

Process of Change

Perspectives from OBM / Lean / Agile work

Presenters: Rolf Olofsson & Laurent Alt

Lean – an overview

TPS: Toyota Production System

- 1900's: Ford and Taylorism
- Optimized factories to produce one model (Ford T).
- Then Toyota started to make cars, for a smaller market (Japan). They visited Ford facilities.
- But the industry was changing: in particular, there was more variability in car models
- How to deal with this ?
 - Western car manufacturers: fork processes, manage stocks very carefully -> more administration, more costs
 - Toyota: manage variability at the workshop level, a new way of working
- Toyota outperformed all car manufacturers, birth of Lean manufacturing
- *For more about this, see "Profit beyond measure" (Johnson & Bröms, 2008)*

Lean: how does it work ?

- Measure everything
 - Lead time as the main KPI, visual management
 - Many metrics to obtain agreement between stakeholders
- Stable teams or equipments
 - Fixed production capacity
 - Ability to change priorities without changing organization
- Problem solving
 - Identify bottlenecks, waste and other problems through data
 - Local problems and global problems !
 - Go and see (gemba)
- Drumbeat
 - Takt-time
 - Needed to be able to synchronize all tasks
- Customer focus
 - Provides feedback loops to organizations



Lean: a new management system

- Lean is not Taylorism. Neither is it about reducing staff.
- Lean is intrinsically respectful of people. Learning is at the core of lean.
- In Lean, leaders are not in charge of assigning work to people: this naturally flows from customers (pull)
- Lean Leaders main responsibilities are:
 - quality
 - help teams solve problems and optimize flow
 - develop people
 - provide vision for the future (alignment)

