



GOSEI



GOSEI

# Scaling Agility with Large-Scale Scrum

*XP2016*

*Ran Nyman*

# Ran Nyman



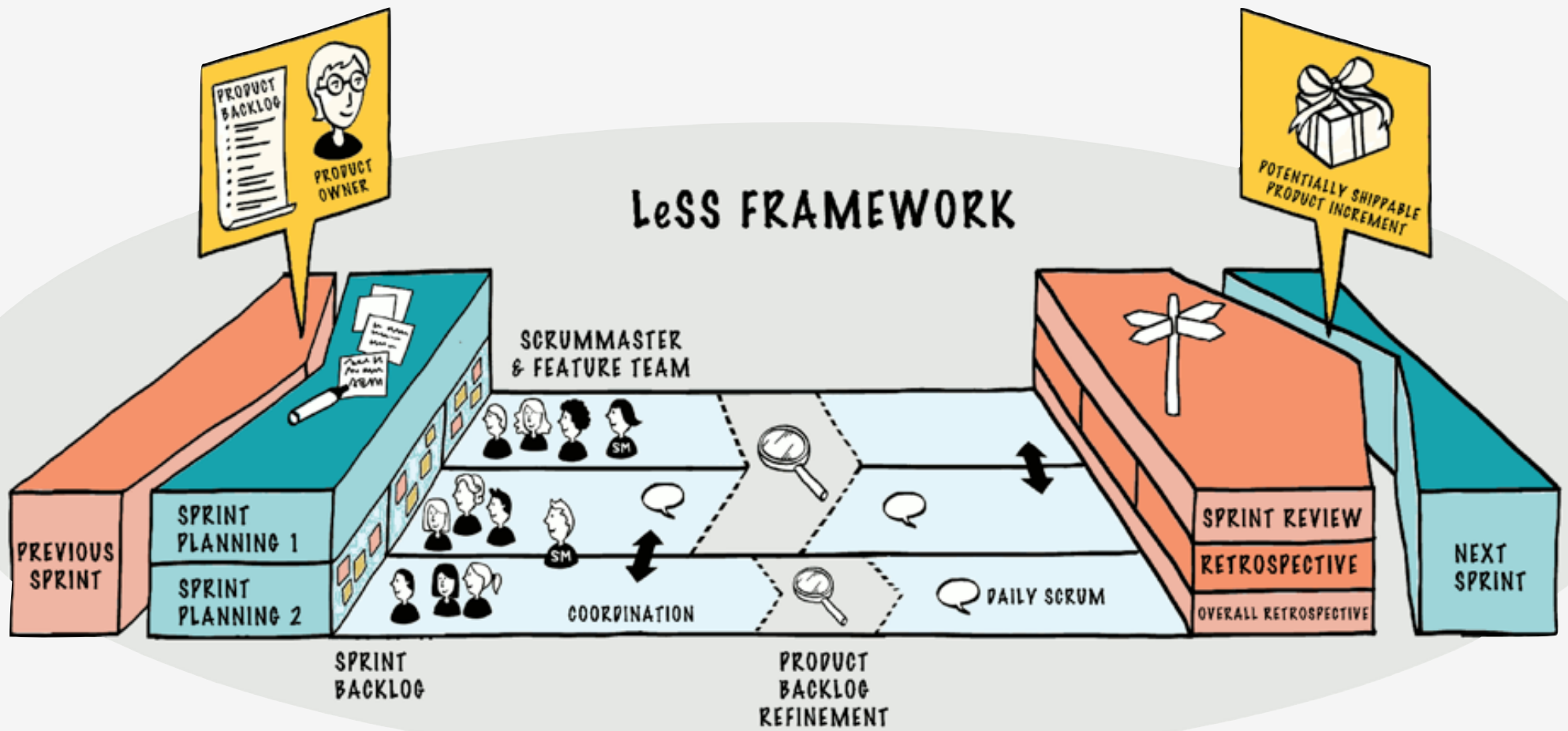
Worked with LeSS since 2005

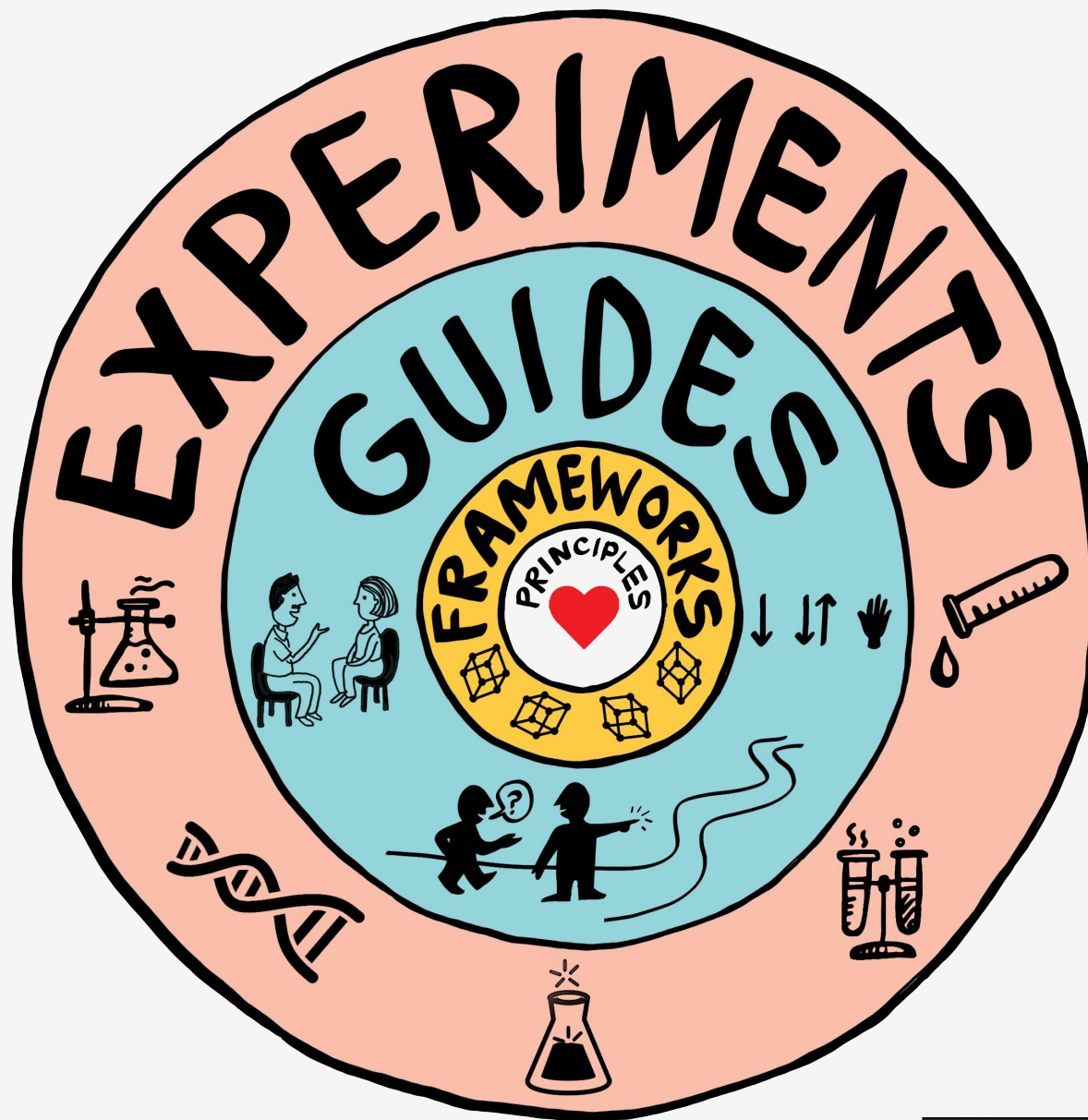
MSc in Computer Science 1999

Since 1995 in professional SW development



# Large-Scale Scrum







LEAN THINKING



SYSTEMS THINKING



PRINCIPLES



COACHING



ADOPTION



CONTINUOUS IMPROVEMENT

FEATURE TEAMS



TEAMS



STRUCTURE



ORGANIZATION



COMMUNITIES



TECHNICAL EXCELLENCE



CONTINUOUS INTEGRATION



