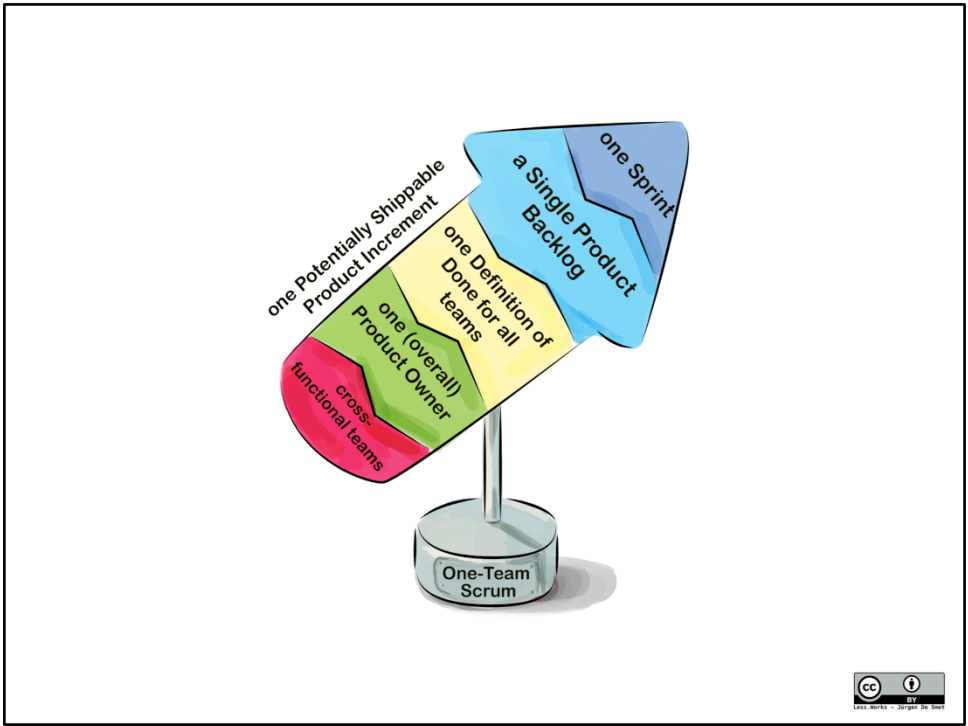
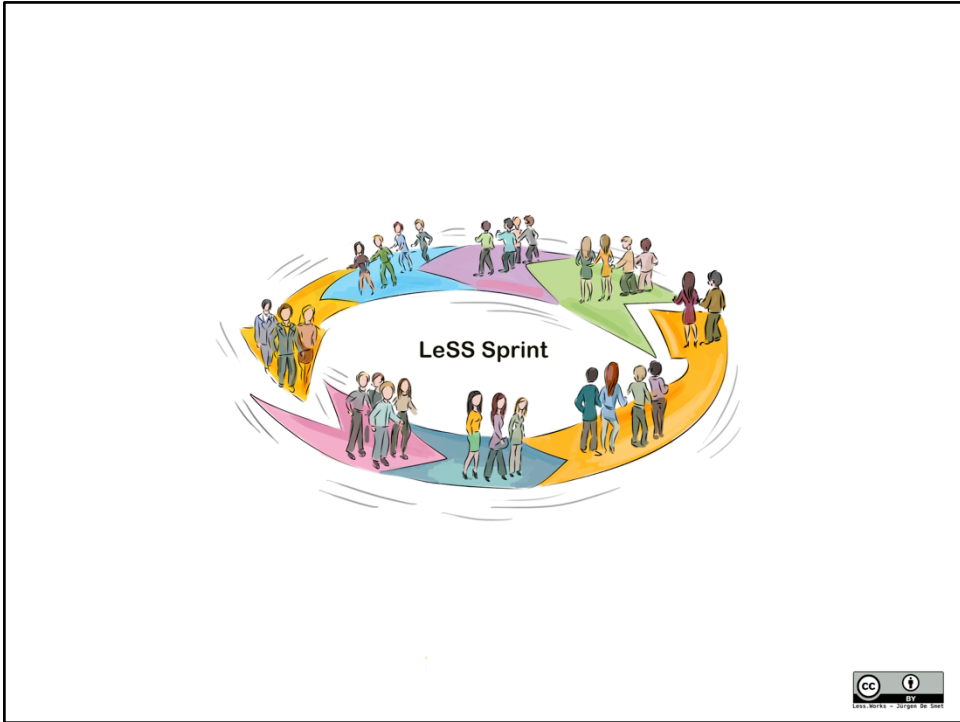


LeSS in a single image!

