Overview

Setting the scene
The challenge

The journey part 1: finding a new way
Planning in uncertain conditions, operational flow, organization, kicking it off.

Intermission:
why were we successful? – some wisdom

The journey part 2:
Experimenting our way forward

Summary
Q&A
Setting the scene

(2008)

The global perspective
In the center

Observation

2000 people
7 sites globally
large, central functions (PMO, Technology, Test)
Dev = execution
Rigid risk and change management
After trying to drive improvements for 5 years

We are always delayed.

There is always drama around a project delivery.

Changes are very, very painful

Let’s do this together

7 cross-organizational workstreams to create change ideas

Invite Product Management, Finance and HR to the work streams

140 people from product development joining the work streams

>600 slides with new, improved process descriptions created in 5 months

Monthly face-to-face leadership team meetings to integrate our findings
The challenge (related to PMO)

How many % of your plans did not need to be changed later?
Getting going

Workstream #5:

Project planning and Scope management

Mission

— Build a cross-organizational team.
— Find out and make suggestions on what we can do against our projects delays.

INSIGHTS

› Roadmap adherence
› Inaccuracy in decision making:
   - frequent scope changes
     → it is impossible to predict the market 2 years in advance
   - slow response time for new requests coming from the markets
     → full development capability working on making the scope settled 2 years in advance → charge means massive re-planning
   - features that are implemented, but never sold
     → again: prediction problem
   - features that contain too much functionality
     → as we don’t know at an early stage what the feature should be capable of, we implement many options just to be on the safe side.

› Ability to learn and improve
  → huge projects, hierarchically organized → learnings only very slowly come up
  → learnings only after each project = every 2 years → very long learning cycles.
  → Waterfall: first specify, then design, then test → long time until test provides feedback to specification or design.
It is difficult to make predictions, especially about the future

Many clever people

Traditional way to deal with uncertainty
Stop ignoring uncertainty!

How to do this practically?
Managing uncertainty

This is the high-level backlog.

We do not pre-decide which feature will be in which release: 
**decide as late as possible**

→ feature-release states:
  — candidate
  — intended
  — confirmed

<table>
<thead>
<tr>
<th>Feature</th>
<th>Priority</th>
<th>Release Intent Market Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature 1</td>
<td>1</td>
<td>Release B</td>
</tr>
<tr>
<td>Feature 2</td>
<td>2</td>
<td>Release A</td>
</tr>
<tr>
<td>Feature 3</td>
<td>3</td>
<td>Release B</td>
</tr>
<tr>
<td>Feature 4</td>
<td>4</td>
<td>Release B</td>
</tr>
<tr>
<td>Feature 5</td>
<td>5</td>
<td>Release A</td>
</tr>
<tr>
<td>Feature 6</td>
<td>6</td>
<td>Release B</td>
</tr>
<tr>
<td>Feature 7</td>
<td>7</td>
<td>Release A</td>
</tr>
<tr>
<td>Feature 8</td>
<td>8</td>
<td>Release C</td>
</tr>
</tbody>
</table>
Visualizing time uncertainty on feature level

Feature burn down chart

Visualizing time uncertainties for features and releases

Release A

Release B

TODAY

Delivery range

Remaining work

time

Feature 1
Feature 2
Feature 3
Feature 4
Feature 5
Feature 6
Feature 7

Priority

cutoff

TODAY

Release Intent

B

A

B

B

A

B

A
Visualization
Uncertainties on portfolio/Scenario level

This needs to be used as a pull system
Lean principle: avoid overburden

Items, that are either mandatory or have passed GO

Capability limit

Release A

Release B

→ time

→ # people

Scenario data from Quick Studies
Ok, Great!

Now we have an idea how to handle planning and risk management in an uncertain environment.

Now: How does the operational flow need to look like?
The 3 main programs

Early phases program
Mission: deliver planning data to be able to plan scenarios and support product and sales decisions.

Feature implementation program
Mission: deliver features, that are integrated into the product.

Release Program
Mission: deliver product releases, that can be rolled-out globally.

The 3 main programs

Study and HL design

Design, implement & test

Integrate continuously

Verify in context of a release & ship out
Sketching an organization, that enables the flow

Not a hierarchy, but distributed planning, learning and decision making.
Central Portfolio & Technology Management – general ideas

Principles:
- The setup should mirror the XFTs and site setup.
- As small as possible to avoid central over-powering.
- Empower the organization.