ARCHITECTURE & DESIGN



ROLE OF MANAGERS

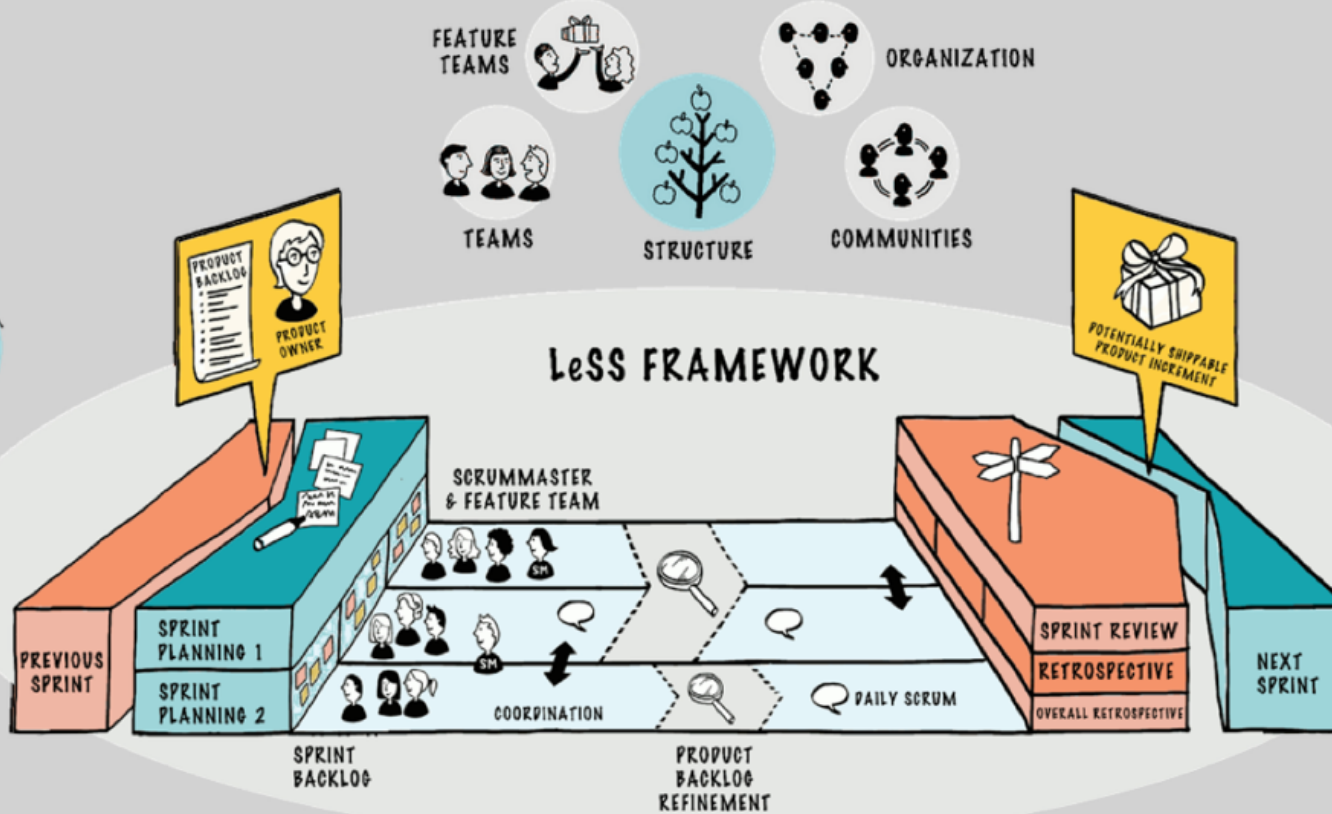


MANAGEMENT



GO SEE

# LeSS FRAMEWORK



PRODUCT OWNER TEAM



LeSS HUGE



REQUIREMENT AREAS

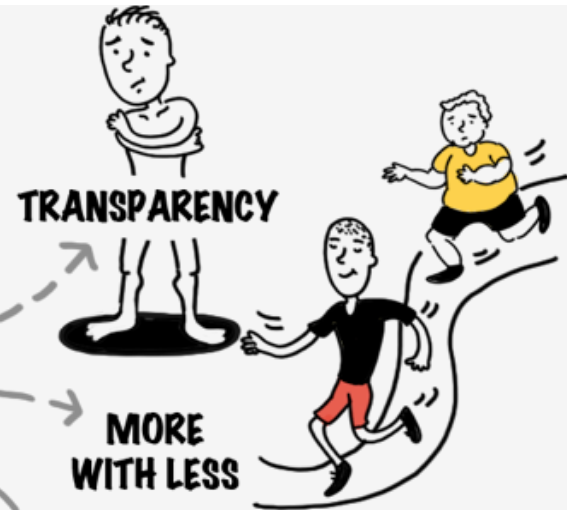
CC BY-NC-ND



**QUEUEING THEORY**



**LARGE-SCALE  
SCRUM IS SCRUM**



**TRANSPARENCY**

**MORE  
WITH LESS**



**EMPIRICAL  
PROCESS CONTROL**



**SYSTEMS  
THINKING**



**LEAN  
THINKING**



**CONTINUOUS IMPROVEMENT  
TOWARDS PERFECTION**



**WHOLE  
PRODUCT  
FOCUS**



**CUSTOMER  
CENTRIC**





**Scaling Agility  
is  
Descaling Organisations**

# LeSS Case 1

# NOKIA

## Nokia Mobile Phones



## Nokia Networks



# Browsing Gateway in 2004

Working with incremental sequential development

- 3 months increments with 3 project managers coordinating
  - Weekly manual build (no CI)
  - Individual component responsibilities
  - Testing and development separated in different wings of building
  - No meaningful automated e2e testing
  - Manual testing using scripts
- > Releases constantly late with quality problems

