SCALING AGILE IN MECHATRONICS-DRIVEN COMPANIES

ULRIK EKLUND, MALMÖ UNIVERSITY
CHRISTIAN BERGER, UNIVERSITY OF GOTHENBURG

SOFTWARE CENTER PROJECT

Goal

to identify how to scale agile software development beyond individual teams in large organizations presently focusing on the development, manufacturing and delivery of their physical products

Industrial Partners:

Volvo Car Group

AB Volvo

Grundfos

SAAB Electronic Defence Systems

Axis

Tetra Pak

CONTEXT

Many organizations have successfully introduced agile development on the team level

- Individual teams define their own ways of working to facilitate speed, short iterations, and delivery quality.

But agile speed is not seen in the organisation as a whole! Causes may include

- S/W subcontractors tied up in sourcing agreements,
- software interfacing with hardware and mechanics, and
- certification processes

VOLVO PRODUCT DEVELOPMENT SYSTEM

Overview of Volvo Cars stage-gate planning omitted on purpose!

For a very high level overview see e.g:

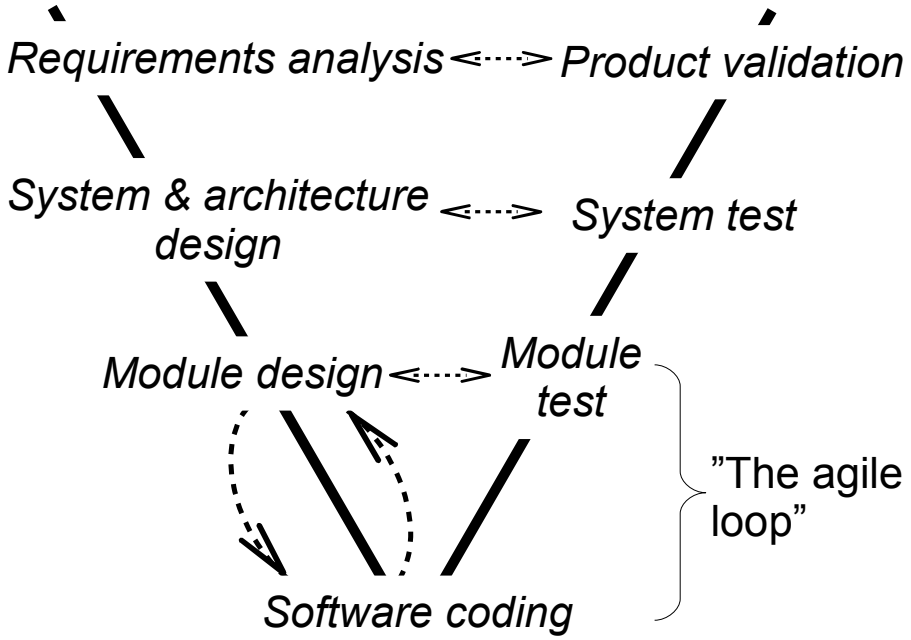
http://www3.volvo.com/investors/finrep/sr11/en/enviromentalrespons/productdevelopment/globalproductdevelo/pdf/Global_product_development_process.pdf

THE LOCAL ”AGILE LOOP”

Projects where manufacturing and hardware development have long lead-times (years instead of weeks)

In contrast individual software teams are able to reprioritize and implement features in 2-4 week cycles

Effort is spent on aligning the practices of the individual teams to the overall R&D process



CASE STUDY

Method

- Empirical study
- Mainly qualitative data collection
- Inductive approach to analysis

