NT KONF

NT KONFERENCA

25. – 27.
SEPTEMBER
2023
PORTOROŽ

# Unleashing Agile Transformation: A Scrum and Azure DevOps Approach

Ognjen Bajić, Ana Roje Ivančić

Professional Scrum Trainers, DevOps Consultants

Microsoft MVPs for Developer Technologies

Agilist IT
INFORMATION TECHNOLOGIES

Scrum.org
The Home of Scrum

# Agenda

Intro to Agile Transformations

Preparation

Our Approach to Running the Transformation

  Using Scrum to Implement Agility

Notable Practices

Case Studies

# Intro to Agile Transformations

# Agile Transformations Fail

Most Scrum implementations and agile transformations fail to realize their full potential!

*"Only 30% of all teams or organizations that use Scrum will become excellent development organizations."*

Ken Schwaber

https://kenschwaber.wordpress.com/2011/04/07/scrum-fails/

# Agile Transformations Explained

Agile Transformations are risky and difficult because they include:

- Instilling the Agile and Lean mindset
- Moving from projects to products
- Changing the culture
- Confronting the „*This is the way we do things around here*" attitude
- Setting up stable, empowered, self-managing, cross-functional teams
- Introducing new leadership management styles - Inverting traditional structures
- Cleaning the remnants of past transformations

# Scrum Helps Solving Complex Problems

How about getting help
from the Scrum Framework?

***Scrum Framework is optimized for solving complex problems where more is unknown than known.***

An organizational agile transformation is a complex problem, and Scrum can help solve it!

***Scrum is a lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems.***

# Culture Eats Strategy For Breakfast

*"This is the way we work around here!"*

Agile Transformations fail when they ignore organizational culture

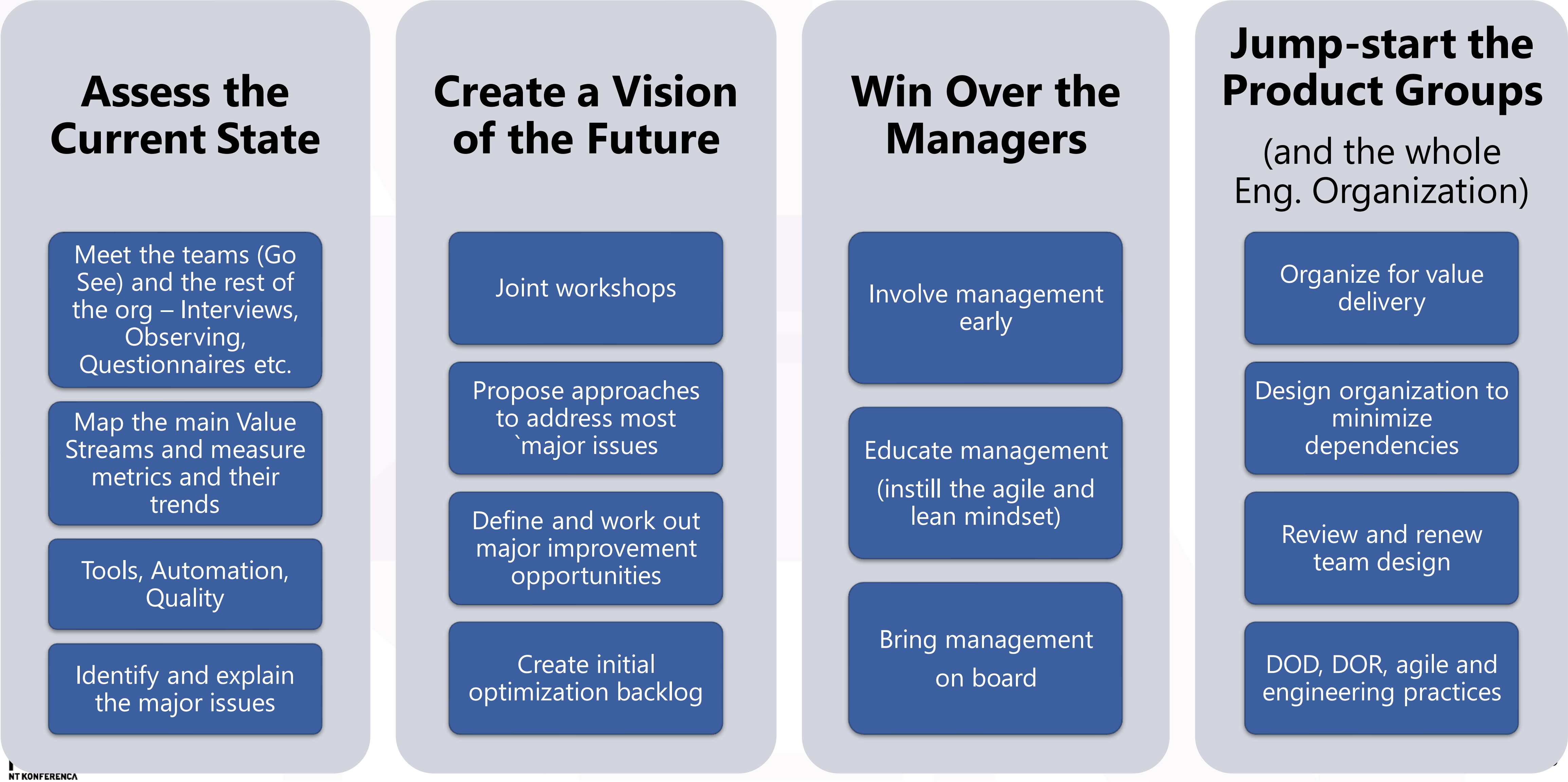
The organization needs to change its structures and control systems

People need to change attitudes, values, and assumptions that underlie their behavior and how they work

Otherwise, change will not sustain!

# Preparations

# Preparations – Assessment and More…



## Assess the Current State

- Meet the teams (Go See) and the rest of the org – Interviews, Observing, Questionnaires etc.
- Map the main Value Streams and measure metrics and their trends
- Tools, Automation, Quality
- Identify and explain the major issues

## Create a Vision of the Future

- Joint workshops
- Propose approaches to address most `major issues
- Define and work out major improvement opportunities
- Create initial optimization backlog

## Win Over the Managers

- Involve management early
- Educate management (instill the agile and lean mindset)
- Bring management on board

## Jump-start the Product Groups

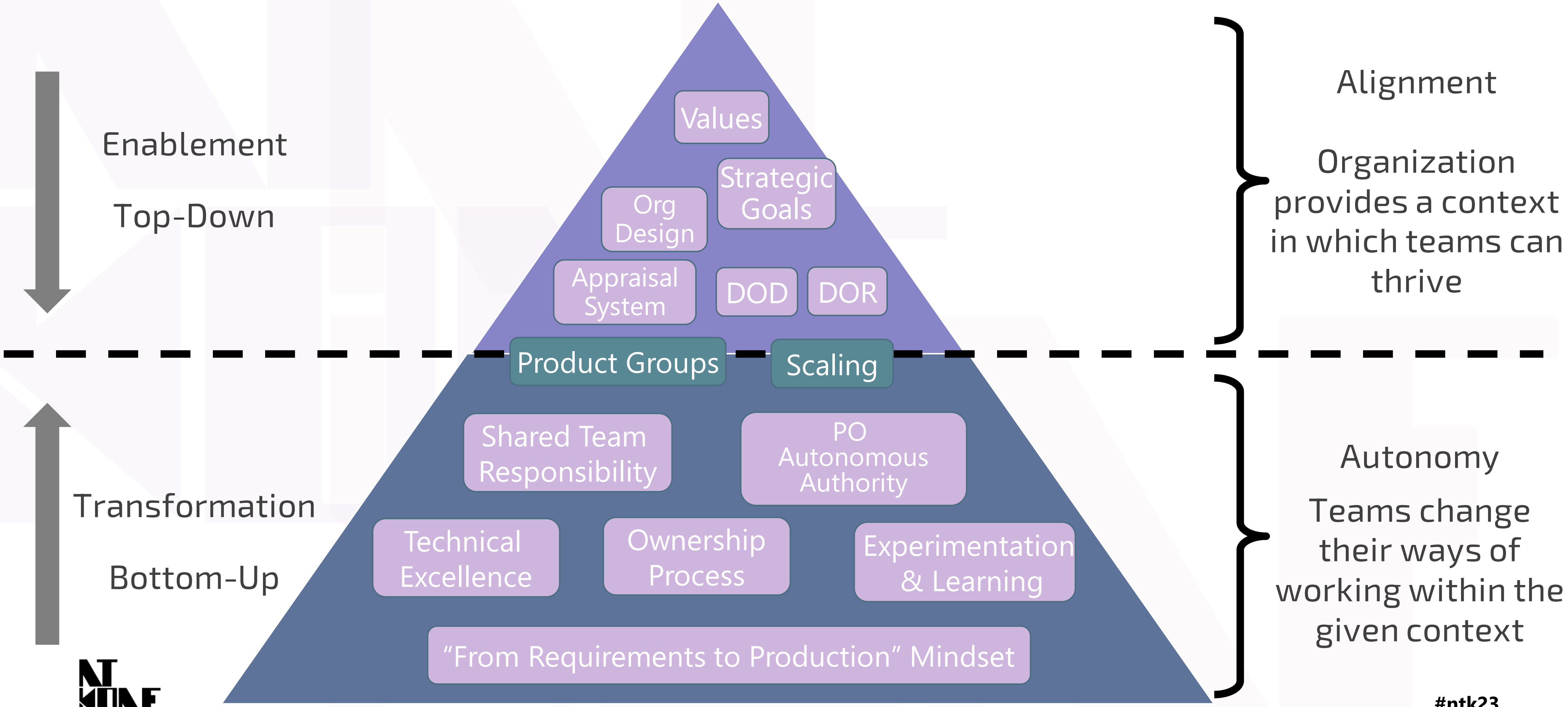
(and the whole Eng. Organization)

