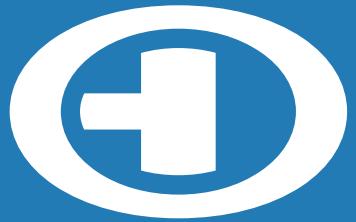


Structuring the Product Backlog

Presented by **Johan Karlsson**
Coach, Hansoft

2016-05-03

This Presentation is Based on Input from

Organizations on 3 Continents

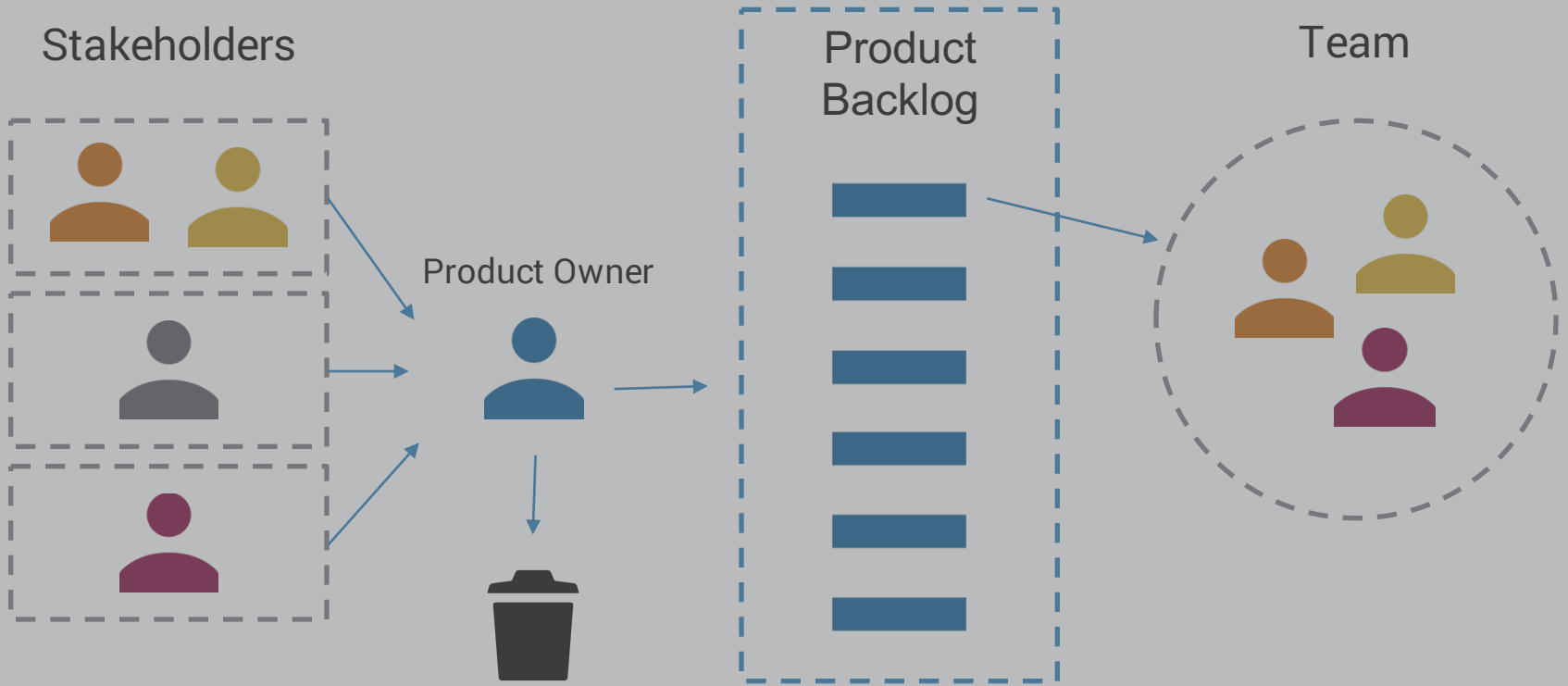
100's of Backlogs

Product Development Organizations

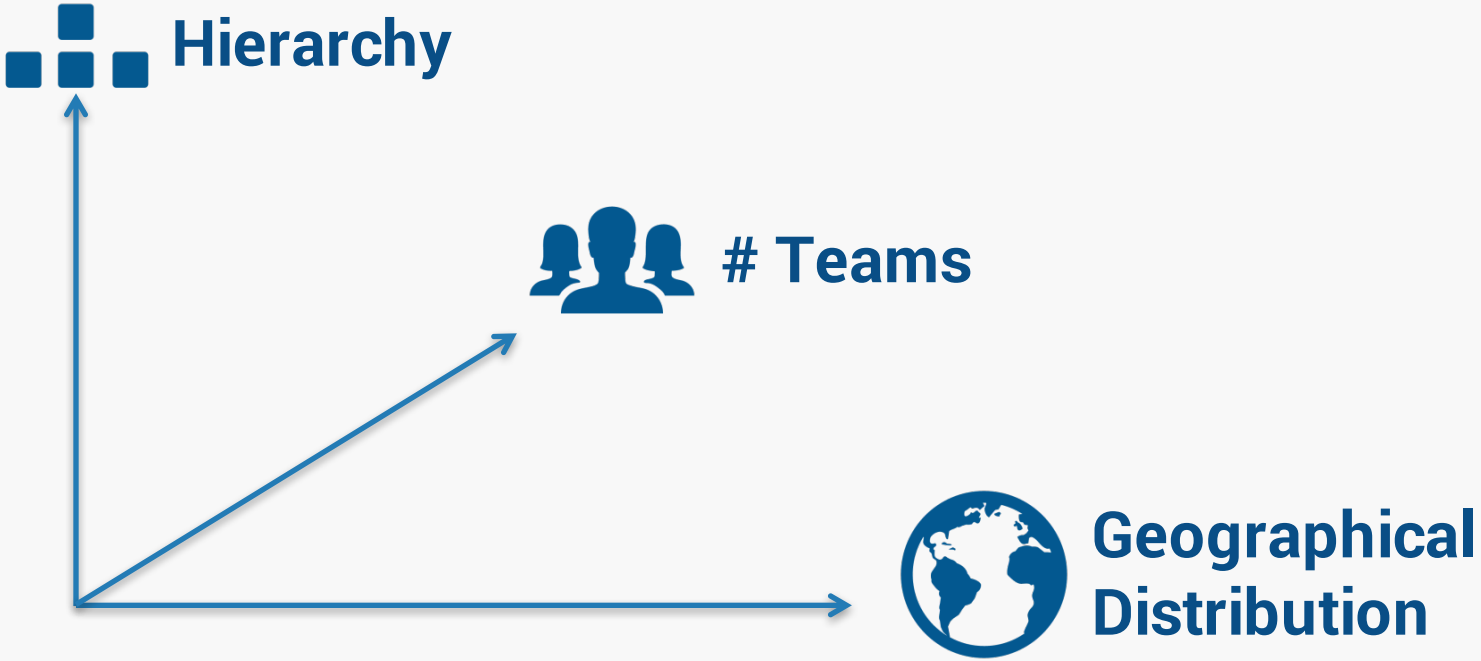
Own Experience

Why care about the **structure** of your Product Backlog?

The Backlog is not always a **Simple List**



Structured Backlogs Enable **Scaling**



A Structured Product Backlog in Transformation

Introduced

- Make Backlog Humanly Readable
- Leave Email Backlogs

Common Backlogs

- One Product Backlog
- Transparency
- Backlog of Backlogs

Forward-looking

- Goals Instead of Issues
- Acceptance Criteria and Definition of Done
- Product Ownership

Flow

- WIP Limits
- Cumulative Flow Diagrams
- Mindset Rather than Practice Discussions

Backlog structures give **competitiveness** through:

EFFICIENCY

Alignment

Focus vs. Choice

VISIBILITY

Transparency

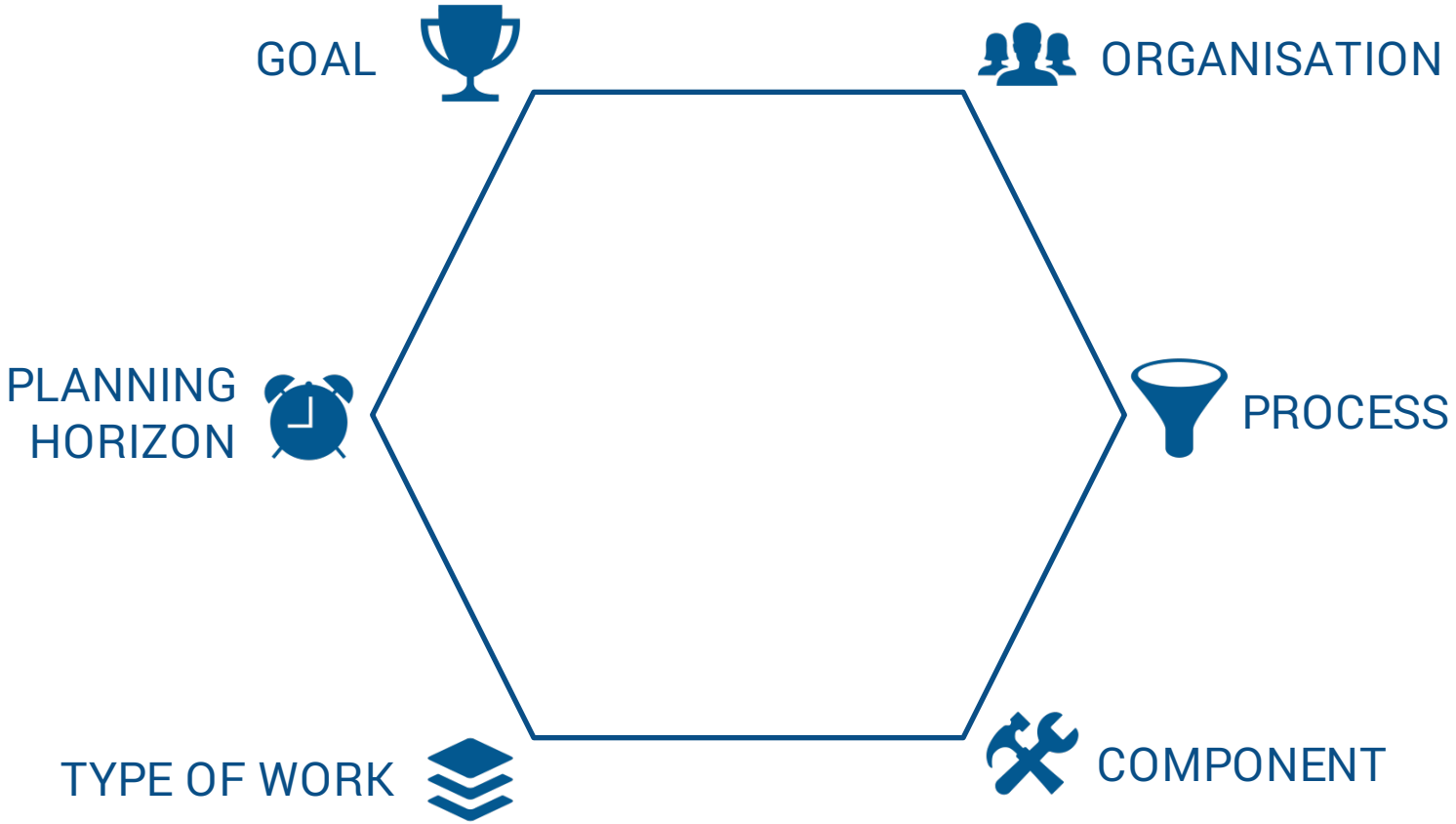
Empowerment

AGILITY

Flexible Requirements

Continuous Improvements

Dimensions of a Product Backlog



Which user story is most valuable?

