Meet-Up

Tracing Local Optimizations throughout the System

November 1, 2018
Thanks to Rabobank Agile Community
Coming Up...

11/17 – Certified LeSS Basics (CLB) | Basking Ridge | NJ

12/10-12 – Certified LeSS Practitioner (CLP) | New York | NYC
What Is Local Optimization?

"Everyone is busy and working so hard. Yet, the system is delivering slow and Users are not happy"

How could that be?

Also at: https://less.works/less/principles/systems-thinking.html#Seeing(andHearing)LocalOptimization
Component Team

With component teams, a project or feature manager is used to coordinate and see to completion a feature that spans component teams and functional teams.

With component teams, there is a tendency to select goals familiar or ‘fast’ for teams, not for maximizing customer value. For example, Component B Team does part of Item 3 because it mostly involves Component B work. This is the “watching the runner rather than following the baton” local optimization.

With component teams, there is increased delay, as one customer feature is split across multiple component teams for programming, and eventually transferred to a separate testing team for verification. There is handoff waste, and multitasking waste—as one component team may work on several features in parallel, in addition to handling defects related to ‘their’ component.
With component teams, there is a tendency to select goals familiar for people, not for maximizing customer value. For example, Component A Team does Backlog Item 3 because it mostly involves Component A work.
With feature teams, teams can always work on the highest-value features, there is less delay for delivering value, and coordination issues shift toward the shared code rather than coordination through upfront planning, delayed work, and handoff. In the 1960s and 70s this code coordination was awkward due to weak tools and practices. Modern open-source tools and practices such as TDD and continuous integration make this coordination relatively simple.
Component vs. Feature Team
Mini-Waterfall

Incorrect:
this is not Scrum nor a Scrum Team, it is a mini-waterfall of single-function groups with handoff and WIP queues.

Correct:
cross-functional team does all work (architecture, analysis, interaction design, internal design, programming, test, ...) without handoff.
Mini-Waterfall – Cont.

Component teams create sequential life cycle development with handoff, WIP queues, and single-specialist groups. This organizational design is not Scrum or agile development, which are instead based on true cross-functional teams that do all work for a feature without handoff. This “mini-waterfall” development is sometimes confused as agile development; that is a misunderstanding.
Seeing (Hearing) Local Optimization in...

- Team Structures
- Org. Structures
- Documentation
- Definition of Done
- Backlogs
- Role Definitions
- Product Design
- Goals & Metrics

Also at: https://less.works/less/principles/systems-thinking.html#Seeing(andHearing)LocalOptimization
Q & A
Coming Up…

11/17 – Certified LeSS Basics (CLB) | Basking Ridge | NJ

12/10-12 – Certified LeSS Practitioner (CLP) | New York | NYC
Scaling Organizational Adaptiveness (a.k.a. “Agility”) with Large Scale Scrum (LeSS)

Organizational de-scaling (flattening) takes months and years to complete and from a high perspective, looks like a gradual process. However, throughout this long process, there are many phases (bursts) of comprehensive preparation, followed by an organizational “flipping”.

LeSS Huge
- As in LeSS, a Product definition becomes too wide to be supported by a single Product Owner. Area Product Owners (+ staff) are identified, to support independent Product Areas. Coordination between Area Product Owners and (Overall) Product Owner ensures good product strategy and long-term planning is balanced across Areas. Changes to organizational policies (e.g., location strategies, compensation) are made.

LeSS
- Simplified organizational design. System Optimization. Reduction of: silos, handovers, translation layers, bureaucracy and “muda”. Scrum is implemented by coordinated, feature-centric teams (2-8), building the same, widely defined Product/serving the same PO. Cases of Local Optimization by single specialty roles are eradicated. Teams are colocated. Multi-site development is used for multiple locations. Strong reliance of technical Mentoring and Communities of Practice (as oppose to first-line management). No subsystem code ownership. Gradual reduction of “undo” work and “undone” department. Heavy focus on Customer values. Strong support of Senior Leadership. Intimate involvement of HR.

Scrum
- Copy-paste scaling (no conscious scaling strategy) of Scrum throughout an organization: many teams doing their “own” Scrum. True product definition is weak. Cases of using Scrum in component-centric development are frequent (often, a result of trying to meet goals of agile transformation (6% annually), set at enterprise level. Importance of Scrum dynamics and roles is viewed as secondary, to existing organizational structures and blueprints. Too many single-specialty experts and very few T-shaped workers. No meaningful HR changes.

Waterfall