SESG Summary and conclusion

- 25-11-2021 @ USN Kongsberg
- 2pm – 5pm

SESG numbers and data

- SESG facilitators: 2
- Coffee with cookies available: Yes ☺
- Duration: 3 hours
- Participants: ~60 (people from industry and academia, including SESG facilitators)
- Presentations: 2 (23 slides total)
- Presenters: 2 (2 from industry)
- Workshop groups: 8 (4-6 people in each group)
- Posters with group feedback

SESG event 25th November, 14:00-17:00

Title: Systems Integration Challenges

Brief description:

Systems Integration is an activity to reduce risk of undesired and unforeseen emergence as early as possible. When components in the system start interacting, and start interacting with its environment, then the system should behave and perform as architects and designers expect. Since our knowledge about the system and its environment is limited, unforeseen behaviour may emerge. If we find such undesired behaviour and performance just before we start system testing, then we have barely means to mitigate. Hence, the focus on systems integration is to verify and validate the system specification and design as early as possible.

Speakers:

- Truls Øhrn, KDA
- Anders Fuglesteg Nilsen, YetiMove

Introduction video from TNO-ESI:

- https://vimeo.com/244302409

Pre-meeting inputs

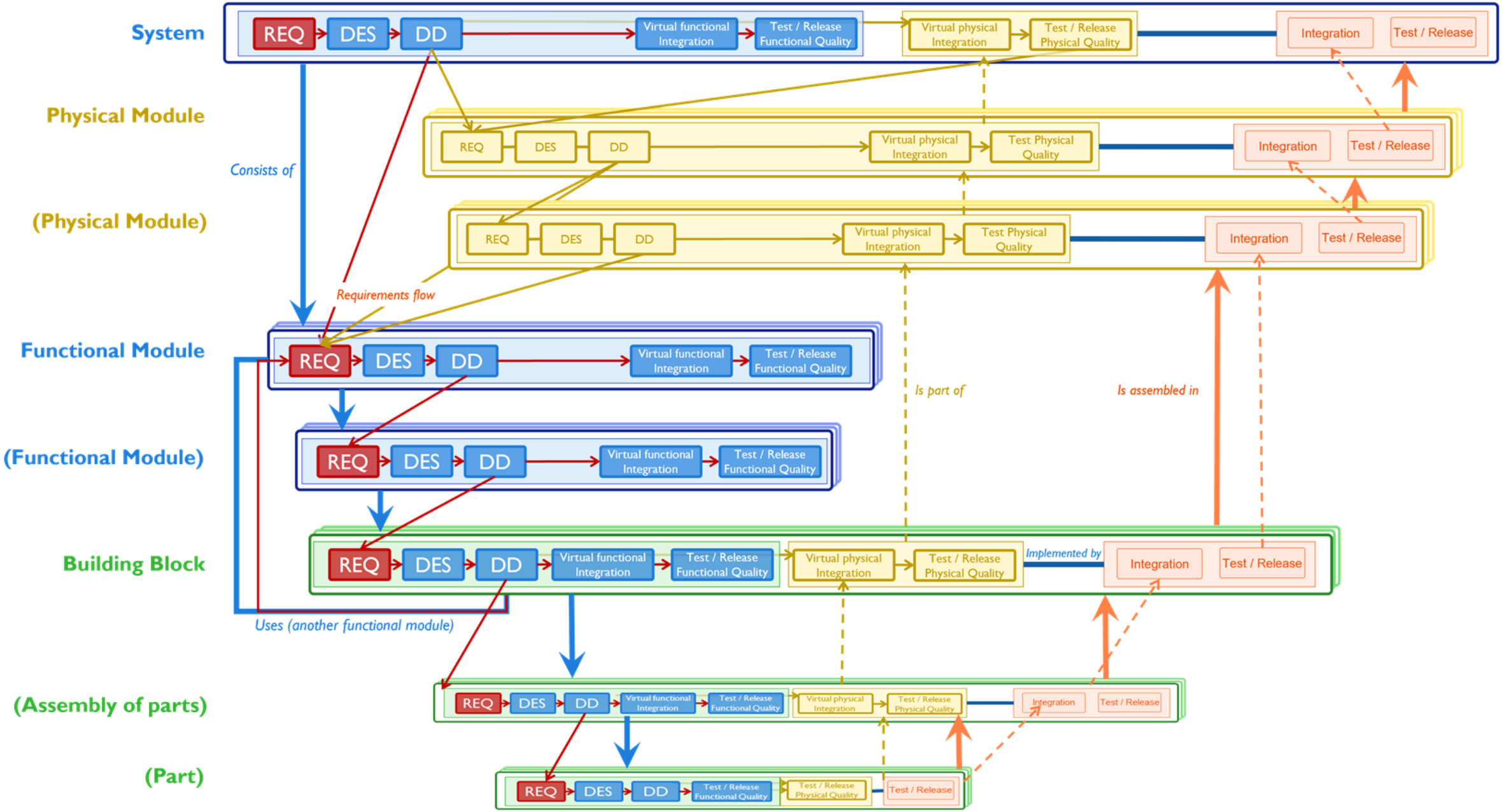
The invitation and the intro video triggered Cees Michielsen from ASML to send me his reflections on how the concurrent development of the **functional** and **physical** decompositions may trigger complications in the systems integration.

Properties like volume and mass require the physical decomposition, while properties like accuracy and thoughput need the functional decomposition.