- As a team member I want to see the next user stories to work on and understand the overall matrix of where it fits in.
- As a Product Owner I want to see how the features are progressing and change priorities and refine them together with the team.
- As a project manager I want to see how the product develops and how overall progress is going.
- As a stakeholder I want to understand when my thing is ready.

Pros/Cons

Organization / Team Backlogs

Good Patterns	Bad Patterns
At the core there is the team, so a team backlog reduces task switching	Difficult with many dependencies and where feature teams are not possible
There is likely a resemblance between organization and product (see Conway's Law)	A backlog item may loose other relevant context for alignment

Component based

Good Patterns

When many components are delivered by third party to manage dependencies and timeline

Works good together with Gantt schedules to show dependencies and what is required for a milestone

Bad Patterns

Stuck in traditional work breakdown structures (WBS) with doing big plans up front

Difficulties with including business value and to set priorities for a Product Owner

Goal Backlog

Good Patterns

Focus on valuable item that drives business strategy and vision

Helps in shaping feature teams

Product Backlog priority is business value

Bad Patterns

Easy to fall into waterfall style breakdown too early (do you know the path to the goal?)

Where does all the work that does not add direct value / fit in go?

Planning Horizon

Good Patterns

Focus on shipping ready product on cadence

In complex configuration management contexts (to plan what/when)

Bad Patterns

Committing too early to a date

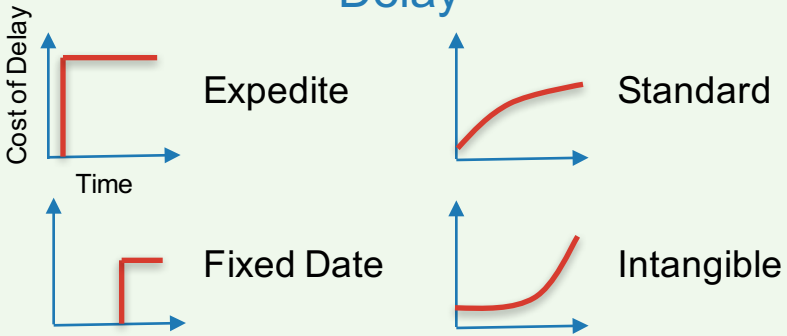
Process-driven backlog

Good Patterns	Bad Patterns
Use the process to limit WIP	Stops questioning the process *or* falls into obsessive workflow tuning
Focus on continuous improvements to remove bottlenecks and shorten lead times	Neglect individuals and interactions
	Seldom the most important thing (regulated industries aside)

Type of Work Backlogs

Good Patterns

When useful for prioritizing, such as Classes of Service based on Cost of Delay



Bad Patterns

Stuck in wasteful discussions such as ”Is it a bug or a feature?” or “Is it severity A or B?”

Bureaucracy: Too many ”types” for reporting or nice-to-have purposes

Questions to ask when changing structure

- **What is the current size of the backlog? Where will it grow?**
- **What are the top challenges right now for the product?**
- **Will I have a better understanding if we are on the right track?**

Item name	Class of Service	Release tag	Committed to sprint	Code Area
Combined Example				
Inbox				
SSO Solution	Feature			Backend
CPU Issue on certain machi..	Bug			Backend
Product Backlog				
Team A				
Send e-mail	Feature	Release 8 2016-03-01	A1 2016-03-01	Frontend UI
Crash at startup	Bug	Release 8 2016-03-01	A1 2016-03-01	Backend
Improve uptime to 99 %	Action	Release 8 2016-03-01		Backend
Team B				
Delete e-mail	Feature			Backend Frontend

Example of How to Introduce the Concept

Introduced

- Todo / Doing / Done
- User stories
- Current+Next sprint

Common Backlogs

- Team
- Product

Forward-looking

- Feature Teams
- Vision based
- Next 4 sprints
- Dependencies

Flow

- Workflows
- Flow
- Classes of Service

THANK YOU!

 @jhnkarlssn

 johan.karlsson@hansoft.com

 Uppsala