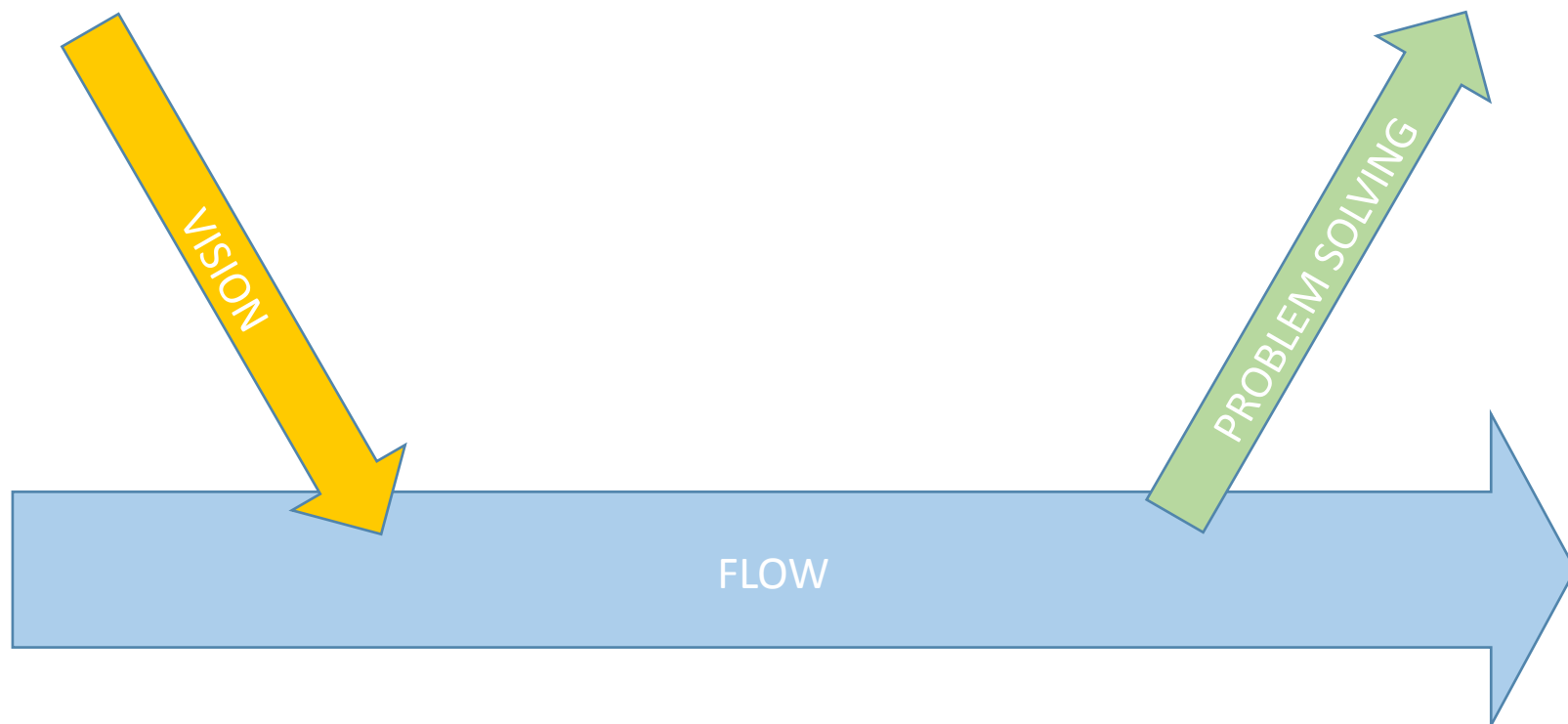
Top Management



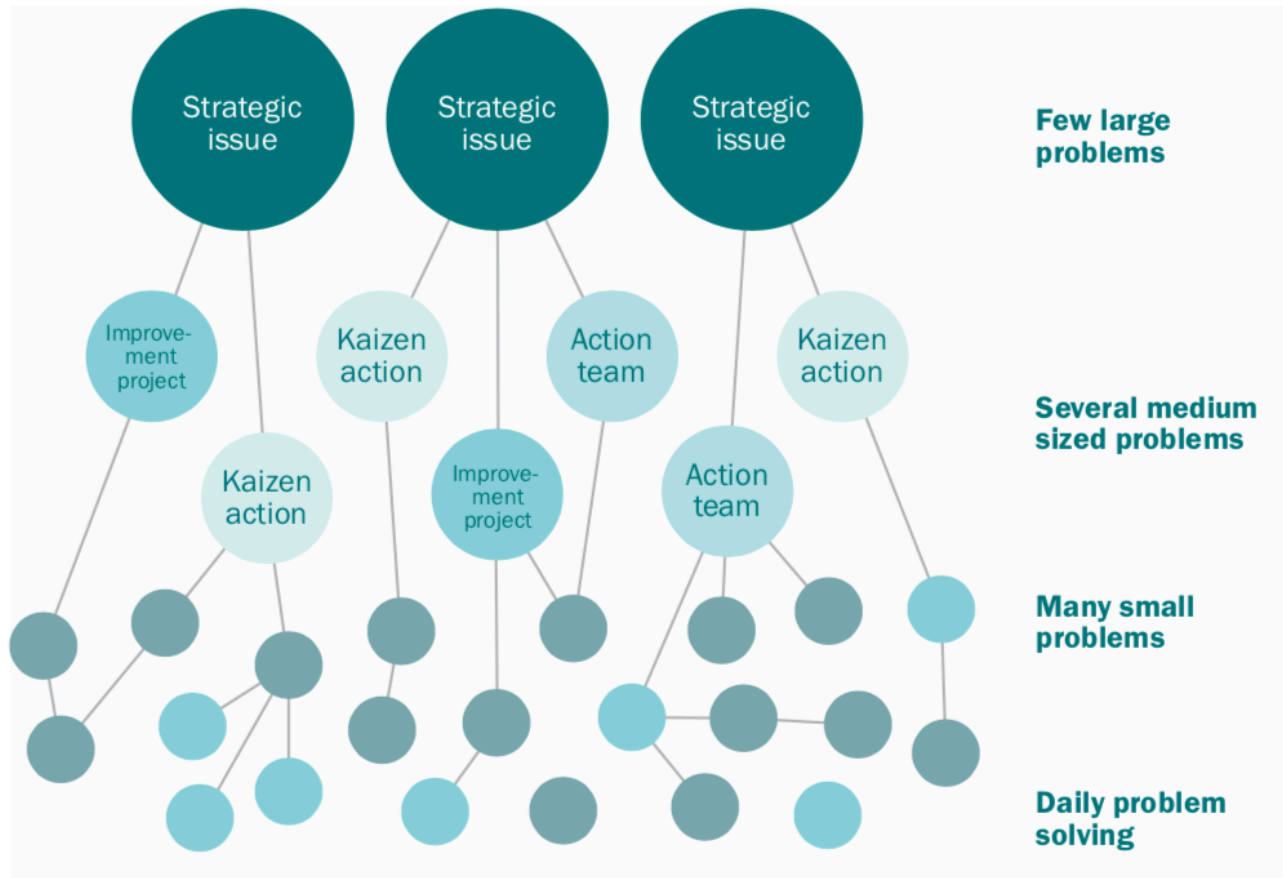
Mid-management



Operatives



Lean: problem solving and hierarchy

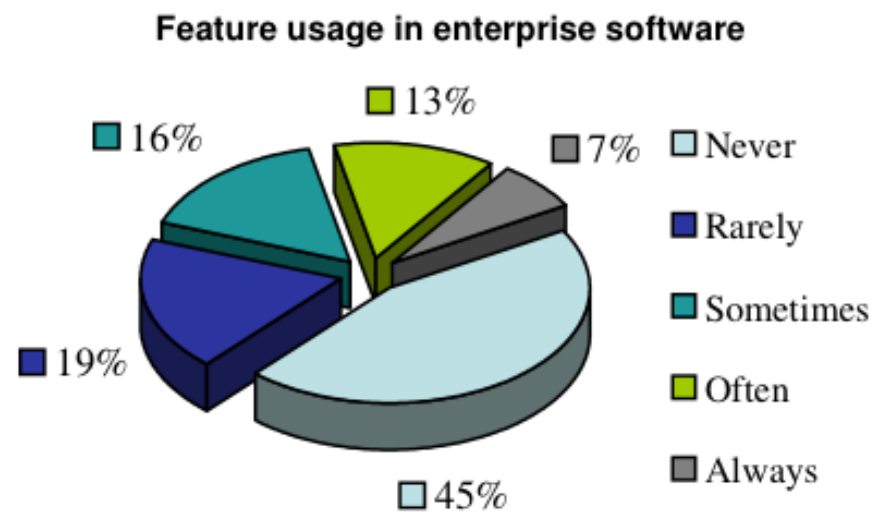
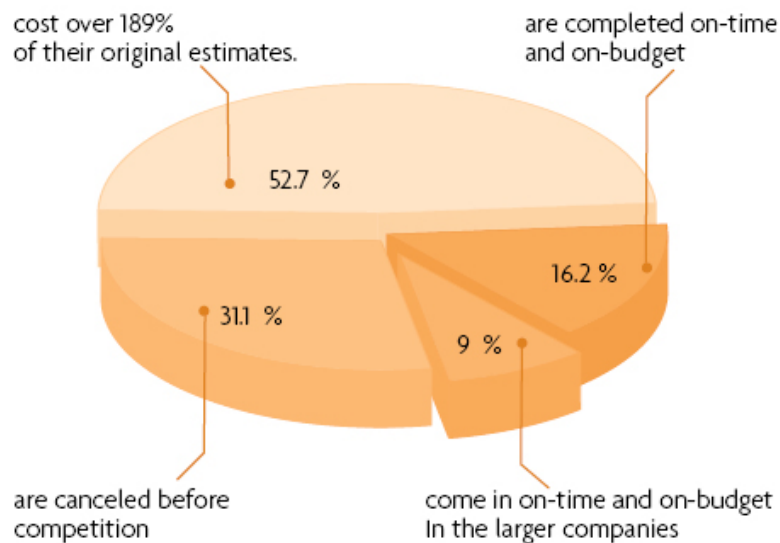


Source: Pascal, D. (2007). *Strategy deployment: What is it? Why should I care?*, Lean Enterprise Institute Webinar

Agile – an overview

Agile

- Agile was officially born in 2001, following many experiments to change the classical way to manage IT projects
- Analysis in 1995 by Standish Group – IT project failures