- Organize for value delivery
- Design organization to minimize dependencies
- Review and renew team design
- DOD, DOR, agile and engineering practices

Inspired by fig 8.1 from Ramos, Cesario; Pavhlichenko, Illia. Creating Agile Organizations A Systemic Approach, 2022

# Our Approach

# Aligned Autonomy in the Agile Transformation



Enablement

Top-Down

Values

Strategic Goals

Org Design

Appraisal System

DOD

DOR

Product Groups

Scaling

Shared Team Responsibility

PO Autonomous Authority

Technical Excellence

Ownership Process

Experimentation & Learning

"From Requirements to Production" Mindset

Transformation

Bottom-Up

Alignment

Organization provides a context in which teams can thrive

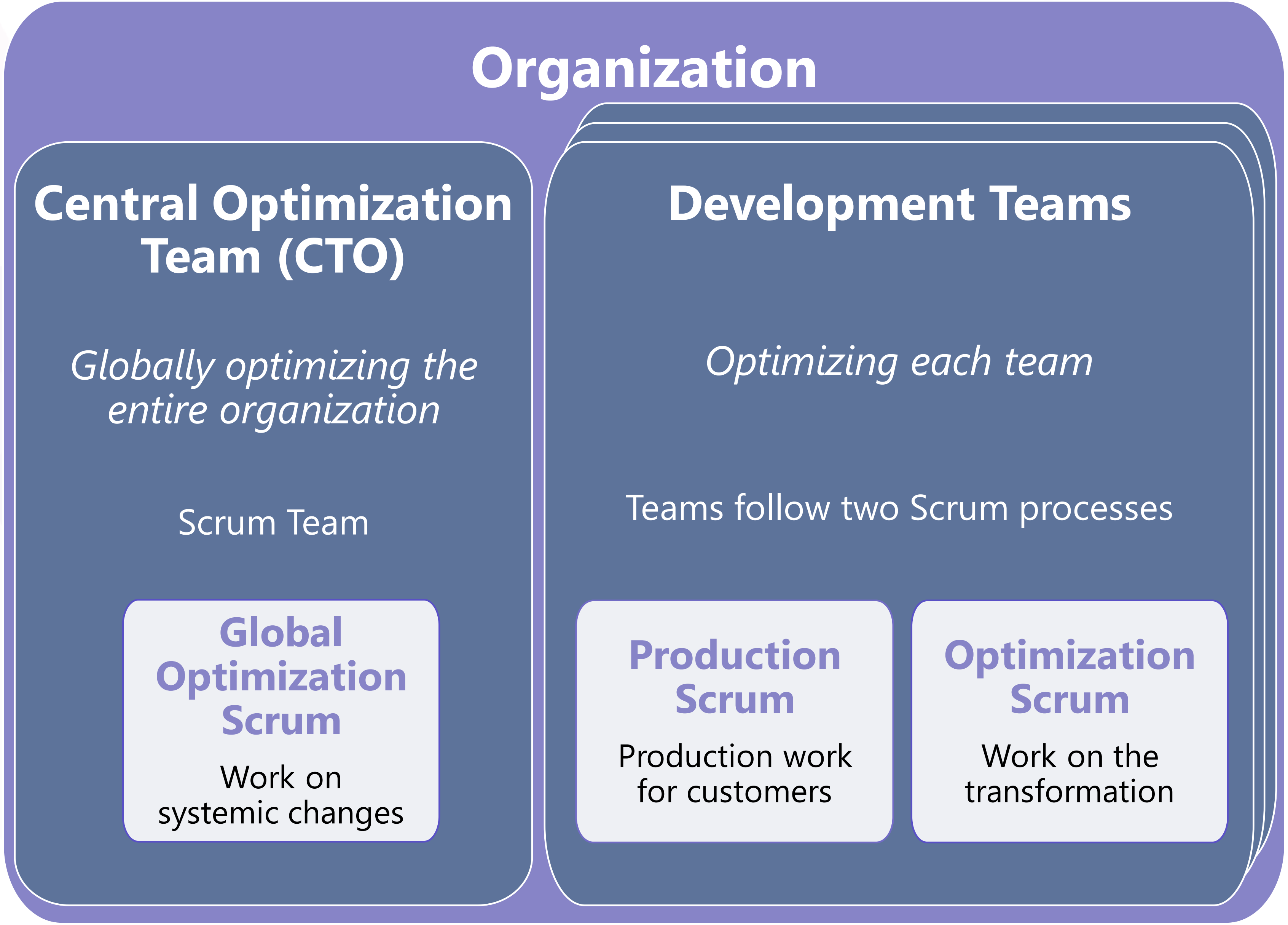
Autonomy

Teams change their ways of working within the given context

# Basic Process - Use Scrum to Introduce Agility

Activities proceed on (at least) two levels

Use production process metrics to drive the transformation



Organization

**Central Optimization Team (CTO)**

*Globally optimizing the entire organization*

Scrum Team

