

Future Software Organizations – Agile Goals and Roles

Maarit Laanti & Petri Kettunen



INTRODUCTION

Statements in this presentation

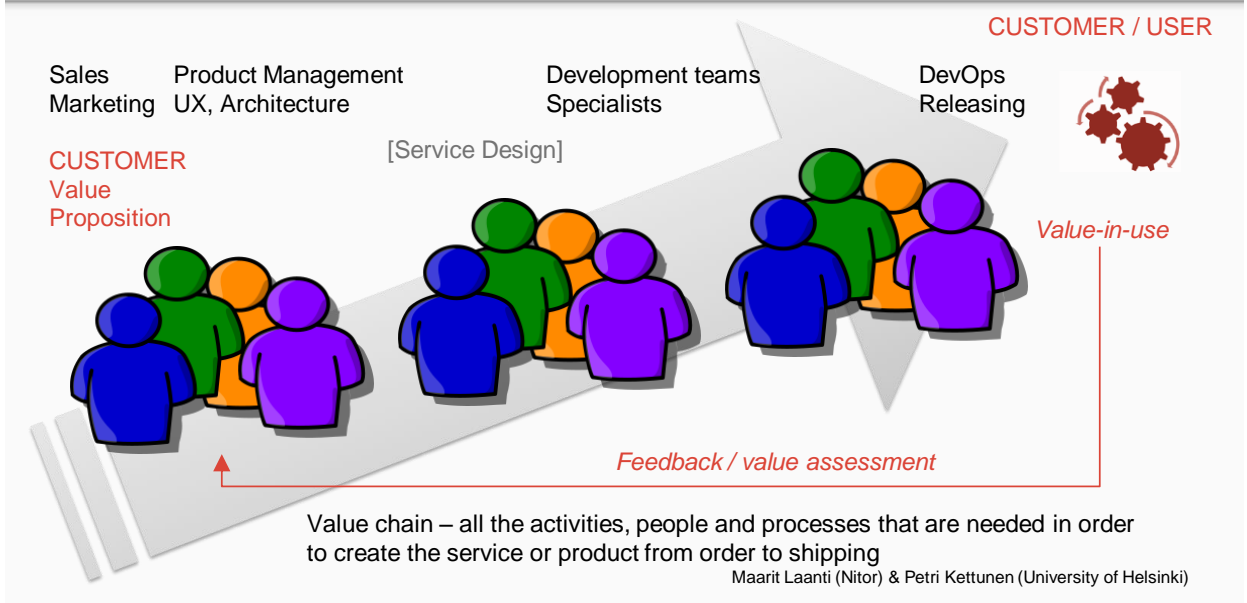
- Digitalization makes the customer software use contexts more complex, and the value of software becomes more intrinsic.
 - ◆ Systems complexity increases, mostly non-controllable (e.g., IoT).
- Software companies must be prepared to master such new concerns in order to be able to serve their customers successfully.
 - ◆ Software organizations may (have to) realize digital transformations internally.
- New kind of structures and roles / competences may be needed to support agile and flexible development needs and goals.
 - ◆ Need to organize for (real-time) continuous software evolution (architecture, organization)

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

DRIVERS AND NEEDS

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

In larger organizations, multiple people and roles are needed to create e2e value



Complexity and Speed

- Complex (large) development organizations tend to have slower decision-making times.
 - Complexity is increased by organizational and technical dependencies (e.g., technical debt).
- Decision-making delays makes the software organization less flexible and non-agile.
- Automation can be used to make it faster (e.g., test automation for developmental software defect detection and correction).
 - E2E transparency becomes apparent.

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

VISION OF FUTURE SOFTWARE ORGANIZATIONS

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Why development and releasing speed matters?

- Every feature and product or service has a window of opportunity – if you are right time on the market you will generate more revenue than those who enter the market later.
- If people are happy with your product / service you can use the revenue to improve the service / product and keep the competitive advantage – it will be harder to enter the market later.
- Lean and agile methods both target to rapid development cycles, fast return on investments and minimum work-in-progress inventory.

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

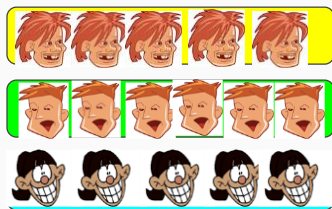
How to achieve continuous development flow and avoid complexity explosion?

1. Information must flow as efficiently as possible to all needed people; preferably broadcasted at regular intervals → avoid communication explosion / missing information
2. New code must be integrated as fast as possible and be immediately available to all people → use of automation (and CI systems) to keep amount of currently ongoing changes in the code small
3. Testing must be planned and executed parallel to development and feedback should be as fast as possible, and the amount of open errors should be kept into minimum → to avoid accumulation of testing debt
4. Architecture should be as simple as and modular as possible → in order to avoid technical debt → FASTER INTEGRATION (2.), BETTER TESTABILITY (3.)
5. Keep documentation up-to-date to in order to avoid documentation debt → INFO FLOW (1.)

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Delays and dependency management

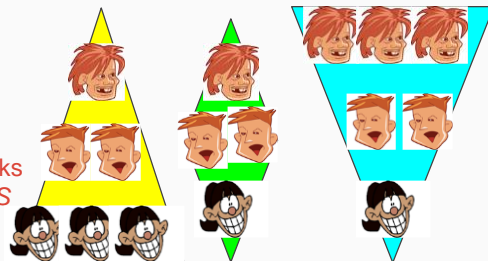
Old way of managing work – to split complex problem into architectural layers and develop each layer separately has caused integration problems and non-fitting components



Work organized as component teams

The new way is to speed up by developing functionality to all layers simultaneously – but integrate frequently and **manage the dependencies during development**

- Frequent feedback; E2E real-time transparency, visibility
- Agile frameworks SAFe and LeSS both promote Feature teams for maximum flow



Work organized as Feature teams

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Key Measurements

Is the only problem we are solving how to achieve faster time-to-market by improving flow?