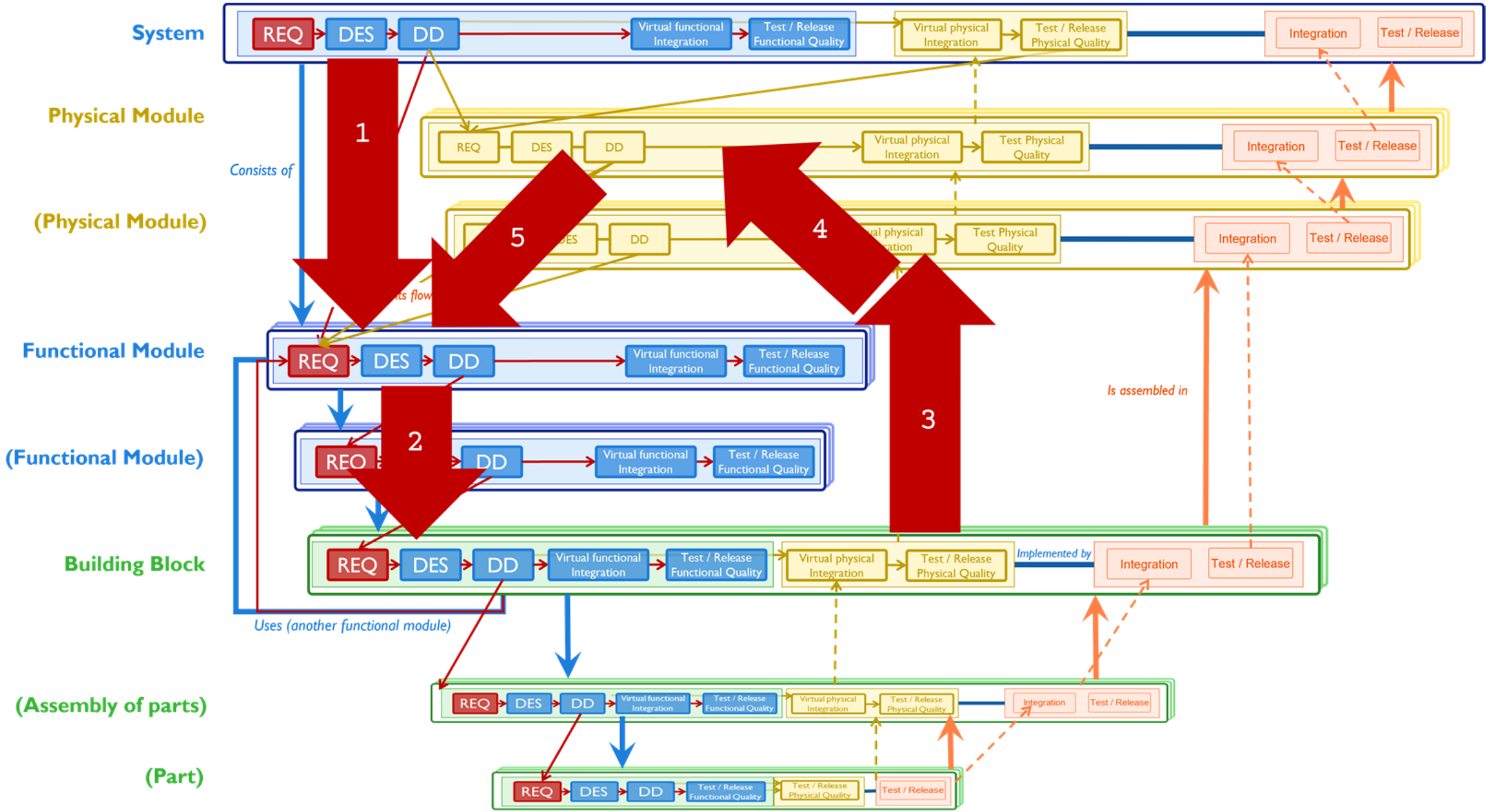
The next 3 slides show Cees' diagrams on ASML's way of working.