**Global Optimization Scrum**

Work on systemic changes

**Development Teams**

*Optimizing each team*

Teams follow two Scrum processes

**Production Scrum**

Production work for customers

**Optimization Scrum**

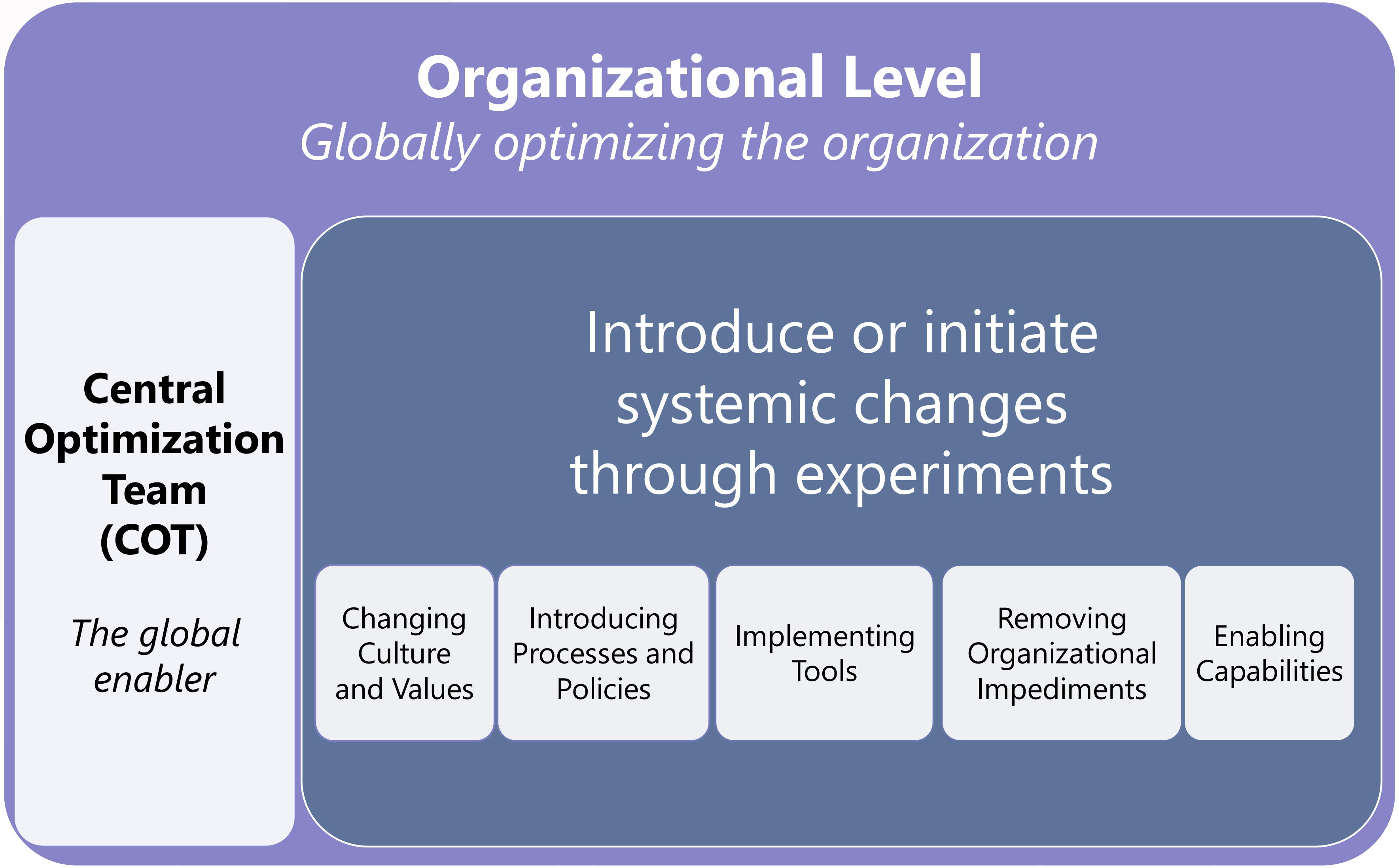
Work on the transformation

# Central Optimization Team (COT)

A Scrum Team working in sprints to globally optimize the organization

Their product is the transformed organization

Proceed based on feedback and the effects of the change



**Organizational Level**

*Globally optimizing the organization*

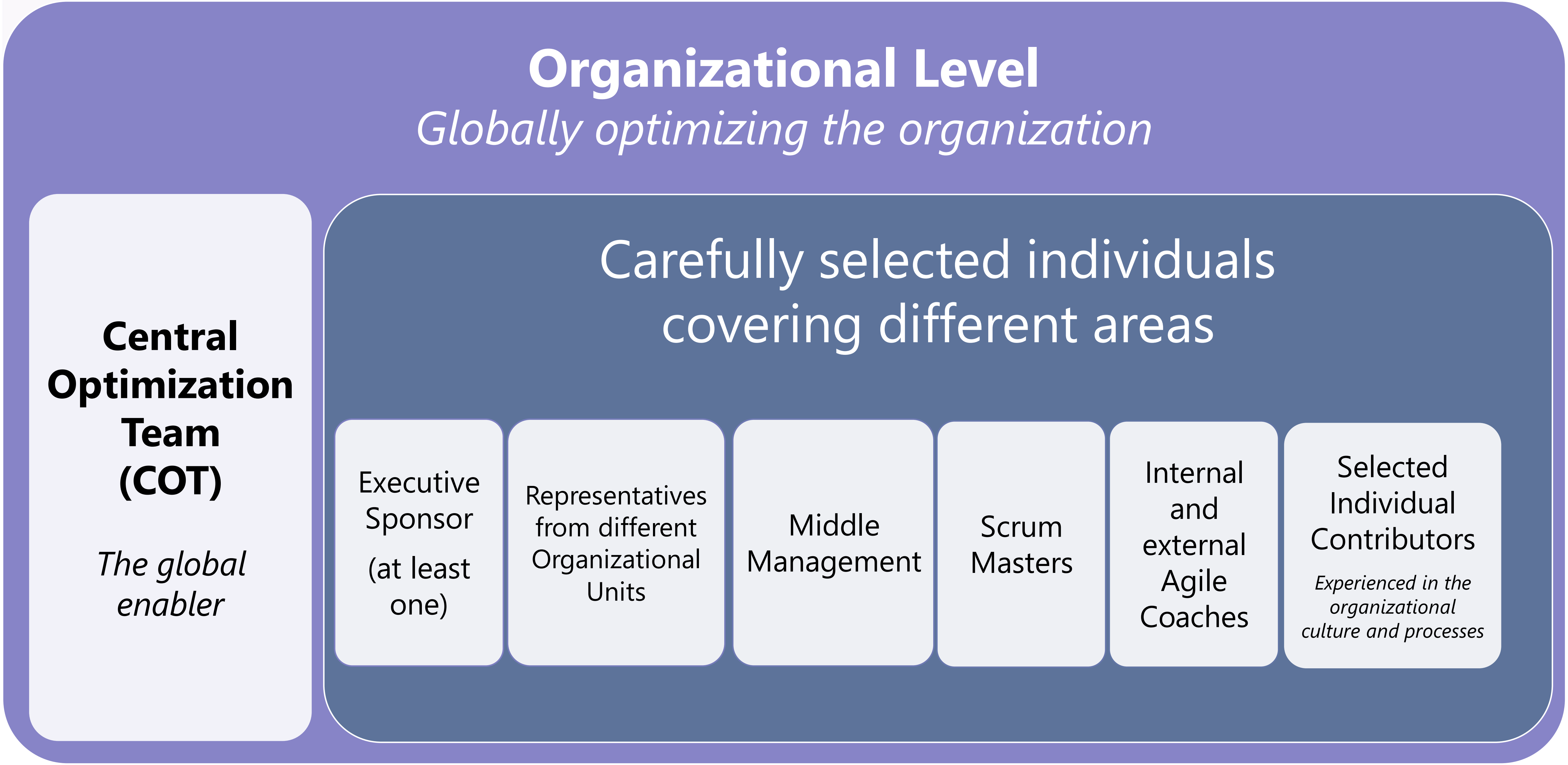
**Central Optimization Team (COT)**

*The global enabler*

Introduce or initiate systemic changes through experiments

- Changing Culture and Values
- Introducing Processes and Policies
- Implementing Tools
- Removing Organizational Impediments
- Enabling Capabilities

# COT Membership



**Organizational Level**

*Globally optimizing the organization*

**Central Optimization Team (COT)**

*The global enabler*

Carefully selected individuals covering different areas

Executive Sponsor

(at least one)