# Introducing LeSS in 2005

Management buy-in

Targeted trainings to key influencers

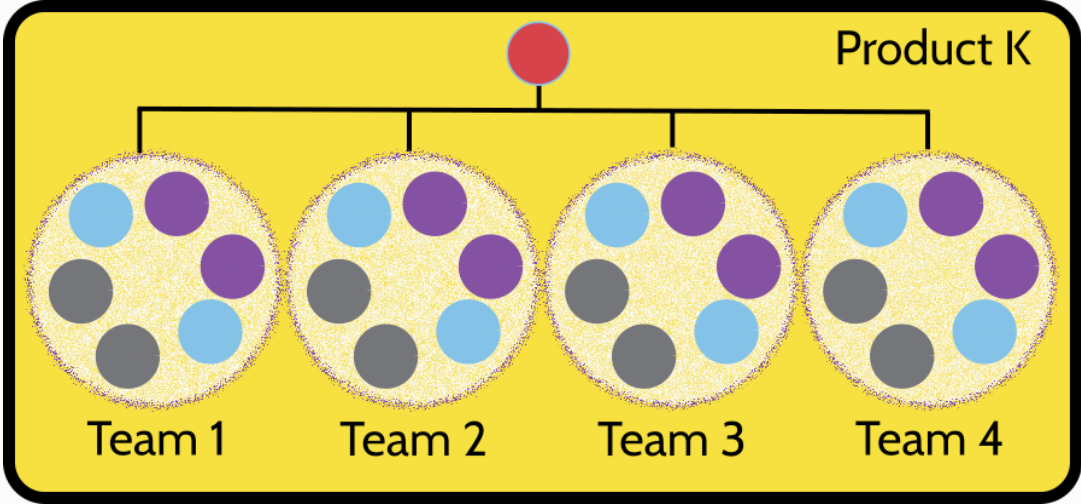
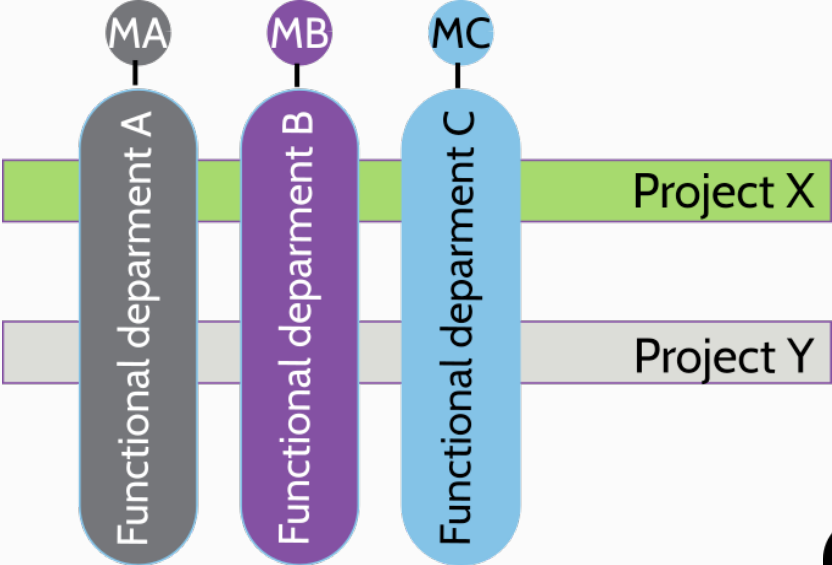
Basic Agile and Scrum training for whole organization