TNO-ESI is teaching Systems Integration at ASML as in-company course

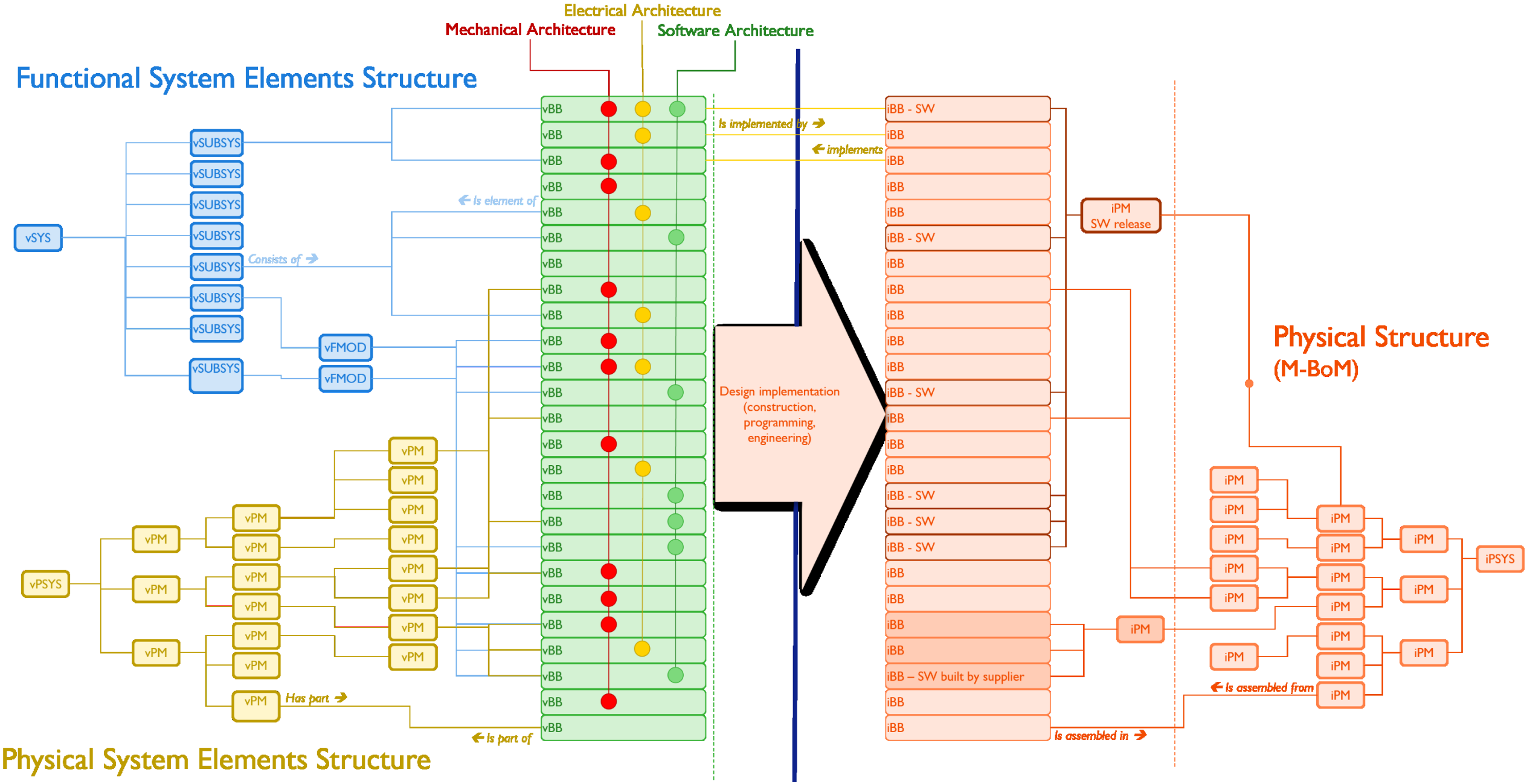
Functional and physical decomposition



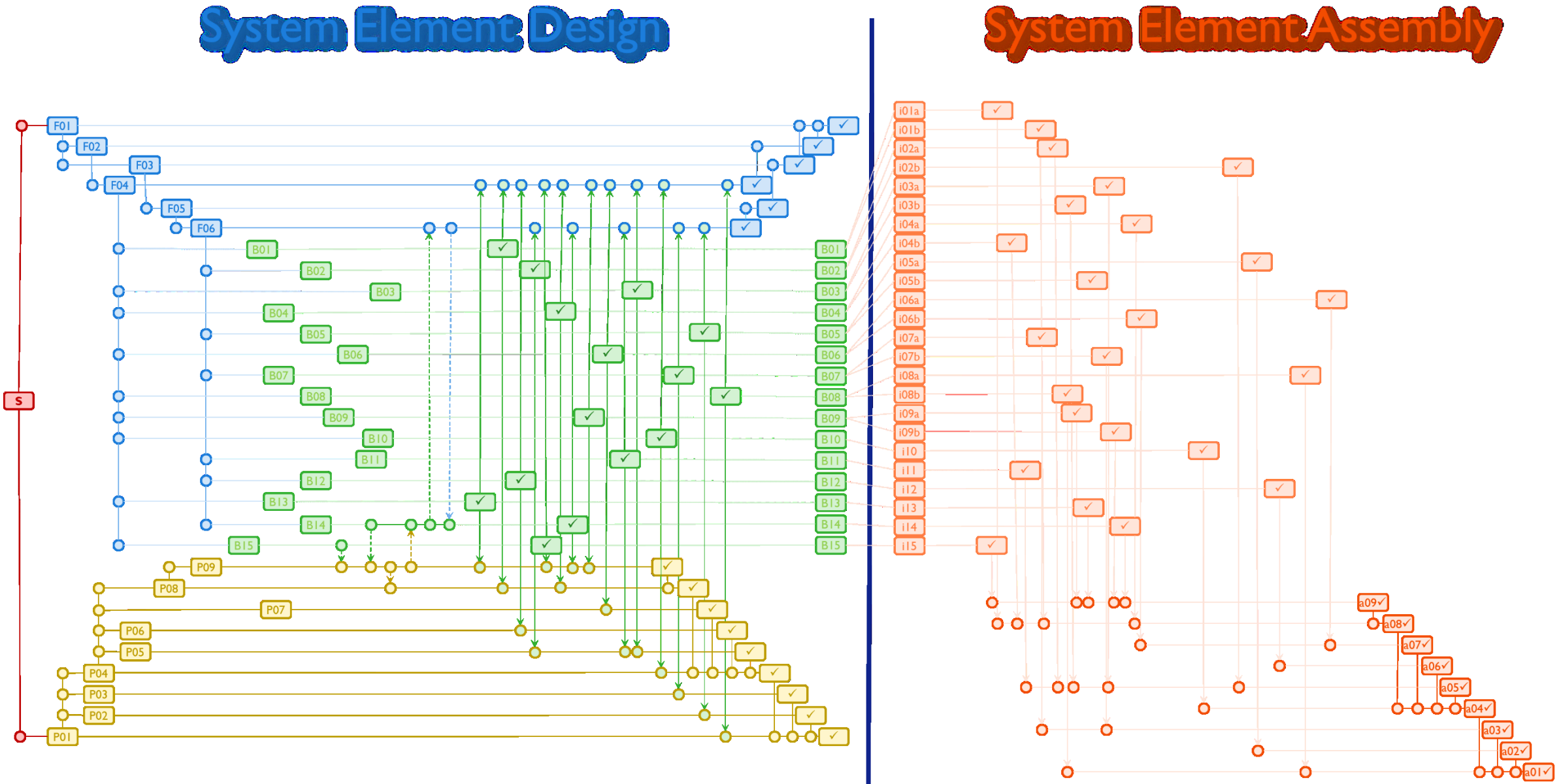
Design Workflow



System Architectures for Configuration Management



Dynamics of the design process