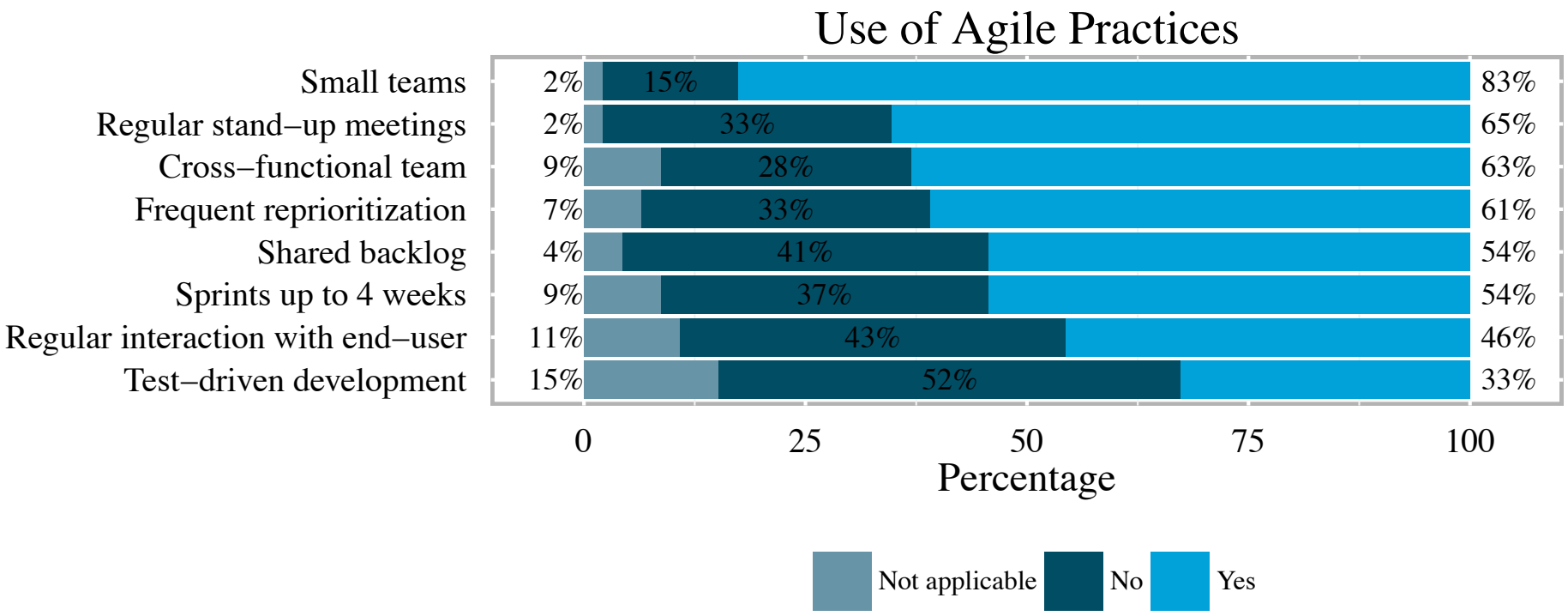
Two phases

1. Investigate expectations and challenges

   Reported at XP conference 2015

2. Identify checklist of practices and actions

SURVEY RESULT: USE OF AGILE PRACTICES

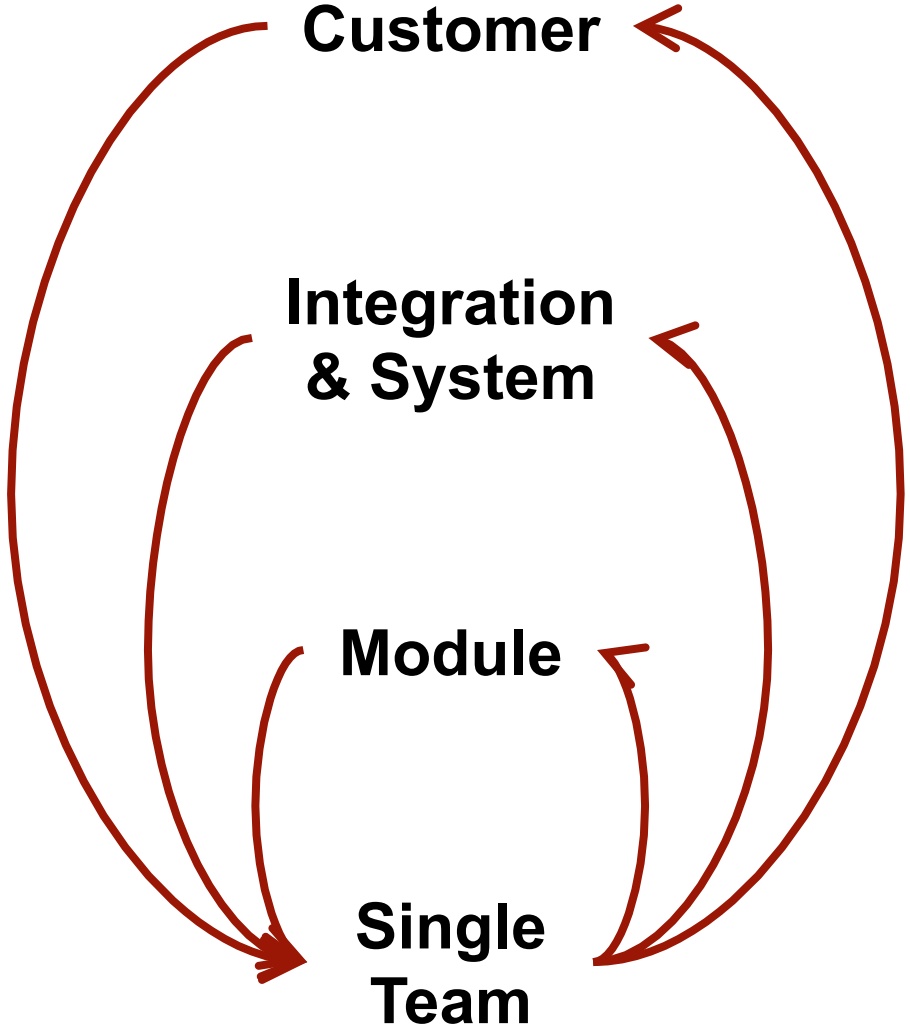


SURVEY RESULT: PERCEPTION OF AGILE

TOP EXPECTED BENEFITS

1. Higher quality
2. Faster time−to−market
3. Shortening lead−times
4. Maximize output from existing development resources
5. Minimize risk to develop wrong things