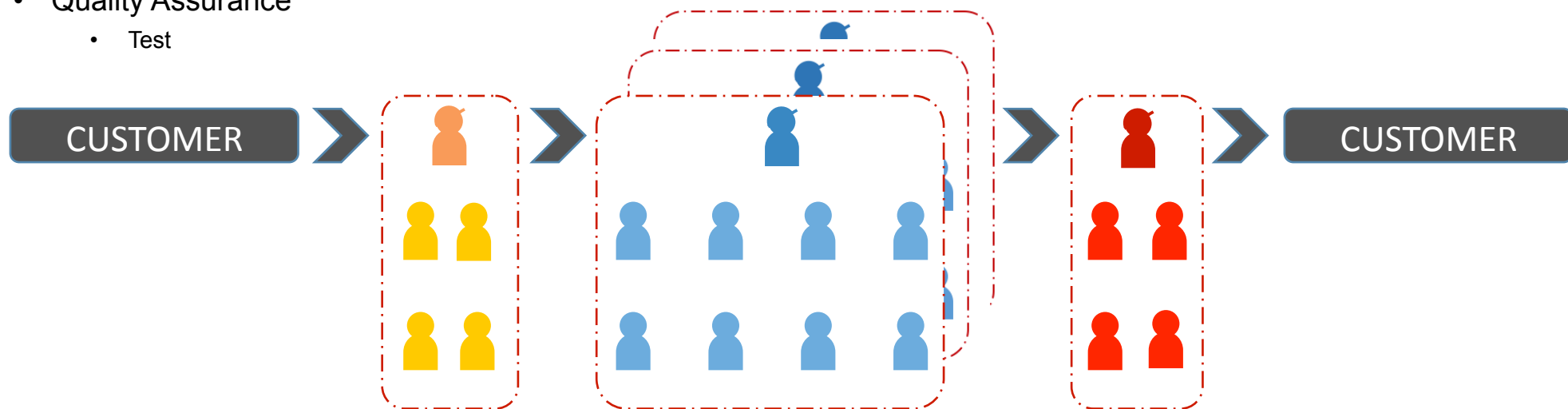
Agile manifesto (2001) – 4 agile values

- *We are uncovering better ways of developing software by doing it and helping others do it.*
- *Through this work we have come to value:*
 1. **Individuals and interactions** over processes and tools
 2. **Working software** over comprehensive documentation
 3. **Customer collaboration** over contract negotiation
 4. **Responding to change** over following a plan
- *That is, while there is value in the items on the right, we value the items on the left more.*

<http://agilemanifesto.org/> - see also the 12 agile principles

Software development – the old way

- Marketing / customer :
 - what is the product the customer is willing to pay for ?
 - what is the problem the customer wants to fix ?
- Functionality
 - List of functionalities the product has to have
 - Think of Powerpoint : File Open, Edit, Manage slides, create animations, ...etc,
- Development
 - Write the code
 - Organized by same-technology teams
- Quality Assurance
 - Test