Representatives from different Organizational Units

Middle Management

Scrum Masters

Internal and external Agile Coaches

Selected Individual Contributors

*Experienced in the organizational culture and processes*

# COT Accountabilities

## Organizational Level

*Globally optimizing the organization*

### Central Optimization Team (COT)

*The global enabler*

### Product Owner

Executive level COT member is an ideal candidate

Owns the transformation as a business initiative

### Scrum Master

One of the company's SMs

### Developers

All other COT members

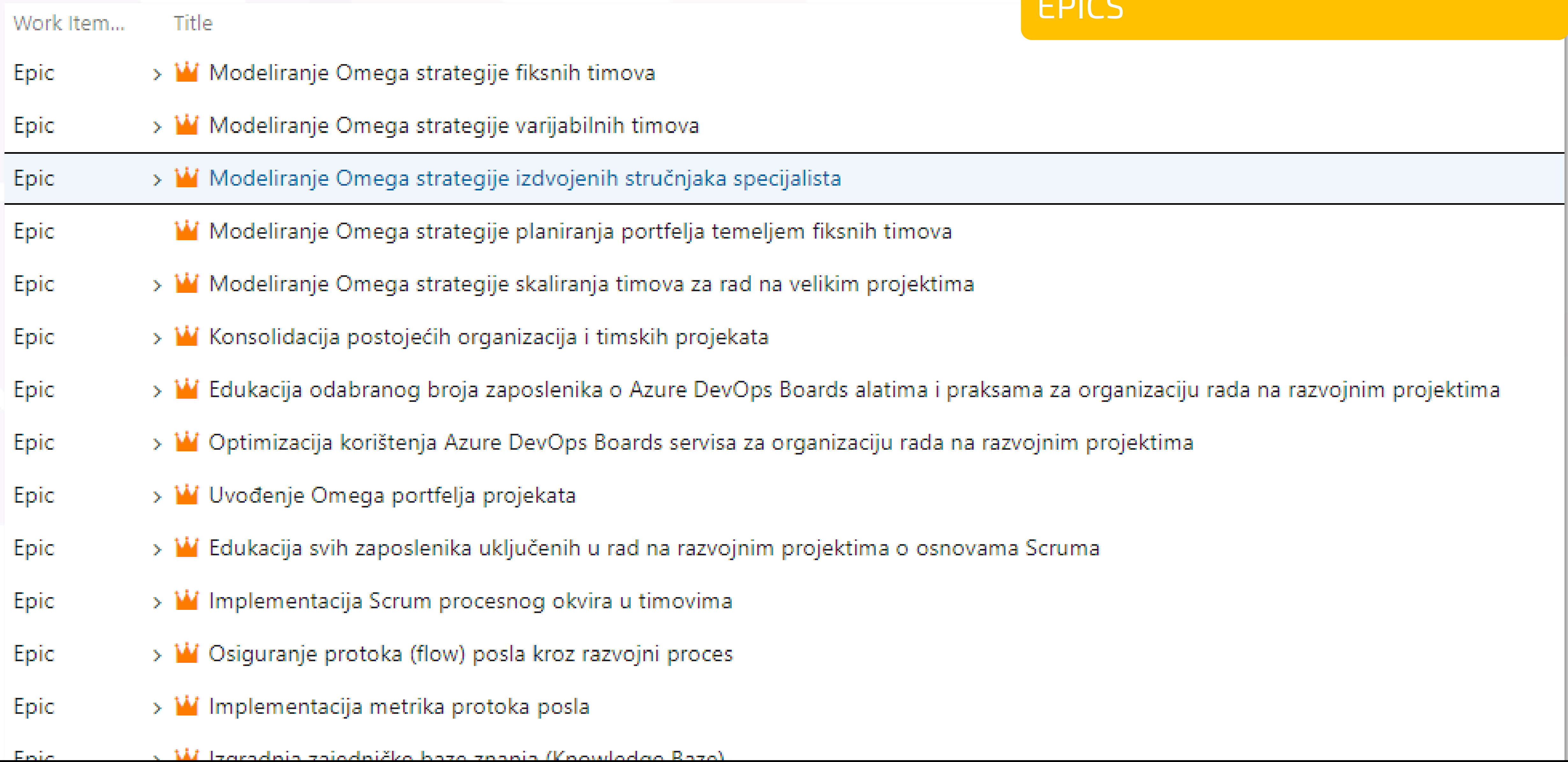
### External Coaches

Ensure proper goals, focus, and direction

Facilitate COT work
Support the PO
Support the wider organization

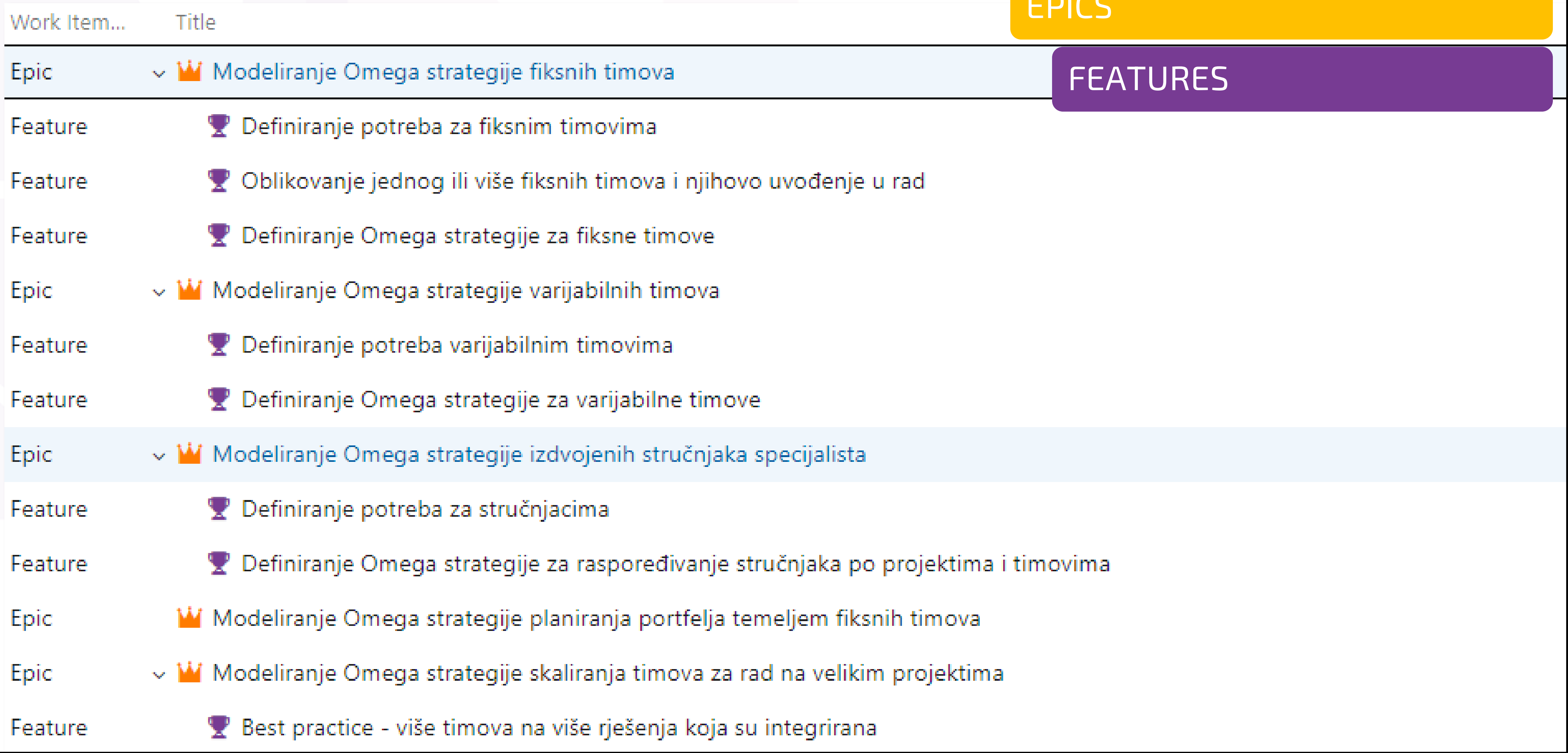
# COT Global Optimization Product Backlog

EPICS



| Work Item... | Title |
|---|---|
| Epic | Modeliranje Omega strategije fiksnih timova |
| Epic | Modeliranje Omega strategije varijabilnih timova |
| Epic | Modeliranje Omega strategije izdvojenih stručnjaka specijalista |
| Epic | Modeliranje Omega strategije planiranja portfelja temeljem fiksnih timova |
| Epic | Modeliranje Omega strategije skaliranja timova za rad na velikim projektima |
| Epic | Konsolidacija postojećih organizacija i timskih projekata |
| Epic | Edukacija odabranog broja zaposlenika o Azure DevOps Boards alatima i praksama za organizaciju rada na razvojnim projektima |
| Epic | Optimizacija korištenja Azure DevOps Boards servisa za organizaciju rada na razvojnim projektima |
| Epic | Uvođenje Omega portfelja projekata |
| Epic | Edukacija svih zaposlenika uključenih u rad na razvojnim projektima o osnovama Scruma |
| Epic | Implementacija Scrum procesnog okvira u timovima |
| Epic | Osiguranje protoka (flow) posla kroz razvojni proces |
| Epic | Implementacija metrika protoka posla |
| Epic | Izgradnja zajedničke baze znanja (Knowledge Baze) |