Creating feature teams

Structural changes in management

Start working with teams in next release

# Focus from Internal Structure to Customer

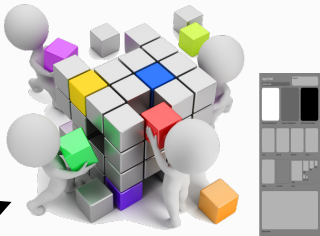


# LeSS Structure

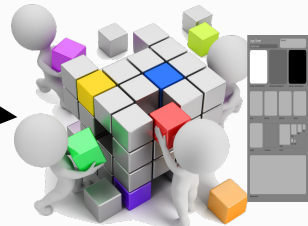
**Product  
Owner**



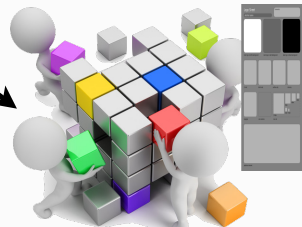
**Product  
Backlog**



**Team a**



**Team b**



**Team c**



**ScrumMaster**

**One Sprint  
Backlog  
per Team**



**ScrumMaster**

# From Physical Silos to Collaboration





# From Manual Build to CI

No other way of keeping SW in shape

Branching did not work

Do not use ClearCase



The screenshot displays the CruiseControl Dashboard interface. At the top, it shows the title "CruiseControl Dashboard" and the last update time "Last Update: Oct 30, 12:57 pm (Refresh)". The dashboard is organized into a table with columns for "Project", "Status", "Unit Tests", and "Force Build". The projects are grouped by "Build Center" (Tiger, Mustang, Liberty). Each project entry shows a "Passed" status, the number of unit tests (e.g., 10 or 12), and a "Build" button. A small rocket icon is visible on the left side of the dashboard.

Project	Status	Unit Tests	Force Build
Build Center: Tiger			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
connectfour3	Passed	11 (100%)	Build
Build Center: Mustang			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
Build Center: Liberty			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
connectfour3	Passed	11 (100%)	Build
Build Center: Mustang			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
Build Center: Tiger			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
connectfour3	Passed	11 (100%)	Build
Build Center: Mustang			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
Build Center: Tiger			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
connectfour3	Passed	11 (100%)	Build
Build Center: Liberty			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
Build Center: Tiger			
connectfour	Passed	10 (100%)	Build
connectfour2	Passed	12 (100%)	Build
connectfour3	Passed	11 (100%)	Build

# From Manual Testing to A-TDD

Getting rid of scripted manual tests

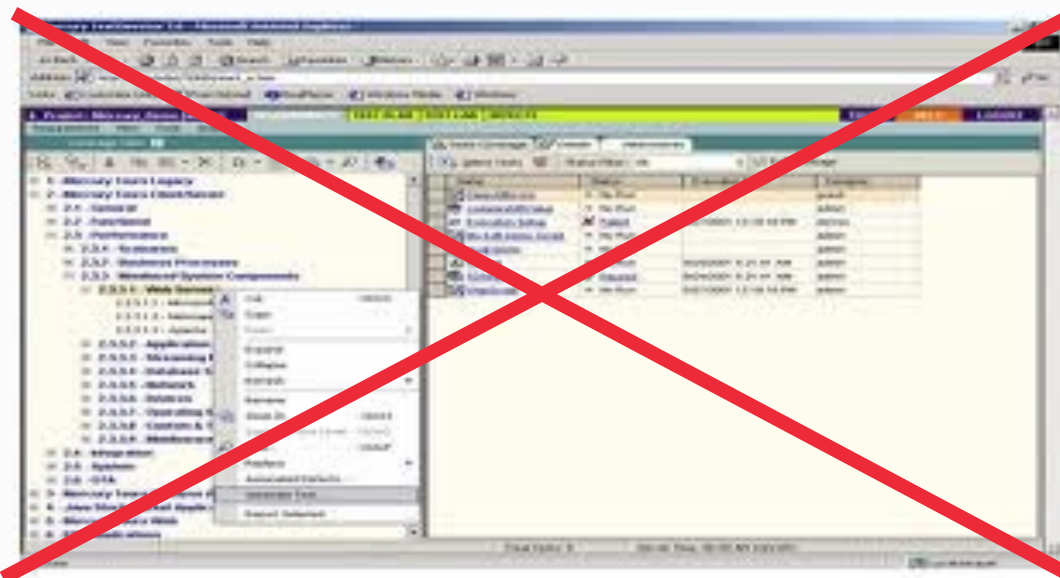
- -> Automate them
- But do not offshore automation!
- But do not create your own framework!

Performance testing?

- We ran it continuously

Stability testing?

- We ran it continuously



# From Project Managers to One Product Owner

One single real PO that prioritises

- Teams clarify directly with stakeholders

Project Managers:

- Kept them away from teams
- -> Not much work



# Results

Great working spirit

100% test automation

Product version 2.1 launched before scheduled date

- PSPI
- Strong Definition of Done

# LeSS Case 2



# Starting Development

# Starting Phase

## Management support

- From R&D Head and BL Head

Two teams instead of one

No legacy code

- How often can you start from scratch?

## First steps

- Create Product Backlog
- Build initial architecture
- Decide tooling and development practices



# Setting up Teams

Getting buy-in from teams

- Feature teams of component teams?

Where to get ScrumMasters?

What do with managers?

Mixing teams a good approach?