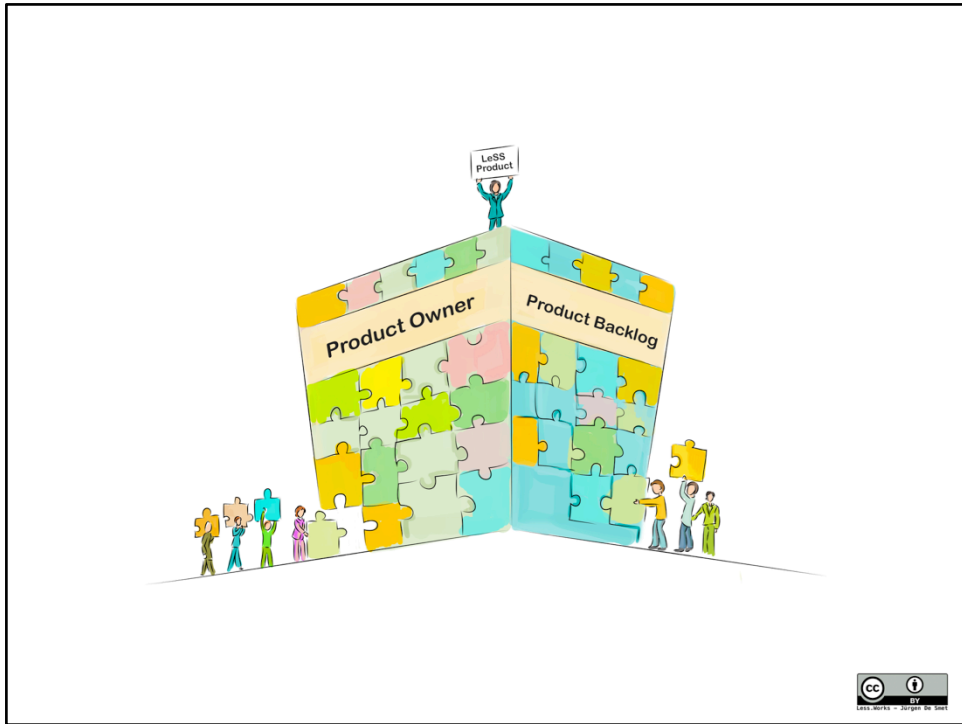
If you would cover the entire image with a black layer you would see a simple structure pointing to the blue sky, having a rock solid foundation. This is similar for LeSS, from the outside LeSS is simple and easy but when going through the inside one has to discover a lot of items to finally get to the diamond. Similar as diamonds, a 100% LeSS is not yet found and most probably you'll end up with a certain carat diamond. Never the LeSS, never give up and keep going for perfection!



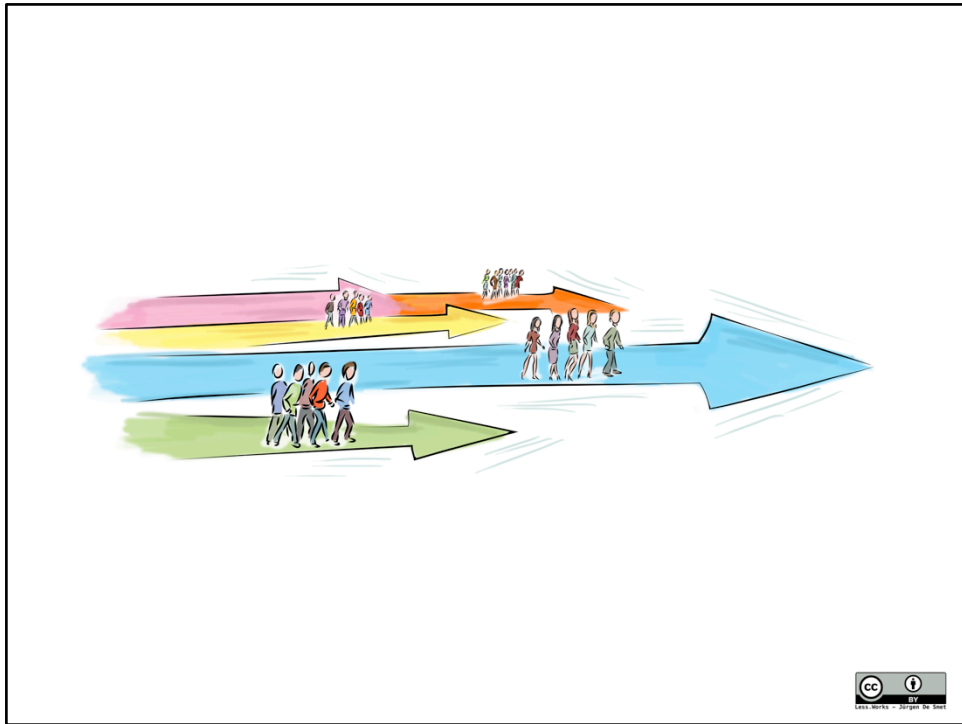
LeSS is a framework that is closed to “one team scrum” and while growing over multiple teams it keeps the basic principles of Scrum alive.