Cross-functional team:
- Project Management competence
- Technical competence
- Test competence
- Delivery competence

Plus:
- Financial
- Supplier management
- Sourcing

Mindset: facilitation / servant leadership:
- Teams own the product creation and delivery.
- We co-own planning and making promises
- We don’t solve team’s problems unless they ask us for help.
- We focus on
  - giving direction and intent.
  - helping teams and the organization grow
  - keeping the overall picture together

Creating the new setup
High-level development center blueprint:
Teams in the center, supporting roles around

Structured as a hierarchy
(to give orientation and aggregation)

but

with a participatory leadership approach.

Agile coach is part of the leadership team.
At scale...
... self-similarity!

Kicking off the PMO
Kicking off the PMO – a lot of questions!

- How do we take decisions?
- Who can decide what?
- How do we stay in sync?
- How many BLs and how do we manage them?
- How do we manage dependencies?
- Meeting structures?

Co-ownership

Backlogs

For a 2000 people org this is difficult to manage.

- Backlog refinement hierarchy:
  - Feature BL (owned by PMO)
  - Epic BL (within feature) (Owned by the XFT)
  - USs within epics (Owned by the XFT)
  - Reserve money, people and a BL subset for things like system care (Technical Debt, Defects, ...)

There can be only one
Scrum of Scrum – for 2000 people?

Maybe it’s obvious, but ...:
— Why SoS?
  — To manage dependencies between teams.
  — To coordinate across a larger organization.

— We need to differentiate:
  — Tightly dependent work.
    → SoS could work!
  — Loosely dependent work.
    → SoS is too heavy.
  — Independent work.
    → SoS doesn’t make sense.

Use SoS where it makes sense:
where tight coordination is needed.

Scrum of Scrum – for 2000 people?

Maybe it’s obvious, but ...:
— Why SoS?
  — To manage dependencies between teams.
  — To coordinate across a larger organization.

— We need to differentiate:
  — Tightly dependent work.
    → SoS could work!
  — Loosely dependent work.
    → SoS is too heavy.
  — Independent work.
    → SoS doesn’t make sense.

— Our approach:
  — Use SoS where it makes sense.
  — Tightly dependent work:
    Example: a feature team consisting of several Scrum Teams. All need to be aligned towards one customer need / business outcome.
    → use SoS, CPO, PPO
  — Loosely dependent work.
    Example: Two features, that hit the same area of the code.
    → Central high-level coordination: feature anatomy plus empowered teams: “Team A and Team B: you need to talk to each other!”
  — Independent work.
    Only central BL management, nothing else.
Decision circles (operations and product)

Product: Strategies, scenarios, expectations
Whole organization

Feature

PM

PDM

Site

Team

Cross-Team

Operational
Product

Results

100%
Product releases on time since the change

Less defects found at customer after product release

Better and faster decisions through significantly improved interactions (collaboration culture)
This was successful! But why?

→ some wisdom we found along the journey

Organizations are human systems (= complex, adaptive)
We need to approach them accordingly!
Approaching complexity
(From the Cynefin framework)

Key characteristics:

Very limited predictability:
— Cause-effect can only be understood in hindsight.

Anything you try will have side-effects:
— positive and negative,
— expected and unexpected.

Experimental approach
(Continuous learning and adaptation)

Driving change

How to influence an organization?

Interdependencies → a system of constraints

People
→ Behavior
→ Capabilities
→ Processes
→ Structures
Eco-system tool

Behavior

Capabilities

desired state or problem statement

Structures

Processes

Our new approach

Behavior

Trust
Transparency
Empowerment
Participation
Help people and teams grow

Capabilities

Responding to change

Lead by example

Recruitment

Structures

Processes
The journey, part 2: Experimenting our way forward
Deal professionally with uncertainty

**Communicate**
- Speak up when you see something unexpected
- Accept, that uncertainty is always there
- Share views and insights
- React constructively on surprises

**Behavior**
- Manage expectations → anticipate expectations & address them
- Make uncertainty visible
- Iterate development → feedback loop

**Capabilities**
- Translate uncertainty into realistic expectations (customers, internal stakeholders)
- Distributed planning on different levels

**Structures**
- Frequent meetings to digest updates/new insights
- Show and discuss uncertainty in governance meetings

**Processes**
- Iterative development
- Communication structure
Accept, that uncertainty is always there

Basic acceptance by PdM

Share views and insights

Portfolio Mgrs

Show and discuss uncertainty in governance meetings

Deal professionally with uncertainty

Scrum

all plans with cost & time ranges

Recruit Portfolio Mgrs with “right” mindset & perseverance

Decision model

Distributed planning on different levels

Intervention 2
Accept, that uncertainty is always there.