COLLABORATION & FEEDBACK



Once per project

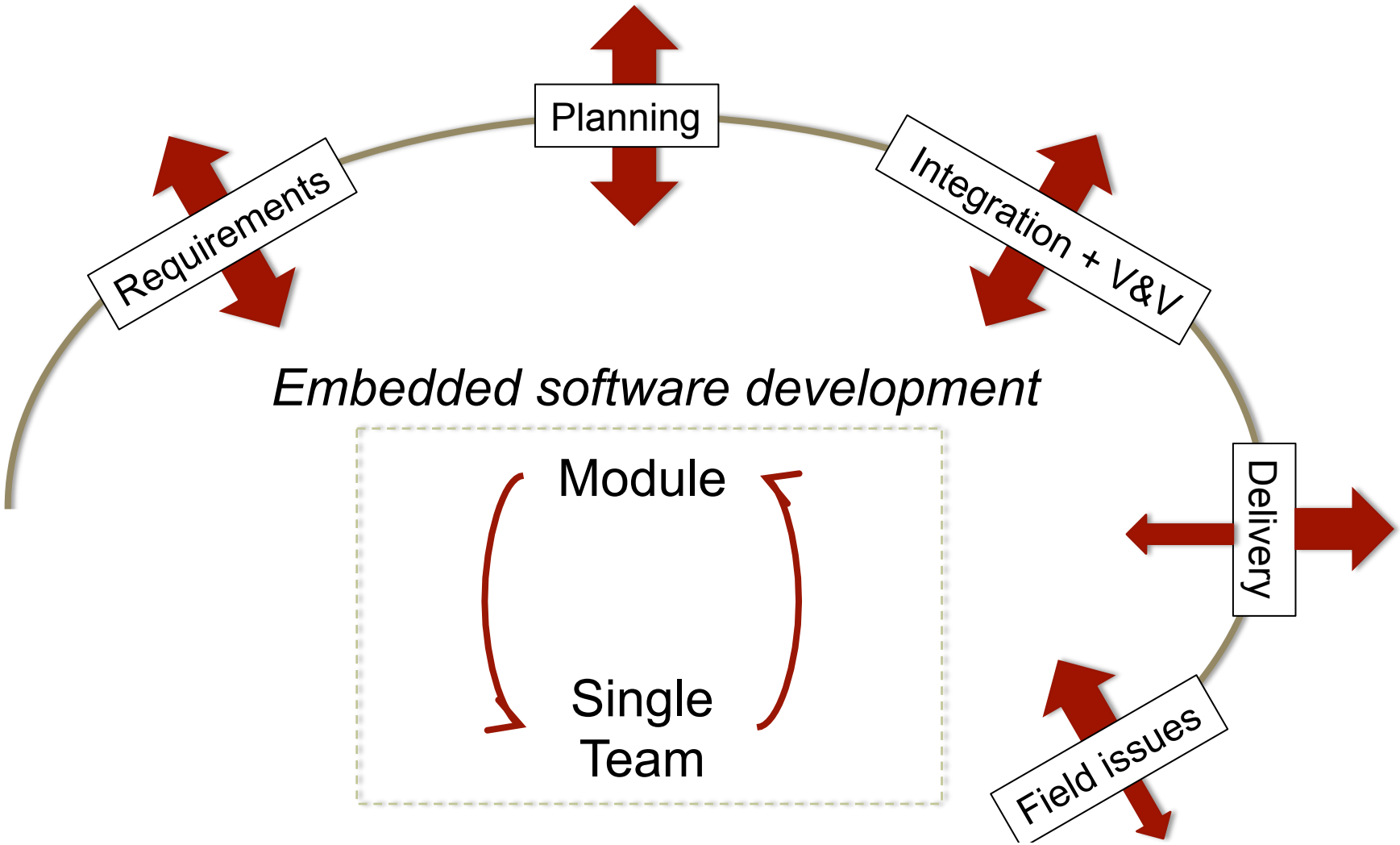
3-6 months

2-4 weeks

TOP CHALLENGES

1. Missing flexibility in current test facilities
2. Better collaboration between all disciplines
3. Changing the mindset in the organization
4. Differentiate lead-times

COLLABORATION & FEEDBACK



LARGE-SCALE AGILE PRACTICES

A checklist of 208 agile practices

- **based on the empirical data in the study, of which**
- **26 practices were unique to the mechatronic domain**

Conclusion: The mechatronic domain should apply known proven practices similar to other domains in large scale agile development

ORGANISED CHECKLIST ACCORDING TO:

Agile value & agile maturity level	Embrace Change to Deliver Customer Value	Plan and Deliver Software Frequently	Human Centricity	Technical Excellence	Customer Collaboration
Encompassing					
Adaptive					
Effective					
Evolutionary					
Collaborative					

I. Stojanov, O. Turetken, and J. J. M. Trienekens, *'A Maturity Model for Scaling Agile Development'*, in 41st Euromicro Conference on Software Engineering and Advanced Applications, 2015, pp. 446–453. DOI: 10.1109/SEAA.2015.29

COLLABORATIVE MATURITY LEVEL

Human Centricity

- Having an agile process to adjust technical intefaces
- Multidisciplinary teams

Technical Excellence

- Have multidisciplinary and joint doumentation
- Quick and dirty hardware available to test software functionality
- Simplify technical interfaces
- Software available to use in tests of hardware development

EVOLUTIONARY MATURITY LEVEL

Human Centricity

- Minimise suppplier lead-times

Plan and Deliver Software Frequently

- Don't modify off-the-shelf products