# COT Global Optimization Product Backlog

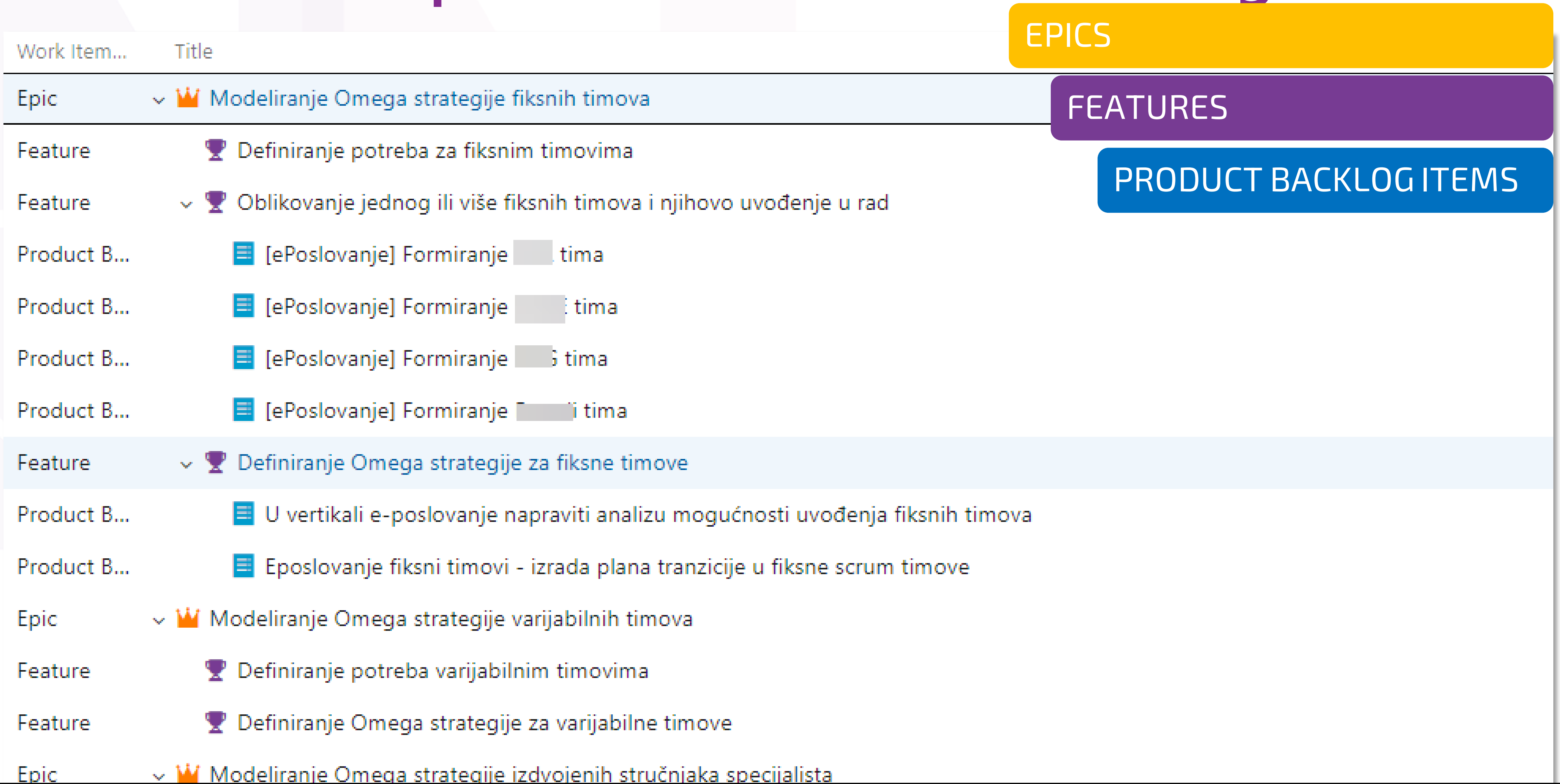


EPICS

FEATURES

| Work Item... | Title |
|---|---|
| Epic | Modeliranje Omega strategije fiksnih timova |
| Feature | Definiranje potreba za fiksnim timovima |
| Feature | Oblikovanje jednog ili više fiksnih timova i njihovo uvođenje u rad |
| Feature | Definiranje Omega strategije za fiksne timove |
| Epic | Modeliranje Omega strategije varijabilnih timova |
| Feature | Definiranje potreba varijabilnim timovima |
| Feature | Definiranje Omega strategije za varijabilne timove |
| Epic | Modeliranje Omega strategije izdvojenih stručnjaka specijalista |
| Feature | Definiranje potreba za stručnjacima |
| Feature | Definiranje Omega strategije za raspoređivanje stručnjaka po projektima i timovima |
| Epic | Modeliranje Omega strategije planiranja portfelja temeljem fiksnih timova |
| Epic | Modeliranje Omega strategije skaliranja timova za rad na velikim projektima |
| Feature | Best practice - više timova na više rješenja koja su integrirana |

# COT Global Optimization Product Backlog



EPICS

FEATURES

PRODUCT BACKLOG ITEMS

| Work Item... | Title |
|---|---|
| Epic | Modeliranje Omega strategije fiksnih timova |
| Feature | Definiranje potreba za fiksnim timovima |
| Feature | Oblikovanje jednog ili više fiksnih timova i njihovo uvođenje u rad |
| Product B... | [ePoslovanje] Formiranje [redacted] tima |
| Product B... | [ePoslovanje] Formiranje [redacted] tima |
| Product B... | [ePoslovanje] Formiranje [redacted] tima |
| Product B... | [ePoslovanje] Formiranje [redacted] tima |
| Feature | Definiranje Omega strategije za fiksne timove |
| Product B... | U vertikali e-poslovanje napraviti analizu mogućnosti uvođenja fiksnih timova |
| Product B... | Eposlovanje fiksni timovi - izrada plana tranzicije u fiksne scrum timove |
| Epic | Modeliranje Omega strategije varijabilnih timova |
| Feature | Definiranje potreba varijabilnim timovima |
| Feature | Definiranje Omega strategije za varijabilne timove |
| Epic | Modeliranje Omega strategije izdvojenih stručnjaka specijalista |