# Initial LeSS with 2 Teams

# LeSS with 2 Teams

Initial Product Backlog Creation

Initial Architecture

Joint Sprint Planning 1 & 2

Coordination during Sprint

- Scrum of Scrum
- Just go and talk

Joint Sprint Review

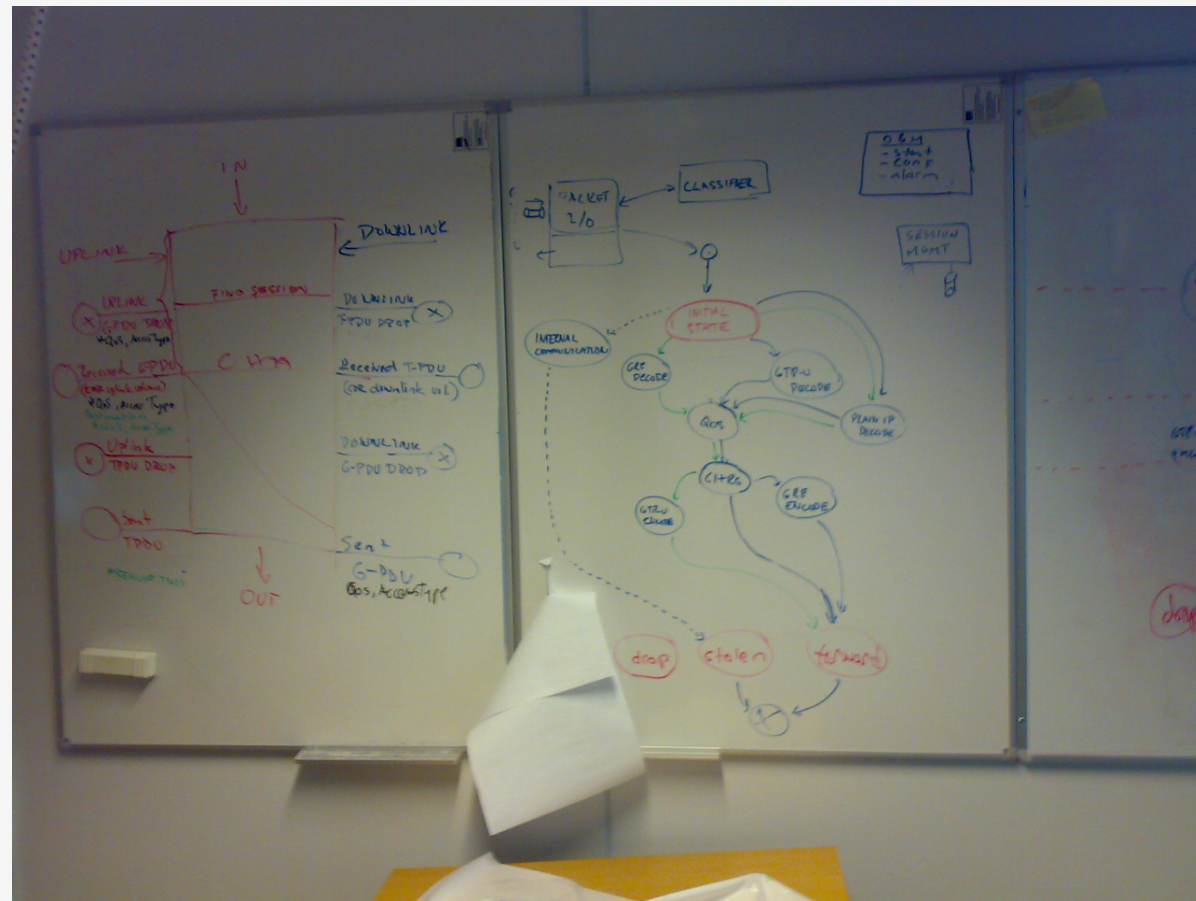
Team Retrospectives

Overall Sprint Retrospective



# Initial Product Backlog Creation

# Initial Architecture





# Joint Sprint Planning 1 and 2

# Initial Team Room





# Better Team Rooms



# Joint Sprint Review



# Overall Sprint Retrospective



# Growing Phase 1

# Growing Phase 1

Adding 4 more teams to existing site

- Non agile background
- Resistance to new ways of
- Coaching helped

# LeSS with up to 6 Teams

Joint Sprint Planning 1 and 2 did not work anymore

- Sprint Planning 1 with Team Representatives
- Sprint Planning 2 within multi/single teams

Joint Product Backlog Refinement did not work

- Joint PBR with team representatives
- Product Backlog Refinement multi/single teams