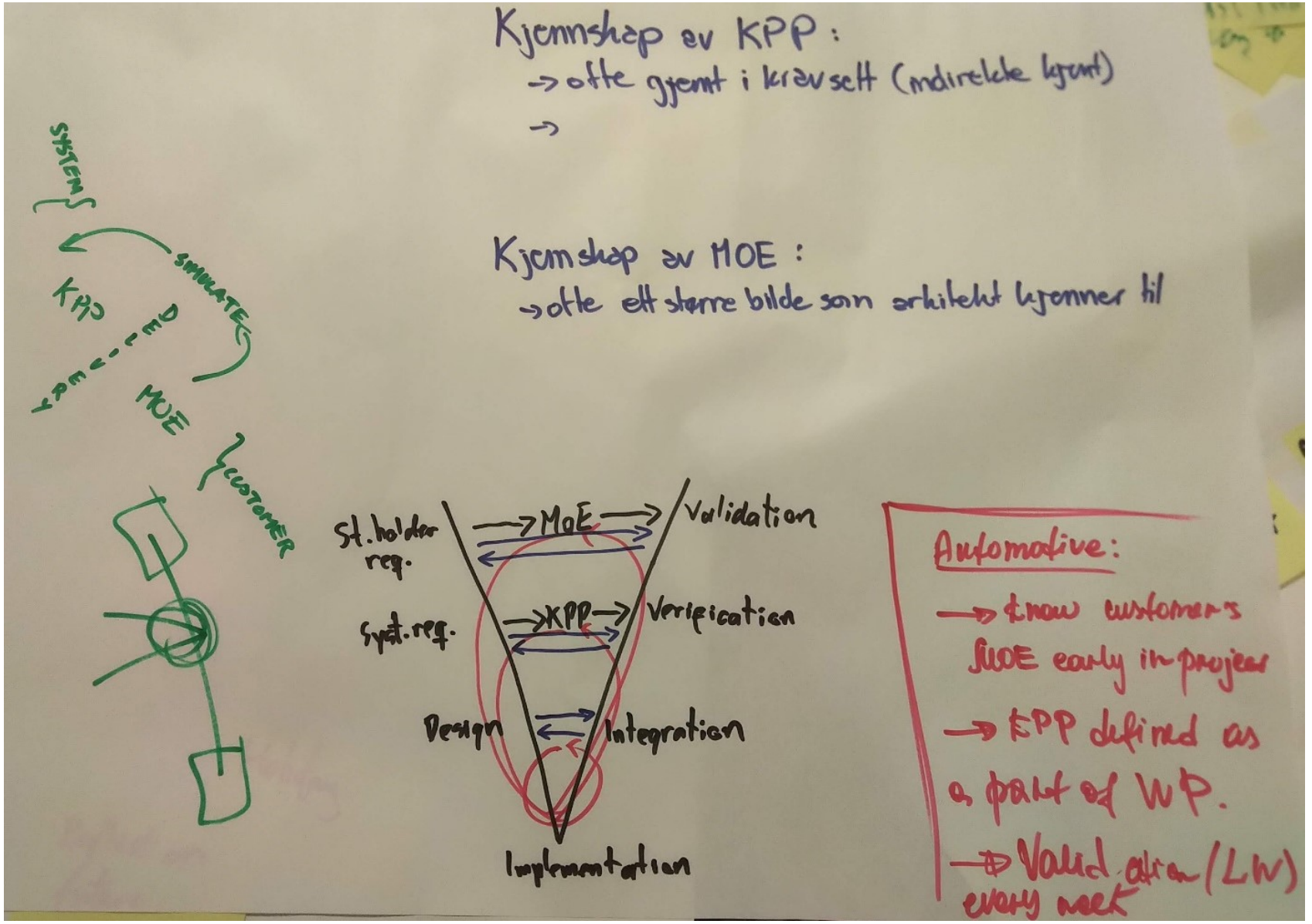


Initiating questions for group work

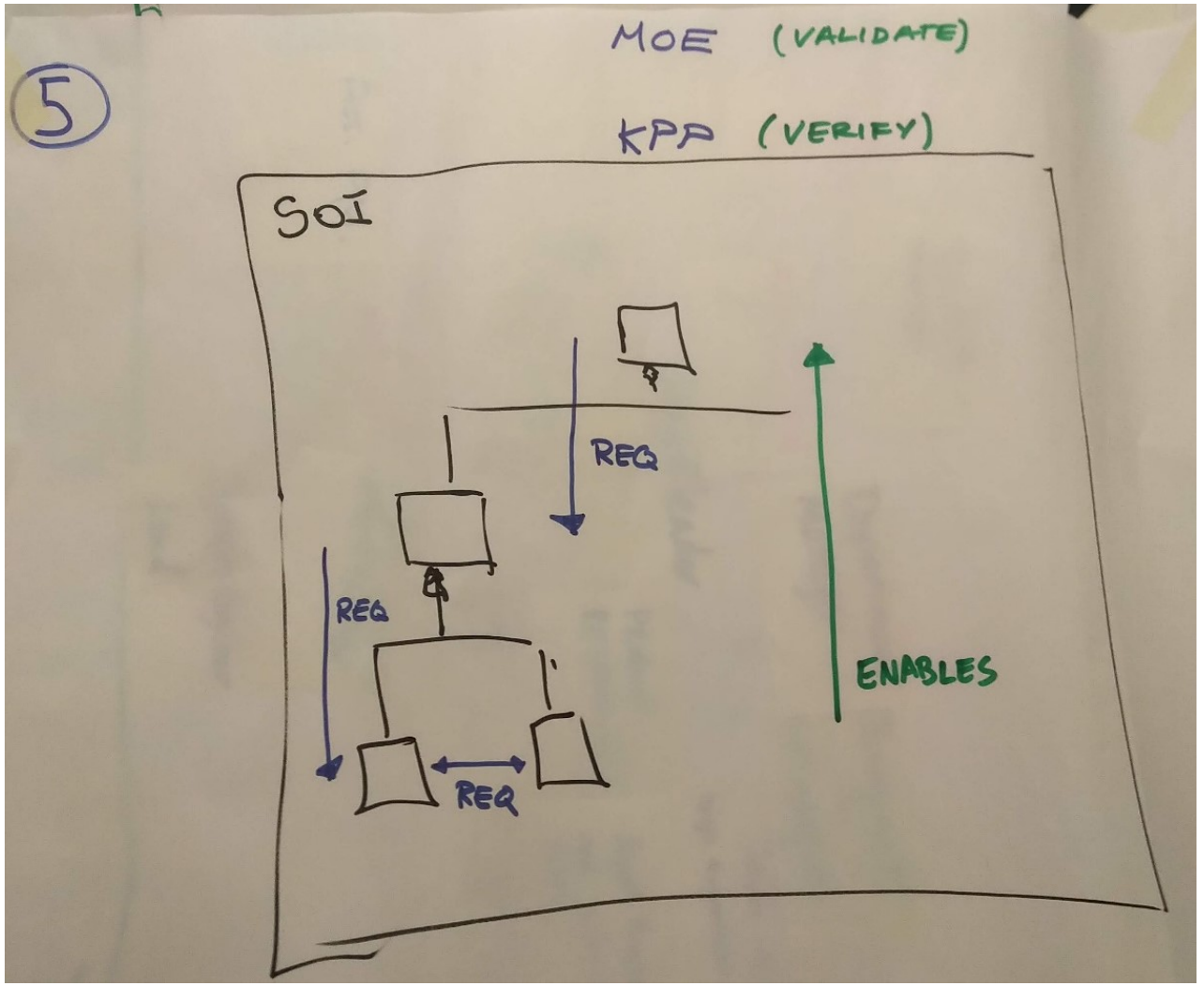
How well do project members know and focus on the **Key Performance Parameters** (@system boundary) and **Measures of Effectiveness** (customer space)?