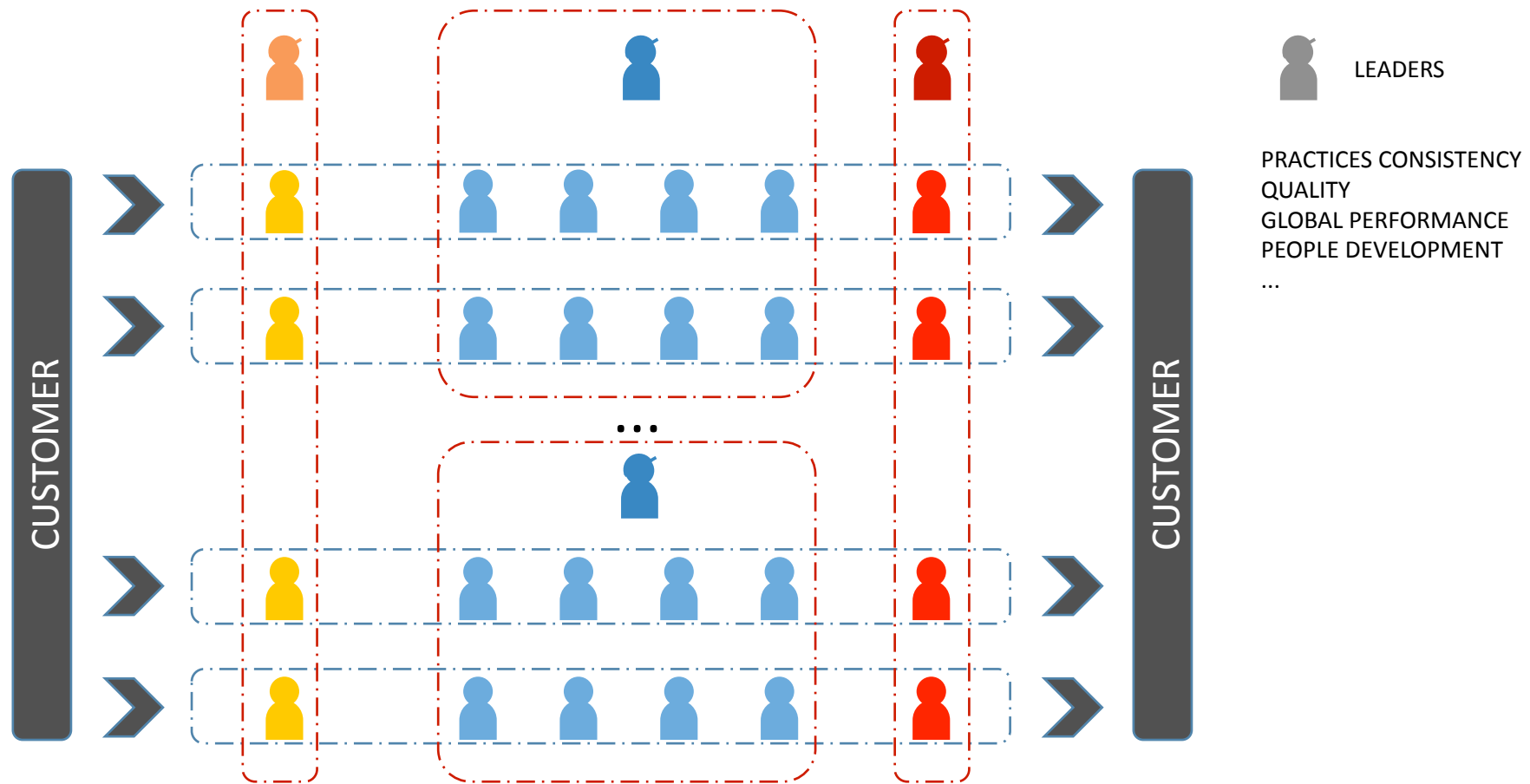


Software development – the agile way

- An agile team is a stable group of people who have autonomy to deliver a part of software
- It includes all key roles to build software
 - product owner : functionality / value (1 per team)
 - tech = software developers (3-5 per team)
 - quality = tester (1-2 per team)
- The team members are full time, other people can support the team (UX, cloud, ... etc) but are usually not part of the team
- Even at a large scale, agile will still involve autonomous 6-to-9 people teams, with various ways to organize and synchronize them

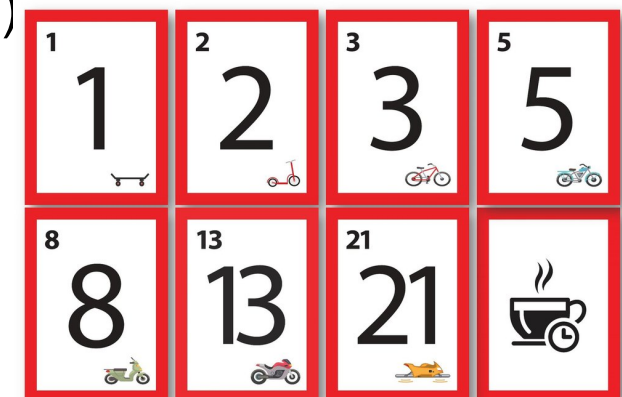
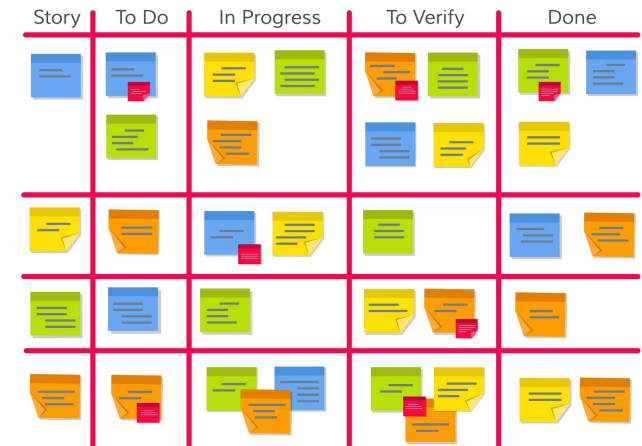


Software development – an agile organization



Agile – how teams work

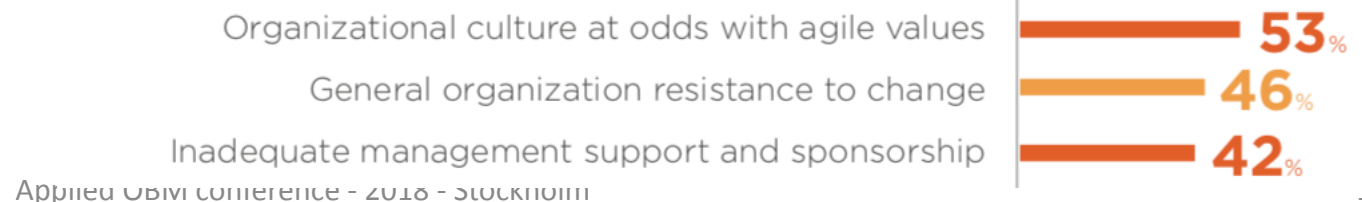
- Stable small teams of 6-9 people
- Breakdown work in small chunks (called *user stories*)
- Show their work every 2 weeks
- Well defined **ceremonies**:
 - planning, 15' stand-ups, demo, problem solving ...
- ... and **roles**
 - ... scrum master, product owner,
- Specific vocabulary (sprints, scrums, product owner, etc...)
- Gaming atmosphere (e.g. play cards to estimate effort)
- Team spirit, collaboration