Joint Sprint Review worked still



# Introduction of Bug Tracking System

# Remarks

Growing from 2 to 6 teams was easy

- No significant changes needed

Coaching helps for new teams

Do not underestimate resistance

# Growing Phase 2

## From LeSS to LeSS Huge

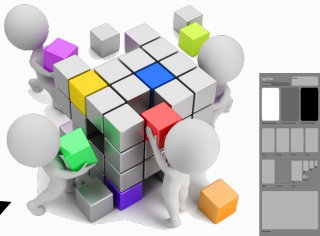


# LeSS Structure

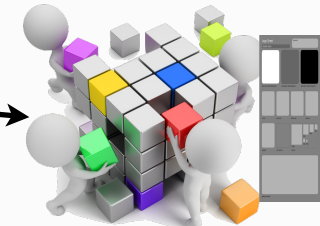
**Product  
Owner**



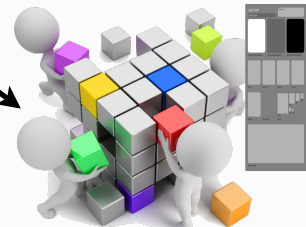
**Product  
Backlog**



**Team a**



**Team b**



**Team c**



**ScrumMaster**



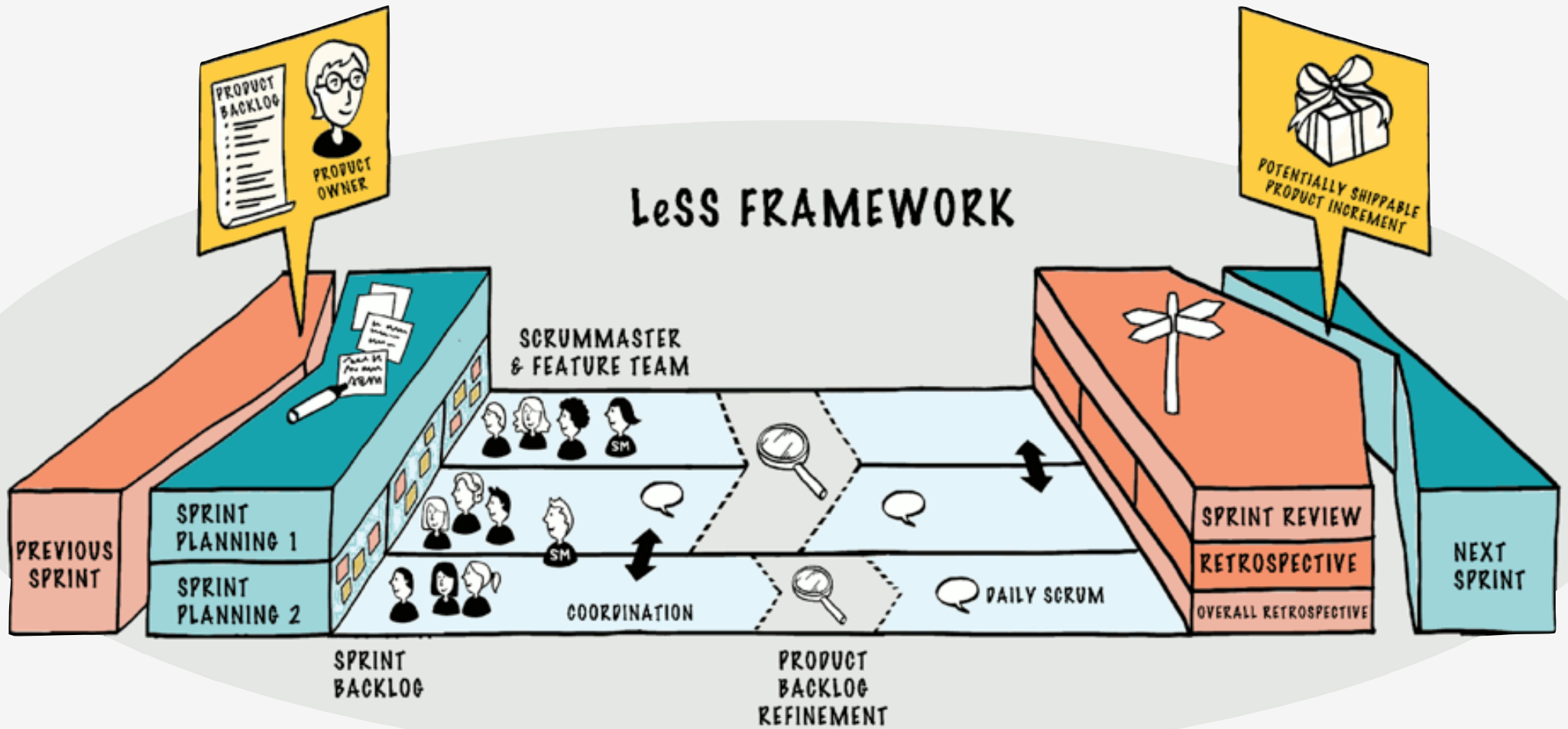
**One Sprint  
Backlog  
per Team**



**ScrumMaster**



# LeSS FRAMEWORK





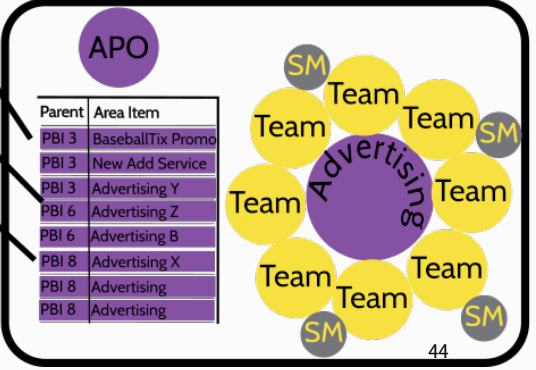
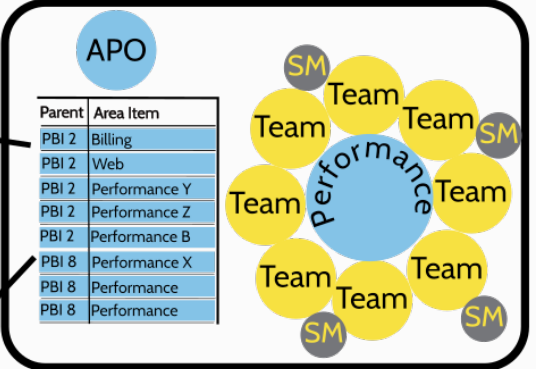
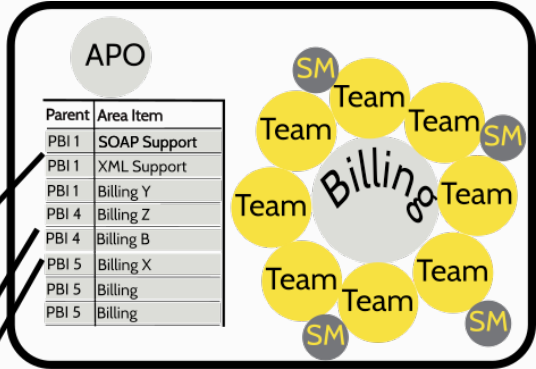
# Moving to LeSS Huge

**Adds:**

- Requirement Areas
- Area Product Owner
- Area Product Backlog

PO

Items	Requirement Area
PBI 1	Billing
PBI 2	Performance
PBI 3	Advertising
PBI 4	Billing
PBI 5	Billing
PBI 6	Advertising
PBI 7	Advertising
PBI 8	Performance



# LeSS Huge

## Introduced Requirement Areas

- 1-2 Teams per RA

## Area Product Owners?

- From Product Management but not real Area Product Owners
- Clarification with them but
- Prioritisation and feedback from PO
- -> Feature experts not real Area POs

# LeSS Huge in Action

Common cadence to all teams

Product Backlog refinement per Requirement Area

Sprint Review to Sequential

- PO + Feature Experts visiting each team