Share views and insights.

Show and discuss uncertainty in governance meetings.

Distributed planning on different levels.

Scrum.

Basic acceptance by PdM.

Recruit Portfolio Mgrs with "right" mindset & perseverance.

Heavy debates in governance meetings.

Teams don't support the ranges.

Lift the abstraction level in governance meetings.

POs asked to support ranges.

Explain purpose of ranges.

Tool support for ranges.

Deal professionally with uncertainty.
Deal professionally with uncertainty

- Less debate in governance meetings
- Accept, that uncertainty is always there
- Basic acceptance by PdM
- Some teams hate the tool
- Lift the abstraction level in governance meetings
- Show and discuss uncertainty in governance meetings
- POs asked to support ranges
- Add “give me ranges” to process without telling “how”
- Make tool optional all plans with cost & time ranges
- Scrum
- Distributed planning on different levels
- Recruit Portfolio Mgrs with “right” mindset & perseverance
- Explain purpose of ranges & perseverance

Structures  Processes  Capabilities  Behavior

Intervention 4
Deal professionally with uncertainty

Accept, that uncertainty is always there

Show and discuss uncertainty in governance meetings

Recruit Portfolio Mgrs with "right" mindset & perseverance

In governance meetings: range is used to discuss what we can tell to the customer

Regular work/scenario meetings between Development and PdM in governance meetings

Provide customer oriented uncertainty view in governance meetings

We will hit the wall! We need the Total Project managers back!

Debete: why can’t we commit earlier?

Work on abstraction level in governance meetings

Recruit Portfolio Mgrs with "right" mindset & perseverance

Make tool optional

Scrum

Deal professionally with uncertainty

Basic acceptance by PdM

Accept, that uncertainty is always there

Recruit Portfolio Mgrs with "right" mindset & perseverance

In governance meetings: range is used to discuss what we can tell to the customer

Regular work/scenario meetings between Development and PdM in governance meetings

Provide customer oriented uncertainty view in governance meetings

We will hit the wall! We need the Total Project managers back!

Debete: why can’t we commit earlier?

Work on abstraction level in governance meetings

Recruit Portfolio Mgrs with "right" mindset & perseverance

Make tool optional

Scrum
Deal professionally with uncertainty

- Closely pair PO with PdM
- Regular work/scenario meetings between Development and PdM
- Lift the abstraction level in governance meetings
- Scrum

Behavior

- Accept, that uncertainty is always there
- Share views and insights

Capabilities

- Recruit Portfolio Mgrs with "right" mindset & perseverance
- Explain purpose of ranges and their ranges in governance meetings
- Make tool optional
- Distributed planning on different levels

Structures

- PdM and Development starting to "join forces"
- Basic acceptance by PdM
- Developers understand PdM better
- POs asked to support ranges

Processes

- Developers understand PdM better
- Closely pair PO with PdM
- Make tool optional all plans with cost & time ranges
- "Give me ranges" to process without telling "how"
- Scrum

Intervention 5
Deal professionally with uncertainty

Behavior
- We don’t need commitment decisions any more
- Accept that uncertainty is always there
- Share views and insights
- Scrum

Structures
- Regular work/scenario meetings between Development and PdM
- Closely pair PO with PdM

Capabilities
- Recruit Portfolio Mgrs with “right” mindset & perseverance
- PdM understand development better
- Developers understand PdM better
- Make tool optional

Processes
- Explain purpose of ranges in governance meetings
- Lift the abstraction level in governance meetings
- POs asked to support ranges
- Scrum
- Distributed planning on different levels
- Make commitment decisions optional
To be continued...

Summary
Summary (1/2)

An agile PMO is the operational facilitator serving the organization.

It needs to lead by example, focus on empowering teams and make them grow.

It acts like a shepherd enabling the organization to create and deliver products, that delight customers profitably.

Summary (2/2)

Try at home:
- Visualize uncertainty
- Cross-functional setup
- Project managers → operational coaching
- Participation: overlapping and empowered decision circles
- Meeting structures, that enable the effective transport of knowledge and ideas for improved decision making.
- Organizational retrospectives (respond to emerging patterns)

Create opportunities for interaction and flow of knowledge and ideas for de-centralized decision making.