How early are they visible?

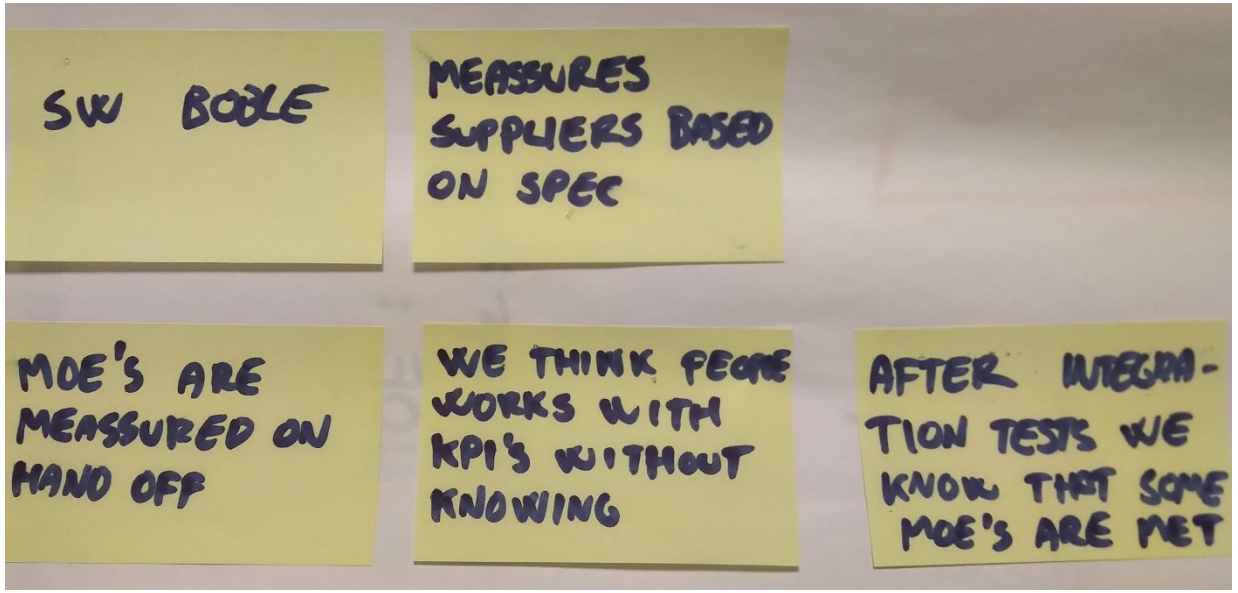
Breakout result (1)

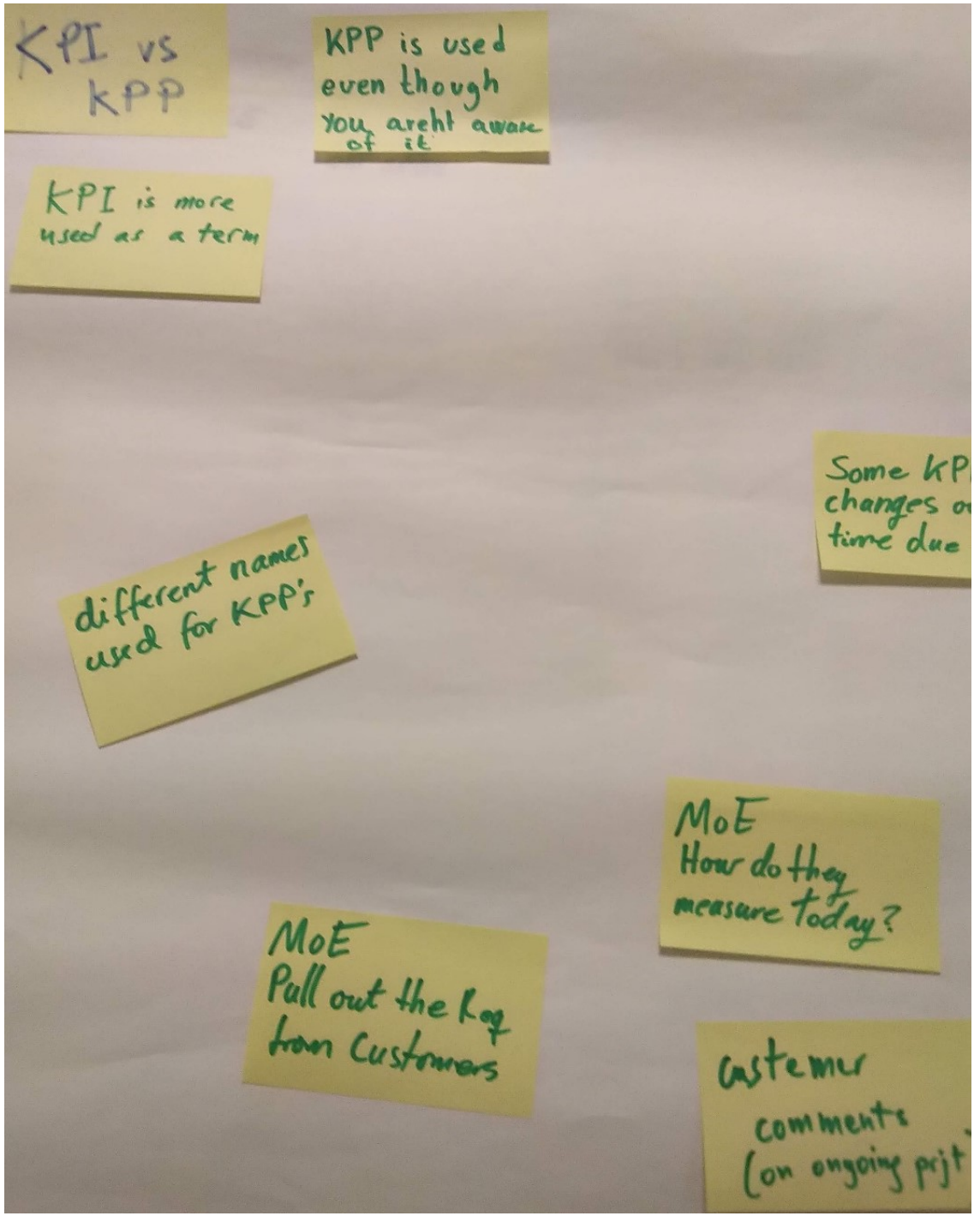


Breakout result (2)