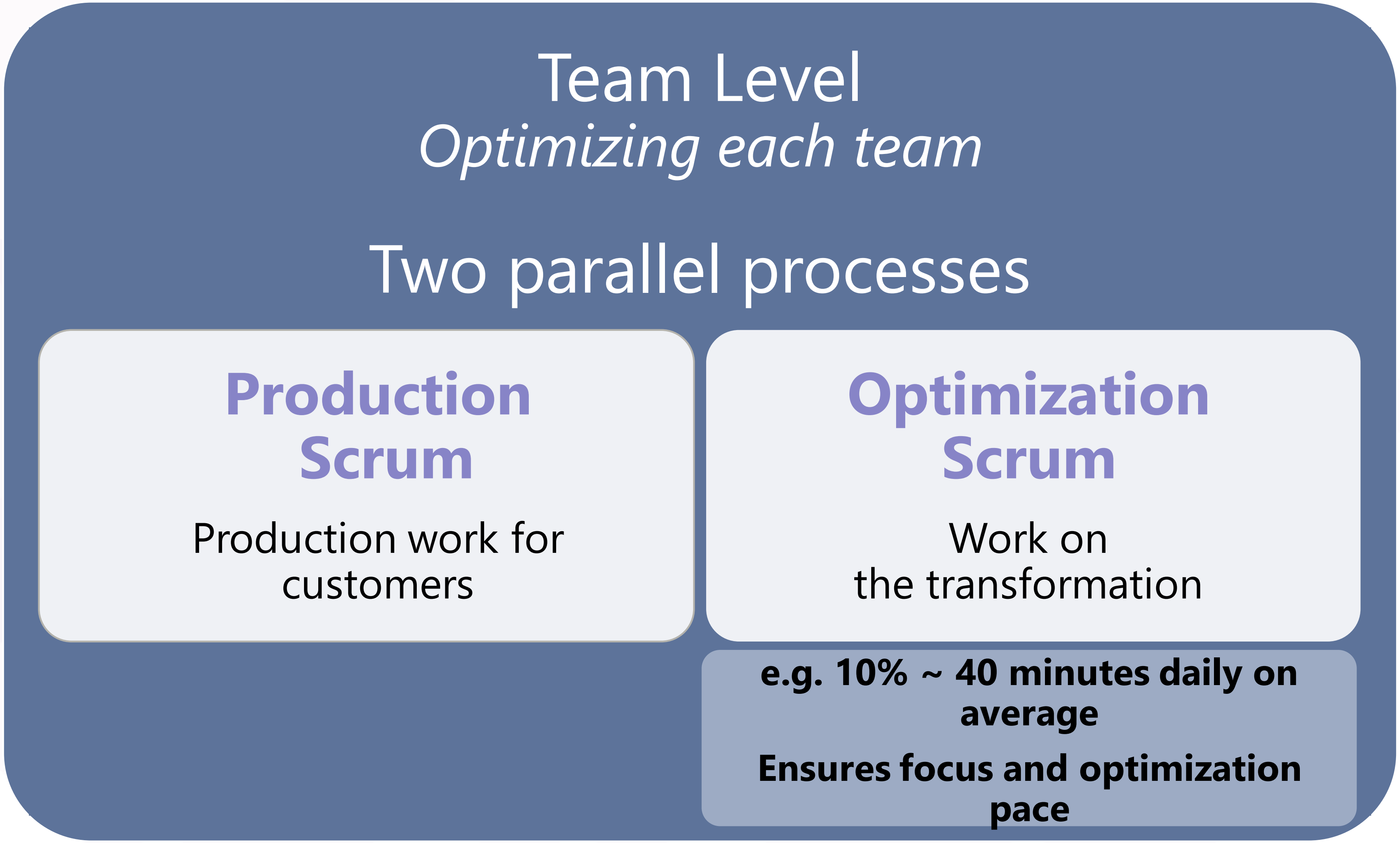
# Optimize Each Development Team

Teams own the change at their team level

Teams invest a certain percentage of their capacity in the Optimization Scrum

Coaches and/or Scrum Masters support their change

Can be influenced by other teams' changes



Team Level

*Optimizing each team*

Two parallel processes

**Production Scrum**

Production work for customers

**Optimization Scrum**

Work on the transformation

**e.g. 10% ~ 40 minutes daily on average**

**Ensures focus and optimization pace**

# Optimization Scrum - Implementation

## Empirically drives team development from zero to fully functional Scrum Team

- Introduces agile and complementary practices and tools

## Optimization backlog as a separate Product Backlog

- Initially based on the shared organizational backlog template
- Enhanced based on the team's assessment results (for existing teams)
- Ensures sustainable progress of both production and optimization work

## Optimization Scrum with all accountabilities and events

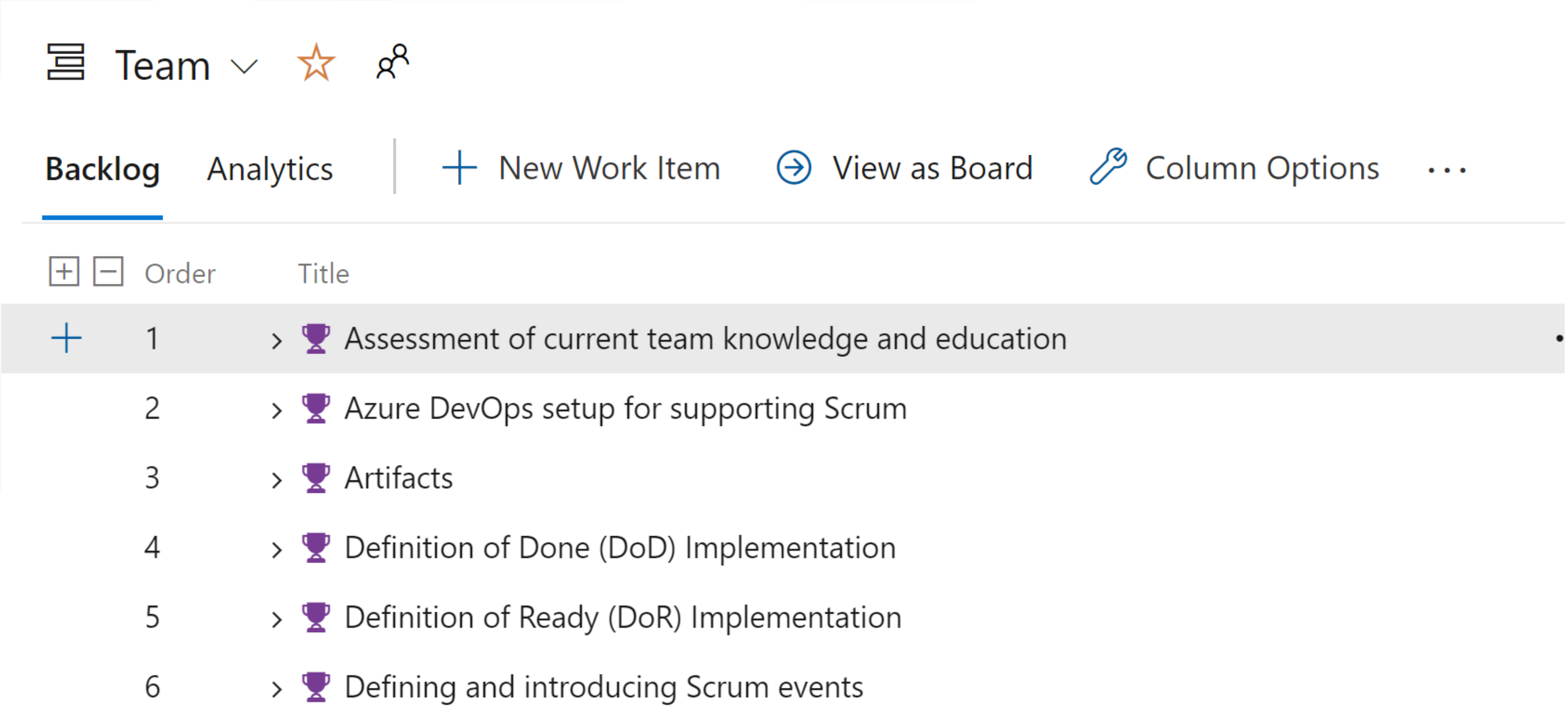
- PO - Internal or external coach, COT member
- SM - Team's SM

## Scrum Events

- Most events start with the optimization part, followed by the production part
- There is only one Sprint Retrospective
- e.g. Daily Scrum: max 15 mins for Optimization followed by max. 15 min for Production