# Observations

Overall Retrospective

- How to get improvements done and keep people motivated

What do with project managers?

Performance testing?

# Analysis

Impediment service

Training

Managers as development team worked fabulously

Real Area Product Owners needed for

- Inspect and adapt
- Feedback and motivation for teams



# Employee Feedback

We would have not succeeded without LeSS

This is the only way of building products. We do not want to go back to sequential development.

- More collaboration, freedom, discussions, impact on product
- Seeing the results with fast feedback

## Challenges

- How to keep architecture in good shape
- Seeing the whole difficult

# Management Feedback

“As a result of our Agility in developing products, we have been able to demonstrate the product capabilities at an early stage of the development cycle itself, helping us to win new customers cases.”

Head of Business-line

# Conclusions

# Common Success Factors

Support from top and bottom

Scrum and LeSS

Structural changes

One Product Owner

One Product Backlog

Continuous experimentation

Passionate ScrumMasters

Education and Coaching

No project managers interfering

Strong DoD

PSPI after each and every Sprint

Practices:

- Continuous Integration
- Main branch development
- Focus on testing:
  - Unit
  - Automated Acceptance
  - Performance and stability

Q&A

[ran.nyman@gosei.fi](mailto:ran.nyman@gosei.fi)

[@ran\\_nyman](#)