0 Introduction - Overview of the Scaled Agile Framework for Lean Software and Systems Engineerin*g (White Paper).

Vernon, V. (2013). *Implementing domain-driven design*. Addison-Wesley.

Vernon, V. (2016). *Domain-driven design distilled*. Addison-Wesley Professional.

Scaled Agile Inc. http://www.scaledagileframework.com/agile-architecture/, 2017.

Larman, C. & Vodde, B. https://less.works/less/technical-excellence/architecture-design.html, 2017.