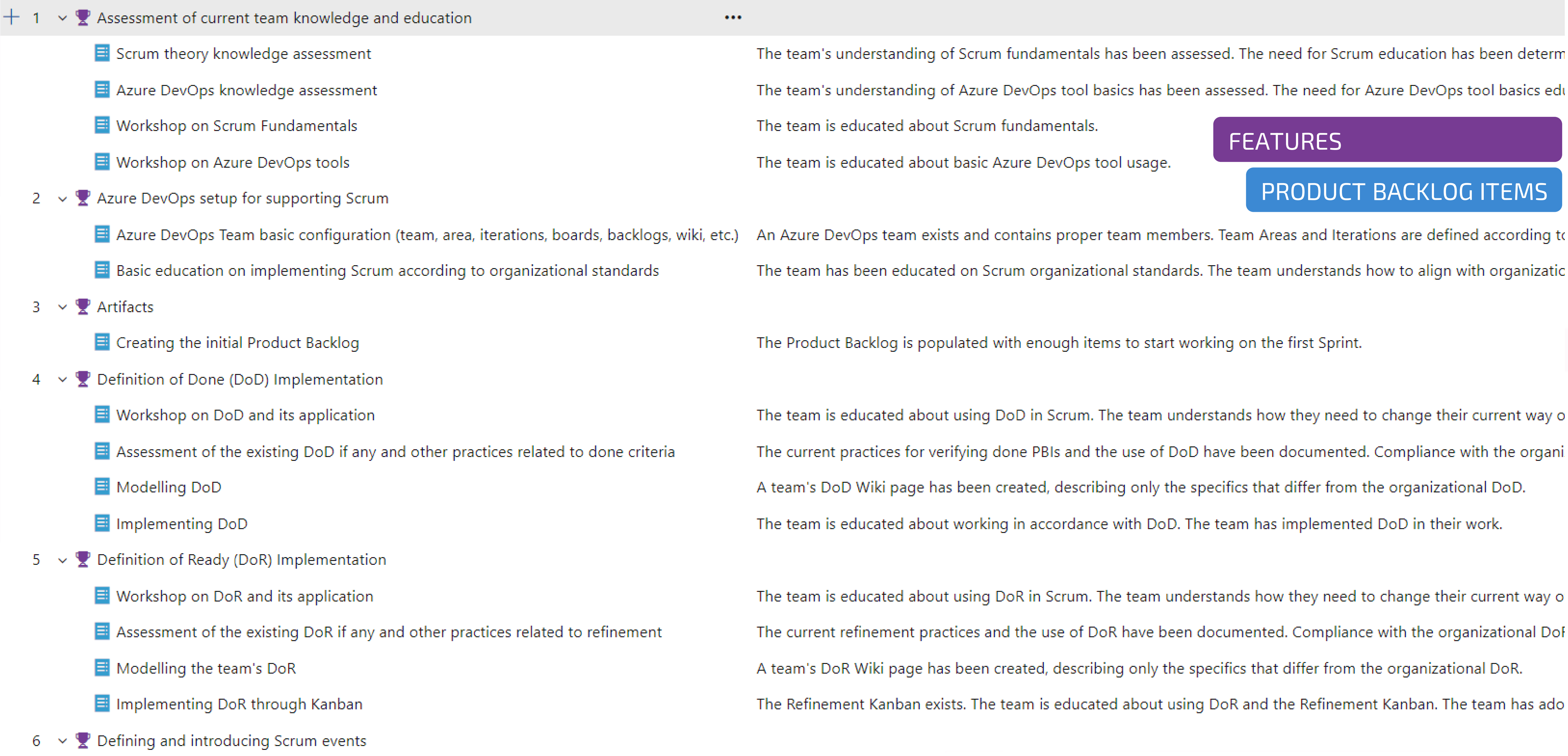
# Team Level Optimization Product Backlog Template

FEATURES



| Order | Title |
|---|---|
| 1 | Assessment of current team knowledge and education |
| 2 | Azure DevOps setup for supporting Scrum |
| 3 | Artifacts |
| 4 | Definition of Done (DoD) Implementation |
| 5 | Definition of Ready (DoR) Implementation |
| 6 | Defining and introducing Scrum events |

# Team Level Optimization Product Backlog Template



FEATURES

PRODUCT BACKLOG ITEMS

| | | |
|---|---|---|
| 1 | Assessment of current team knowledge and education | |
| | Scrum theory knowledge assessment | The team's understanding of Scrum fundamentals has been assessed. The need for Scrum education has been determ |
| | Azure DevOps knowledge assessment | The team's understanding of Azure DevOps tool basics has been assessed. The need for Azure DevOps tool basics edu |
| | Workshop on Scrum Fundamentals | The team is educated about Scrum fundamentals. |
| | Workshop on Azure DevOps tools | The team is educated about basic Azure DevOps tool usage. |
| 2 | Azure DevOps setup for supporting Scrum | |
| | Azure DevOps Team basic configuration (team, area, iterations, boards, backlogs, wiki, etc.) | An Azure DevOps team exists and contains proper team members. Team Areas and Iterations are defined according to |
| | Basic education on implementing Scrum according to organizational standards | The team has been educated on Scrum organizational standards. The team understands how to align with organizatic |
| 3 | Artifacts | |
| | Creating the initial Product Backlog | The Product Backlog is populated with enough items to start working on the first Sprint. |
| 4 | Definition of Done (DoD) Implementation | |
| | Workshop on DoD and its application | The team is educated about using DoD in Scrum. The team understands how they need to change their current way o |
| | Assessment of the existing DoD if any and other practices related to done criteria | The current practices for verifying done PBIs and the use of DoD have been documented. Compliance with the organi |
| | Modelling DoD | A team's DoD Wiki page has been created, describing only the specifics that differ from the organizational DoD. |
| | Implementing DoD | The team is educated about working in accordance with DoD. The team has implemented DoD in their work. |
| 5 | Definition of Ready (DoR) Implementation | |
| | Workshop on DoR and its application | The team is educated about using DoR in Scrum. The team understands how they need to change their current way o |
| | Assessment of the existing DoR if any and other practices related to refinement | The current refinement practices and the use of DoR have been documented. Compliance with the organizational DoF |
| | Modelling the team's DoR | A team's DoR Wiki page has been created, describing only the specifics that differ from the organizational DoR. |
| | Implementing DoR through Kanban | The Refinement Kanban exists. The team is educated about using DoR and the Refinement Kanban. The team has ado |
| 6 | Defining and introducing Scrum events | |

# Notable Practices

# DevOps Highly Correlates with Agility

Drive the transformation to improve DevOps and achieve Agility

Prefer end-to-end optimizations across the entire value-stream business process (Systems Thinking)

Proceed based on feedback and the effects of the change

Create a culture of experimentation and learning

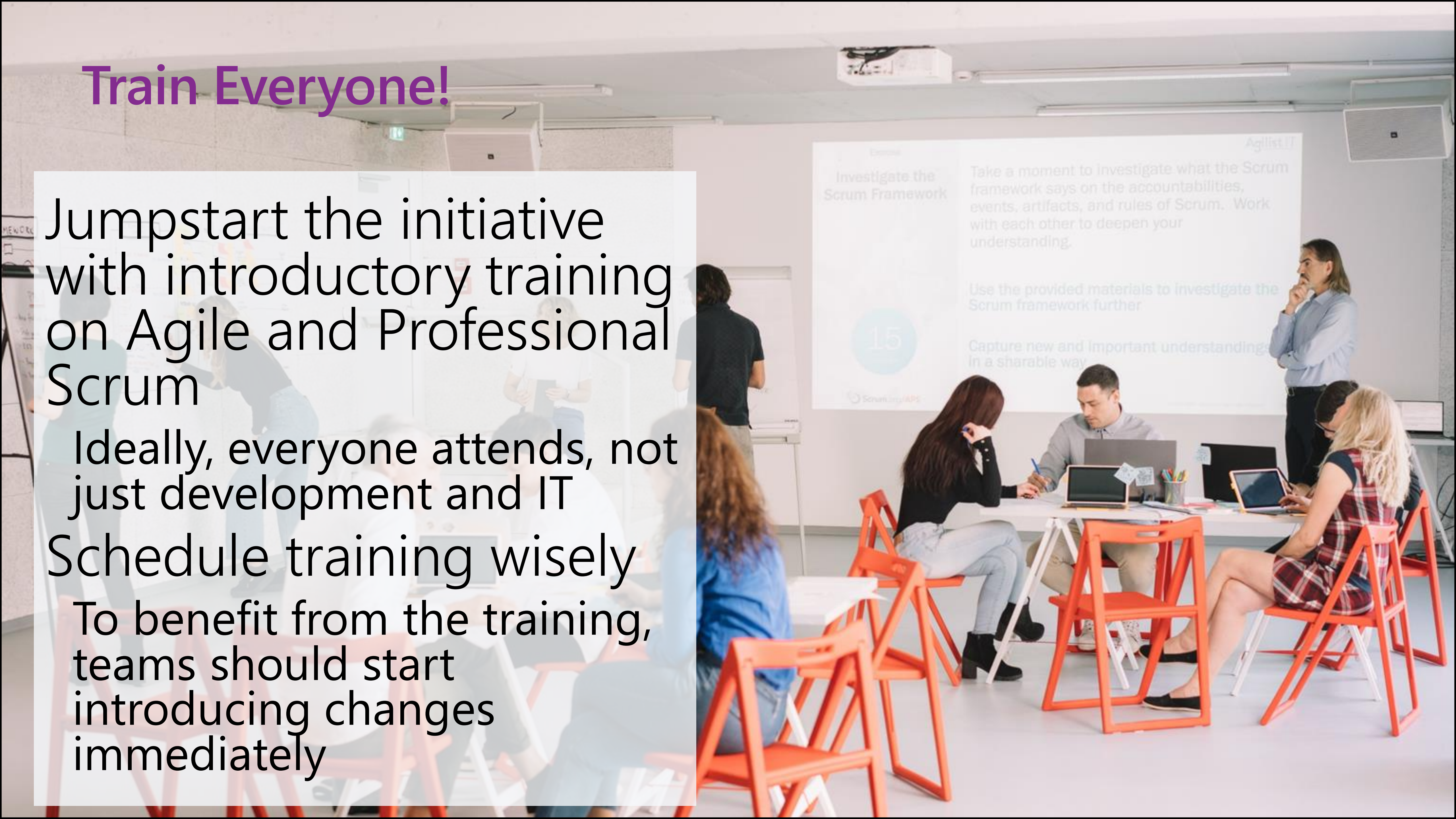
Involve the entire organization, not only development and IT

Engage sales, marketing, business, etc. in the transformation

Follow the Three Ways of DevOps

# Train Everyone!

- Jumpstart the initiative with introductory training on Agile and Professional Scrum
  - Ideally, everyone attends, not just development and IT
- Schedule training wisely
  - To benefit from the training, teams should start introducing changes immediately



Agilist IT

Exercise

Investigate the Scrum Framework

Take a moment to investigate what the Scrum framework says on the accountabilities, events, artifacts, and rules of Scrum. Work with each other to deepen your understanding.