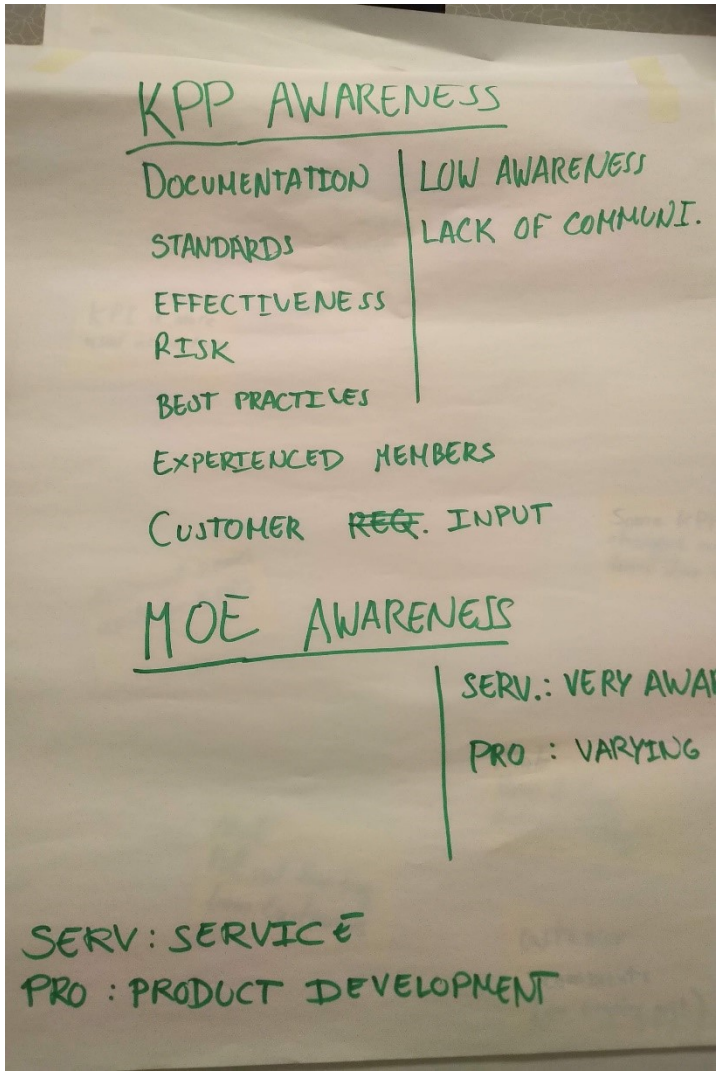


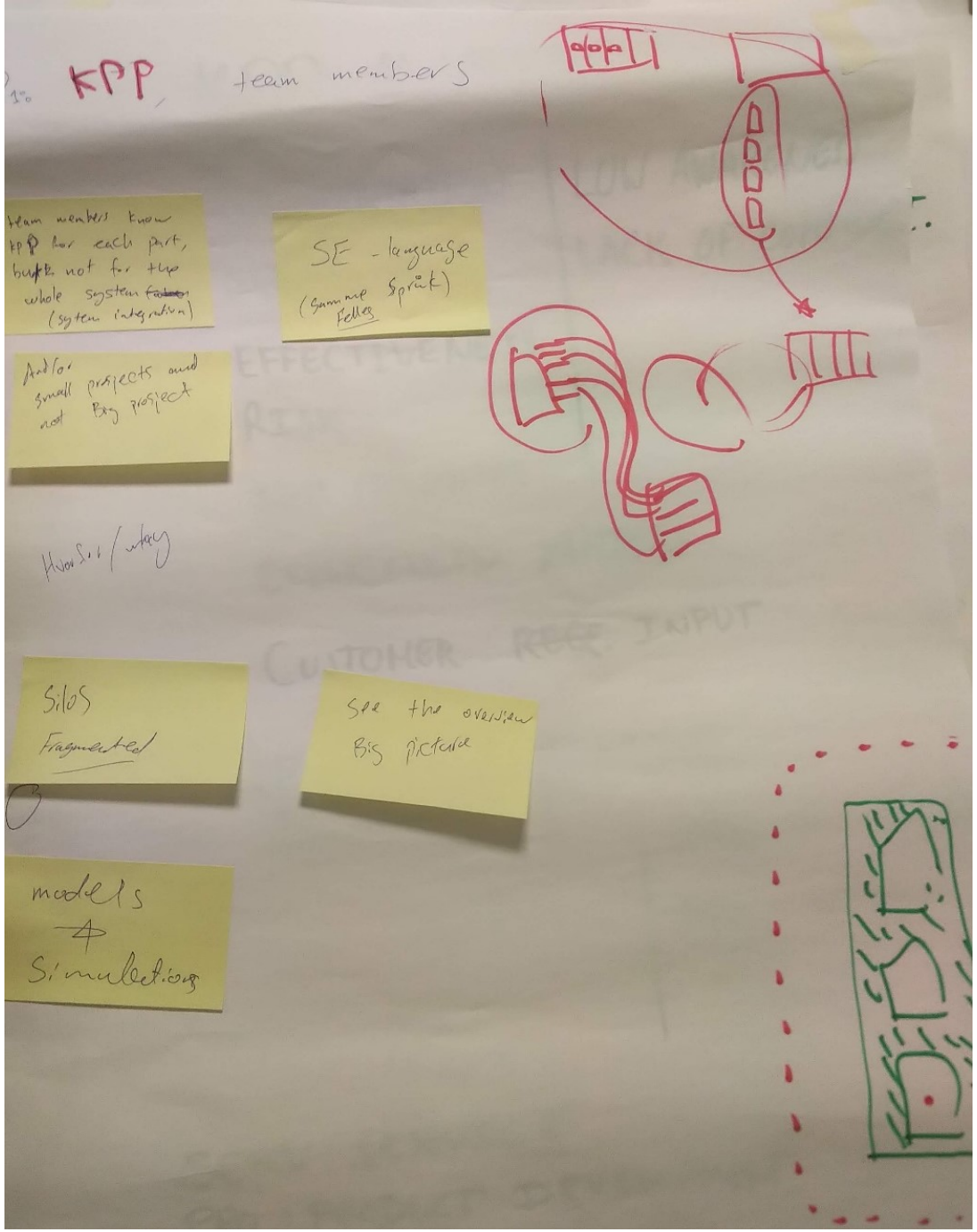
Breakout results (3)