Changing at scale

The challenge of changing at scale

- Lean and Agile now expand from manufacturing and IT to logistics, services, sales, marketing ...
- Lean
 - Since many years, large variations in Lean implementations are found across facilities of the same company
 - Frustration: famous paper by Jim Womack : <http://planet-lean.com/jim-womack-lean-thinking-promise/>
 - « *The results were strikingly positive, but the organizational immune reaction was immediate and crushing.* »
 - Also a reputation of cost-killing approach
- Agile
 - Small companies manage to become Agile
 - Startups can have an Agile culture from the start and scale then (like Google, Amazon, etc...)
 - But large companies have to deal with the existing culture
 - According to State of Agile 2018 report, the top 3 « *Challenges Experienced Adopting & Scaling Agile* » are:



What Lean and Agile have in common

- Lean and Agile put an emphasis on many aspects of work:
 - stable teams, cadence
 - prescribed roles and ceremonies
 - delegation / autonomy
 - focus on product and customer feedback
 - wide use of data and metrics
 - visual management – information transparency
 - experimentation, learning by doing, innovation
 - importance of collaboration and group learning
 - problem solving as the backbone of improvement
 - flexibility with respect to priority changes (Agile)

Changing at scale

- The previous characteristics describe how a lean / agile system works, but we may not want to tackle change the same way at different hierarchy levels. Not the same roles, not the same context.
- For simplicity, let's look at 3 levels:
 - teams (individuals in groups of 6-10 people)
 - groups of teams (100's of people, including leadership)
 - corporate (vision, corporate policies)

Top Management



Heard that Lean / Agile is needed

Management by goals, not by means
Not used to participate in production
Teams stability: need to change budget policies
Collaboration culture: need to change HR policies
Problems ? What problems ?

Mid-management



More collaboration between management
Want to be positive leaders

Fear of not making their figures (individual goals)
Risk of becoming useless
Used to coercion, and management by exception

Operatives



Roles and ceremonies are well defined
More autonomy
Receive frequent feedback

Do not use metrics (used against them ?)
Do not report problems (same)
Not used to have autonomy or innovation

Reality: at the team level

- Lean and Agile prescribe how people work, with roles, ceremonies, and we usually have little difficulty for adoption here.
- people get more autonomy, they receive feedback on a regular basis, they improve their way of working
- the key behaviors that are always difficult to setup in teams are:
 - the use of metrics to measure team performance
 - and transparency of information
 - ... these are key in order to have a good predictability, provide feedback and measure improvement, but people are afraid that data will be used against them by management.
 - sometimes innovation does not come spontaneously: people are used to being told what to do

Reality: at the group level

- in larger organizations, dealing with groups of teams becomes a challenge because giving more autonomy to teams requires another way to coordinate work across teams, and a new style of management.
- Usually, leaders (mid-management) are harder to convince, because they need to:
 - delegate more, but provide vision and guidelines upfront
 - not interfere with the teams while they work
 - give feedback at a regular pace based on objective judgment
 - contribute to continuous improvement by helping solve teams problems
- The main concern is that their responsibility is diluted in a system. They do not know how to combine respect of people, and performance. They have been trained to be coercive to get results (negative reinforcement).

Reality: for top management

- budget and contracts based on given targets, not on means:
 - money is managed by assigning targets to teams, not by providing them the means to work well and asking them to get the most out of it (*management by results vs management by means*), which is somewhat incompatible with giving autonomy to teams
 - projects are managed by cost, not by added value
 - this has an impact on how the company manages projects, and contracts
- HR policies:
 - performance evaluation based on individual performance, not on collaborative behaviors
 - jobs defined too precisely that make it difficult to smoothly evolve from one role to another (prevent collaboration too)
- same impediments to change as for mid-management (previous slide),
- plus, a strong focus on KPIs (but less on the product itself), and often a lack of corporate vision (which prevents good delegation of work)