- **OUTCOME ECONOMY**
 - *right customer value delivered economically*
- **REAL-TIME BUSINESS**
 - *responsiveness to customer feedback and software use data*
 - *sensitivity for new value-creation and business opportunities*

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

TRANSFORMING

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Digitalization changes

Digitalization changes former physical goods ...

into abstract form.

In physical form, the domain knowledge has been essential.

Domain knowledge becomes hygiene; knowledge on meta-level is essential (who builds the platform).

Old players have a challenge to use the digital channels.

Automation reduces transaction cost. Delivering digital (or physical) goods via internet costs less, and subscriptions can happen more often.

The market could change or extend because of digitalization.

New players (having software background) have been able to enter to digital market.

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Digitalization enables

- Cost innovation:
 - Same service or product can be implemented, provided and serviced with a lesser cost but similar quality.
 - Moving digital data costs less than moving physical goods.
 - Software component in goods allows reduction of the production costs compared to physical / hardware components.
 - Shared computational services / more IP-addresses enable to enlarge the networks and computational capacity – and e.g. new business with shared services.
 - Thinking whole chain from production to consumer some steps can be streamlined for business opportunities.
- Business innovation (smart products and services, new business models in digital economy)

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Strategic changes cause need to change value streams and change organization

- Organizing for value:
 - Value streams are your revenue streams: a **fluid** organization should be organized around the value stream
 - An **adaptive** organization should be able to quickly align itself along the new value / revenue stream
 - Cause minimum disturbance in organization and team structures and allow the organization to self-morph into new needed state
 - Allow learning of new (needed) competences
 - A **flexible** organization is extending and relying on to 3rd party suppliers
- Impose new strategic changes with modification in value stream:
 - Value streams may be combined or changed → platforms
 - New startups and ecosystems create new value streams

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Roles in SAFe (Scaled Agile Framework) and in LeSS (Large Scale Scrum)

- SAFe and LeSS both promote the need to organize for maximum flow
 - Both promote Feature teams and Communities of practise
 - Both recognize the need for frequent communication: certain roles have a responsibility for sharing information frequently
 - SAFe promotes Tribes: daily sync on Program level in SAFe
- SAFe Roles scale up:
 - Release train Engineer is über-scrum-master and Product Manager is similar role in Program level as Product Owner is on Team level
- LeSS promotes scaling down to smaller organizations and challenges the number of needed teams and number of product owners

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

RESULTS AND EXPERIENCES

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Current State-of-the-Art

Software-originated agile process models are successfully applied outside software arena:

- Many hardware teams are using Scrum, e.g. Intel is making microcode using Scrum
- E.g. Finnish Broadcasting Company Yle / graphics team is organized as agile teams although making graphics
- 11th State of agile survey (Version One, 2017) reported growth in non-software companies using agile
 - 36 % software only companies

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Future Challenges

- Many (Finnish) companies have trouble in understanding the actual impacts of digital disruption for their businesses (<http://teknologiateollisuus.fi/fi/ajankohtaista/uutiset/digikyvykkyys>).

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

CONCLUSIONS

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Futures of a Complex Software World

- When software is increasingly embedded and ubiquitous, people connect ever more with systems and with each other in many different levels (socio-tech-economic-environmental). Moreover, the connections are often real-time (IoE).
- Those trends bring up totally new opportunities but also challenges for software organizations which have traditionally designed specific IT systems and separate software products.

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

KEY FEATURES OF FUTURE SOFTWARE ORGANIZATIONS (IN AND OUT)

- E2E value
 - ◆ Transparency
 - ◆ Outcome economy
- Organizational integration (information & people & tools)
 - ◆ Reorganizing for digitalization
 - ◆ External integration (customers / users & partners & ecosystems / platforms)
- Software-intensity (decisions & roles & competencies)
 - ◆ Automation
 - ◆ Data

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Transformation: What does this mean?

GOALS	MEANS
<ul style="list-style-type: none"> If you are a big company: How to make your structures lightweight, and increase flow through the system? 	By fluidity : <ul style="list-style-type: none"> Decouple architectures and teams.
<ul style="list-style-type: none"> If you are a traditional company: Understand the opportunities of cost reduction and innovation 	By adaptability : <ul style="list-style-type: none"> with the use of software
<ul style="list-style-type: none"> In ecosystem (digital economy): 	By flexibility : <ul style="list-style-type: none"> Build and streamline value streams for cost efficiency and new value.

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

Industrialism created hierarchies and benefit of scale.

Internet created networks and benefit of connectivity and digitalization.

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)

CONTACTS

Maarit Laanti

Nitor
Aleksanterinkatu 44 A 8
FI-00100 HELSINKI
Finland

E-mail: [maarit.laanti\(at\)nitor.fi](mailto:maarit.laanti@nitor.fi)

Petri Kettunen

University of Helsinki
Department of Computer Science
P.O. Box 68
FI-00014 UNIVERSITY OF HELSINKI
Finland

E-mail: [petri.kettunen\(at\)cs.helsinki.fi](mailto:petri.kettunen@cs.helsinki.fi)

Maarit Laanti (Nitor) & Petri Kettunen (University of Helsinki)