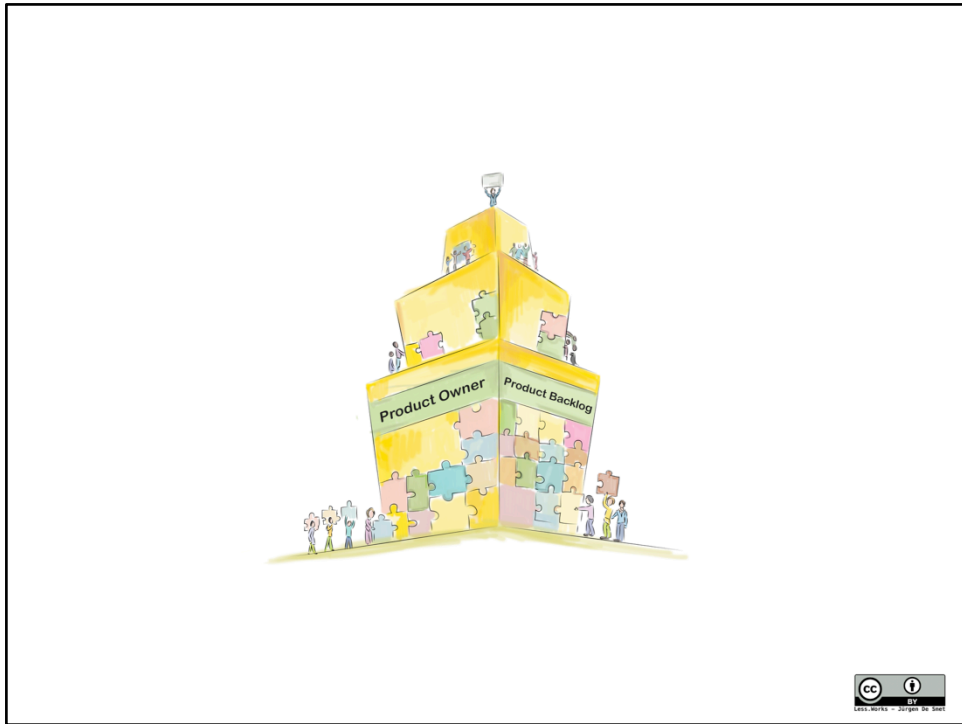
Up to 8 teams doing scrum. Explain what a sprint is and how it works with PO/Team/SM...