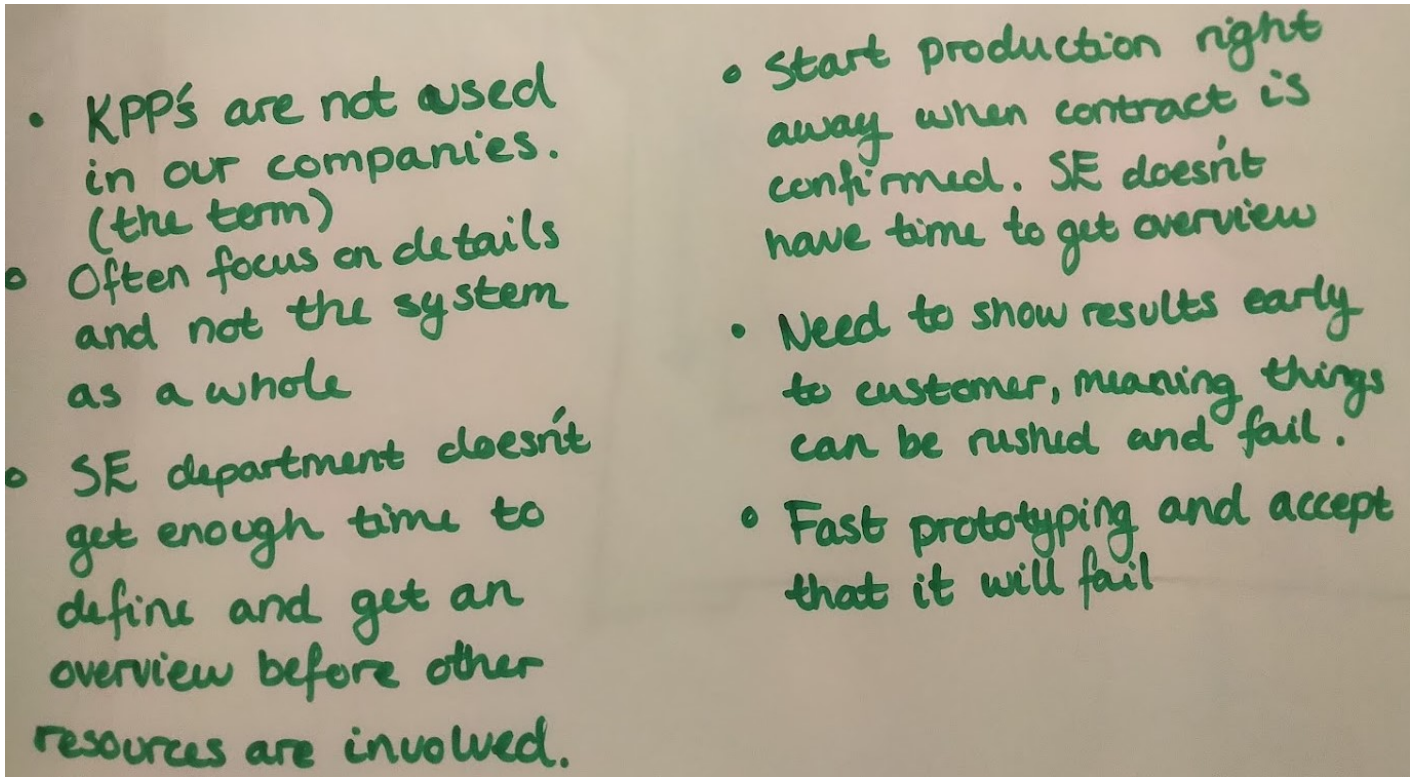


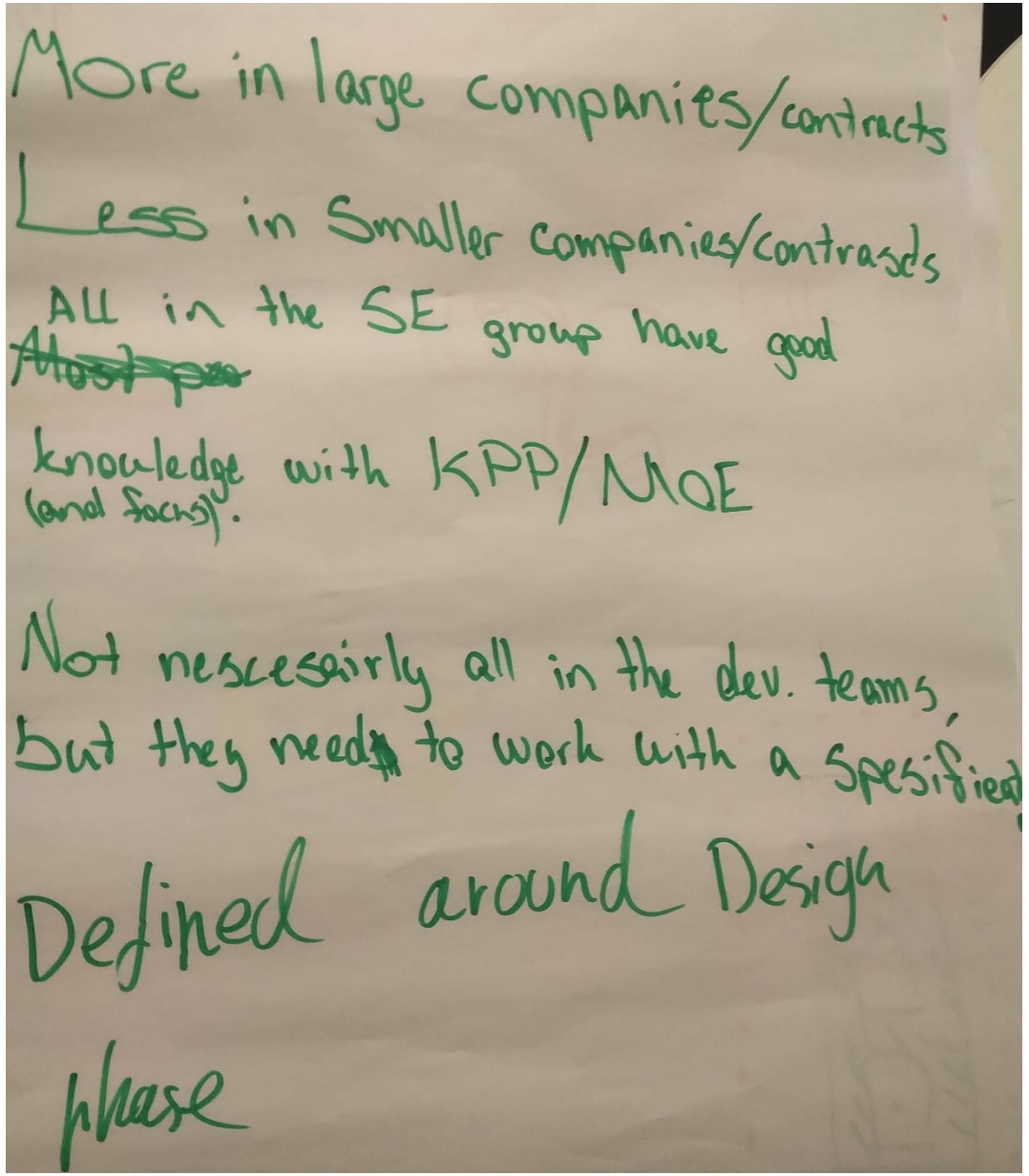
Breakout results (4)





Breakout results (5)





Conclusion(s)

Gerrit's conclusions:

The **awareness** project members have of **KPPs**, e.g. what performance is key and the actual numbers, **is shockingly low**.

This emphasizes the need for **early verification and validation** of the design and specification.

After SESG reflections and inputs

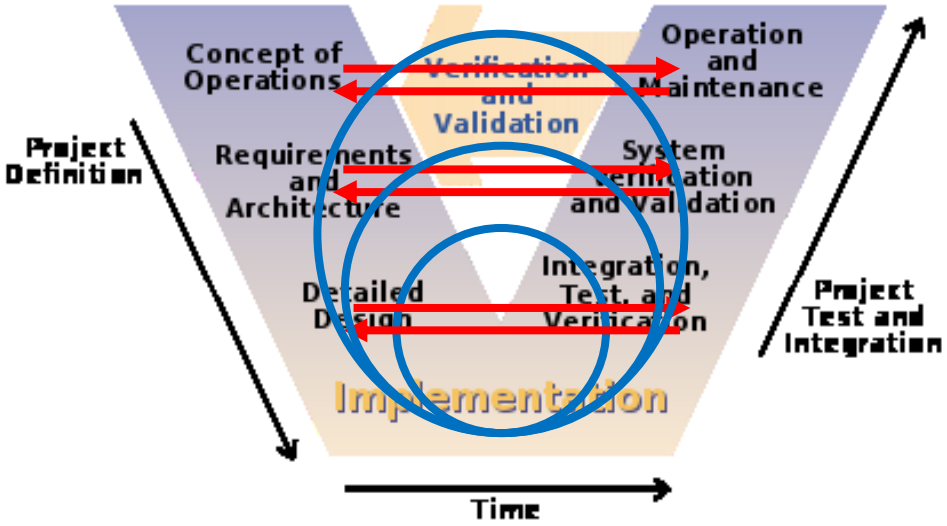
- Rune Andre Haugen is working on a PhD at KDA, on th topic how automation may help to detect undesired emergent behavior early during systems integration testing.
- He did send the following 2 slides as reflection on the SESG
- And after following the SERC December 1, by Sandra Hobson, he did send his take-aways that relate again closely to the systems integration challenges topic, see the third slide.

Reflections Rune Andre Haugen

- Key Drivers from customers can be monitored through MoEs
  - Stakeholder reqs and validation proceedures are updated as you learn iteratively
- KPPs from system reqs can be monitored through MoPs
  - System reqs and verification proceedures are updated as you learn iteratively
- Key Tech & Design from system design can be monitored through model & sim and/or system integration testing (SW and/or HW)
  - Design and integration/test proceedures are updated as you learn iteratively
  - SW and/or HW are updated as you learn iteratively

To V or not to V?

Not V, but Vo!



Reflections Rune continued

- Emergent Behaviors of the system can be monitored through MoPs and MoEs
  - MoPs can show performance levels dropping (below threshold)
  - MoEs can show effectiveness of system in operations dropping (below threshold)
- What prevents us from detecting emergent behaviors of the system?
  - Model correctness
  - Implementation (SW/HW) correctness
  - State space of testing regarding coverage of the real world system operations
  - Lack of knowledge, time, and equipment access

Rune's take-aways from the SERC talk on December 1, by Sandra Hobson "How is T&E Transforming to Adequately Assess DOD Systems in Complex Operating Environments?"

- Digital Engineering (MBSE, Digital Twin, Digital Thread, etc.) improves SE
- The design of experiments is important to achieve an effective and efficient test design
- We can only test a fraction of the operational space (infinite state space)
- Automation is an important resource in test and analysis
- Human-in-the-loop testing is equally important even with more automated testing (common misunderstanding)
- Data is the key
- Continuous testing is necessary due to the fast evolution of products
- Iterative development and testing is essential as (SW intensive) systems are continuously evolving