Technical Excellence

- Identify the Minimum Viable Product to do software integration
- Speedy deployment of test software to the (prototype) product

EFFECTIVE MATURITY LEVEL

Human Centricity

- Do not involve suppliers
- Don't isolate disciplines

Technical Excellence

- Do not depend on manual deployment
- Integration is a continuous activity (every 4 weeks)
- Move complexity from mechanics to software / moves lead-time
- Move towards platforms
- Software development is allowed to deliver a new release to production every sprint
- Target software is put as last on the hardware in production

ADAPTIVE MATURITY LEVEL

Embrace Change to Deliver Customer Value

- Reduce variant complexity

Plan and Deliver Software Frequently

- Front-loading of the development process to stream-line industrialisation is avoided
- Not using the same planning/project gates for HW and SW
- Reduce variant complexity

ADAPTIVE MATURITY LEVEL

Technical Excellence

- Allow for integrations of not the full product (e.g. by simulations)
- Minimise the number of point of contacts between SW, HW and mechanics
- Reduce variant complexity (physical as well)

ENCOMPASSING MATURITY LEVEL

Customer Collaboration

- **Allow for software deployment after production**

GOLDEN RULE OF INTEGRATIONS

A simple rule of thumb for integrations:

- The cycle time for full integrations between software, hardware and mechanics should be **no longer than 4 weeks**
- A shorter cycle time (i.e. continuous integration on a product level) benefits software development, but not the other disciplines.

- KPI for improvement: **How many full integrations are done in a project of a certain length?**