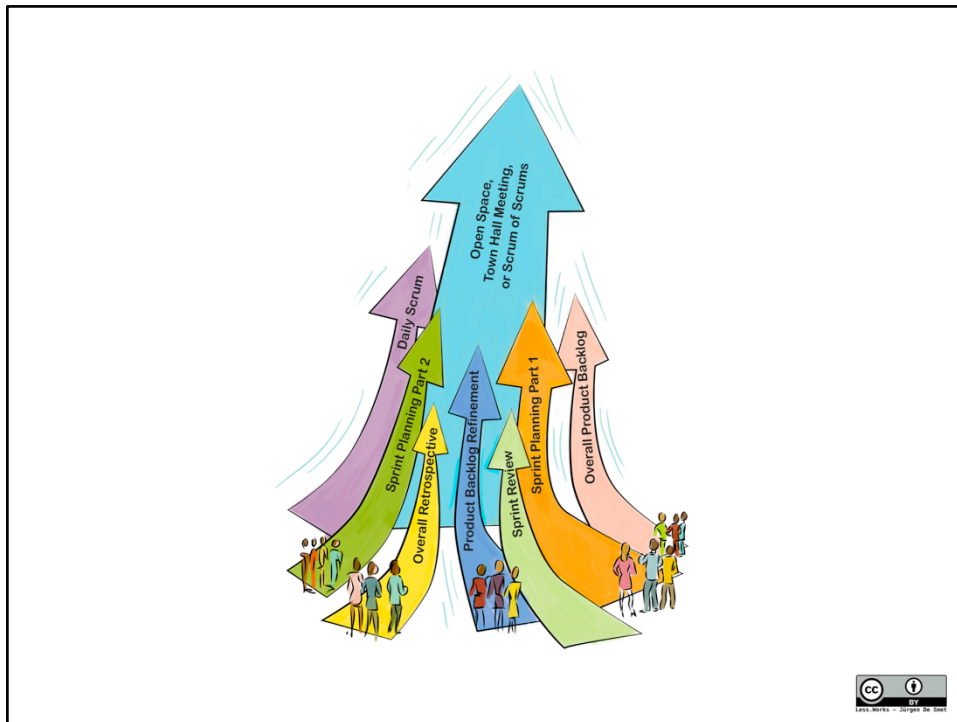
Delivering one product with a customer centric view. Explain a bit more around the PO activities and having a good PBL



Having several area groups doing sprints will get you to a light speed delivery system. Get into a interactive session with attendees on scaling the basic scrum using different goals maybe, address some of the issues and let them share thoughts, then go into “more with less”.



Most ideas on scaling will divert into component teams or development areas etc but let the group concentrate on customer centric focus and see if we can guide them to 1 PBL for a Huge Product. (Notice there is work ongoing on each floor block)



It splits different aspects as group activities or team activities. Having sprint planning part 1 as a group activity, part 2 as a team activity, commitment if following the second part and is done in group again. From that moment on teams are pretty much working independently having their own daily scrums and more. To align over different teams open spaces, town hall meetings and scrum of scrums are organized. This while we keep working on a single product backlog, refine the items together and deliver a increment as a group.

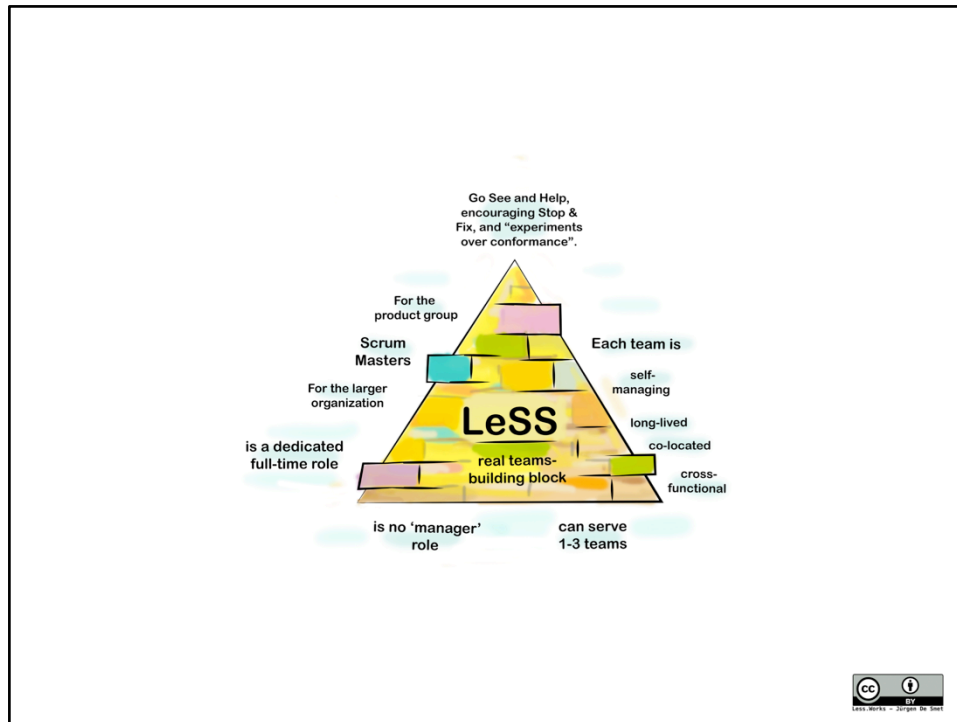
We get fireworks ongoing, fun, colorful, fast and very enjoyable.



LeSS is a framework that helps us create “winning” organizations that go from a single team of 8 persons (right side) to one with hundreds of people (cup).



The one and single way to get from A to B is to adhere to the basic LeSS principles, they are the tree of life within a good LeSS environment/system.



Have a short discussion with the group on each item and what it would mean within their organization.

Important to announce Larman's Law during this part

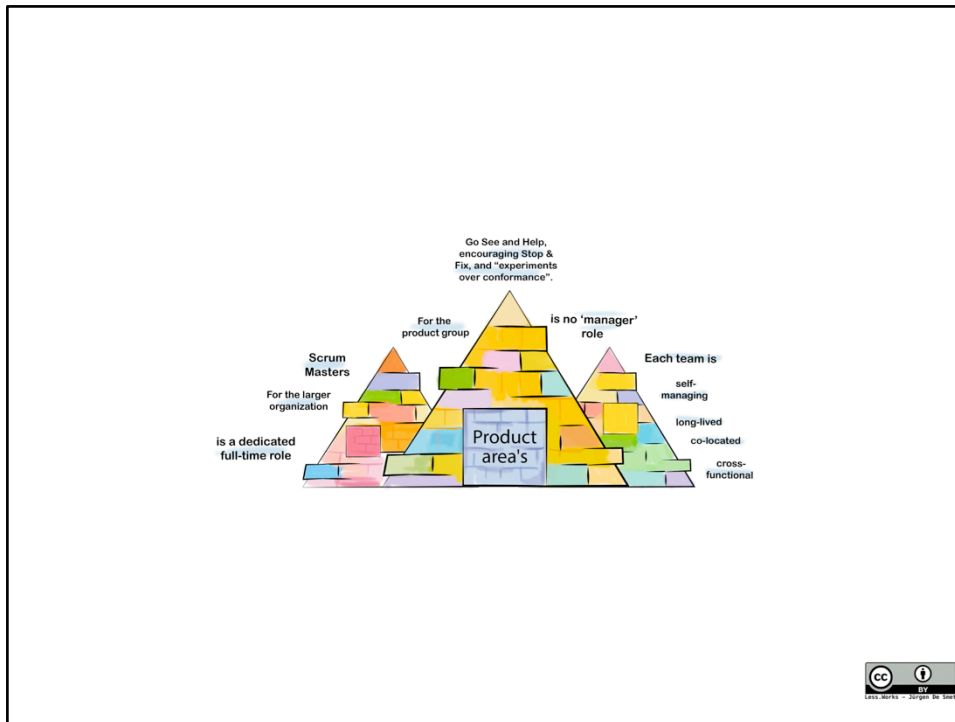
Larman's Laws of Organizational Behavior observes that...

Organizations are implicitly optimized to avoid changing the status quo middle- and first-level manager and single-specialist positions & power structures.

As a corollary to (1), any change initiative will be reduced to redefining or overloading the new terminology to mean basically the same as status quo.

As a corollary to (1), any change initiative will be derided as "purist", "theoretical", and "needing pragmatic customization for local concerns"—which deflects from addressing weaknesses and manager/specialist status quo.

Culture follows structure.



LeSS Huge (going beyond 8 teams) is basically the same but replicating the structure over different product area's that are servicing a 360° product having a customer centric view on the deliverables. Explain to the audience that having good defined area's will reduce dependencies and simplify their architectural structure along the way. (culture follows structure)

Important to make sure people understand difference, see below.

Requirement Area	Development Area
organized around customer-centric requirements architecture	organized around product's architecture
no subsystem code ownership	code ownership per subsystem
temporary in nature; should change over the lifetime of the product, but not at every iteration over the lifetime of the product	tends to be more fixed
focus on the customer, using customer language	focus on the architecture, using technology language



Of course this can only really be successful when having Software Craftsmanship and Technical Excellence continuously on the “radar”.