Use the provided materials to investigate the Scrum framework further

Capture new and important understandings in a sharable way

# Drive Transformation by Metrics

## Team Level

### Flow Metrics

Cycle Time (CT)
Throughput
Work In Progress (WIP)
Work Item Age

## Product Group Level

### "Accelerate" Metrics

Cycle Time (CT)
Deployment frequency (DF)
Change Fail Rate (CFR)
Mean Time To Recovery (MTTR)

## Organization Level

### Evidence-Based Management Metrics (EBM)

Current Value (CV)
Unrealized Value (UV)
Time to Market (T2M)
Ability to Innovate (A2I)

# Make Use of Liberating Structures

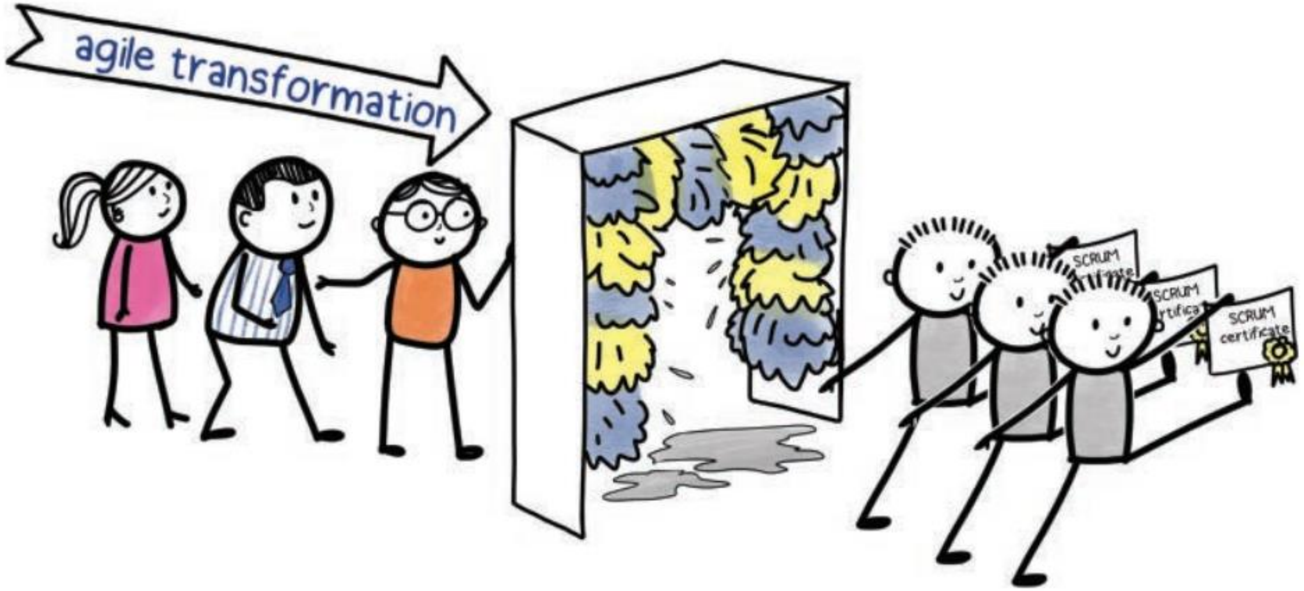
## *e.g.: Shift and Share for the Sprint Review*

Easy-to-learn techniques to facilitate meetings and conversations in groups of any size

- Effective for any interaction within the team
- Useful for enhancing any Scrum event
- Crucial when reaching out to stakeholders (for the feedback!)

Image Source: https://www.linkedin.com/pulse/shift-share-scrum-barry-overeem/

# Beware of Zombie Scrum



Teams doing the motions, but without a living heartbeat

- They do not inspect and adapt based on feedback
- They don’t live the Scrum values

Main symptoms (according to survey.zombiescrum.org):

- Teams do not know stakeholder needs
- Teams do not ship often
- Teams do not improve
- Teams do not self-manage

Image Source: Zombie Scrum Survival Guide, Verwijs, Schartau, Overeem, 2021

# Case Studies

# Case Studies

//webmarketing

Transformation ongoing since July 2020

Web and

E-commerce

Solutions

500+ Web Sites

250+ Clients

Cross-Functional Team

10 Business Analysts, Designers and Developers

omegasoftware

Transformation ongoing since September 2020

Leading Croatian IT Company

Custom apps, ERP, Enterprise Solutions, Smart City, …

10 Cross-Functional Teams

120+ Business Consultants, Developers

# Similar Challenges

## Inefficient Process

Gated waterfall process with customer approvals blocking the work

## Frequent Task Switching

Difficulties balancing priorities for ongoing projects

## Delays

Difficulties keeping up with the project schedule

## Quality Issues

Bugs, unfinished features

## Unclear Accountabilities

Micromanagement

# Similarities

Early management involvement

The majority attended formal training

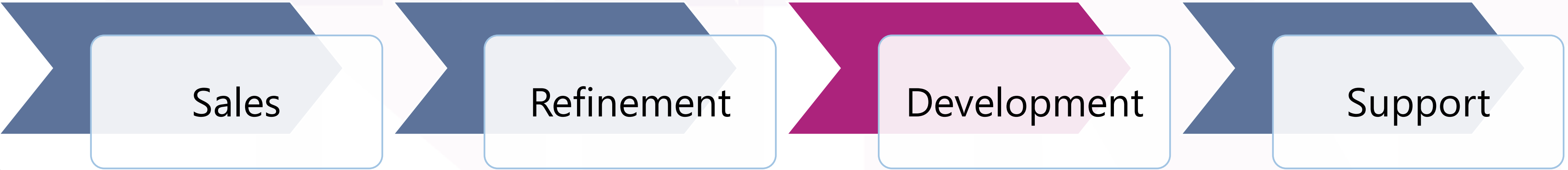
Needed systemic organizational AND team level optimizations

- Project teams replaced by stable (product) teams
- Clear accountabilities within teams (PO, SM, devs) and beyond (PMO, sales)

Supporting Business as usual is critical

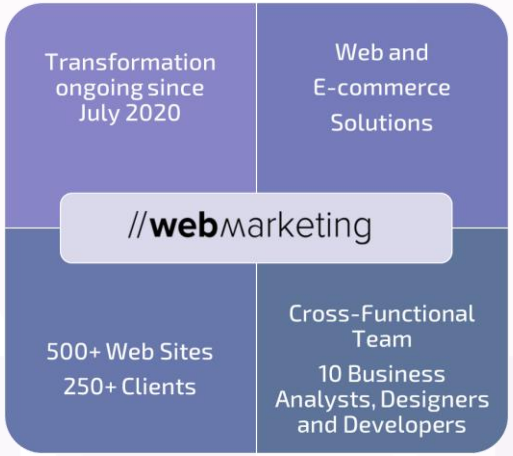
Tool's standardization as an enabler (do it during the preparation)

Desire to globally optimize the end-to-end value stream



🟦 Kanban 🟪 Scrum

# Differences



Transformation ongoing since July 2020

Web and E-commerce Solutions

//webmarketing

500+ Web Sites
250+ Clients

Cross-Functional Team
10 Business Analysts, Designers and Developers

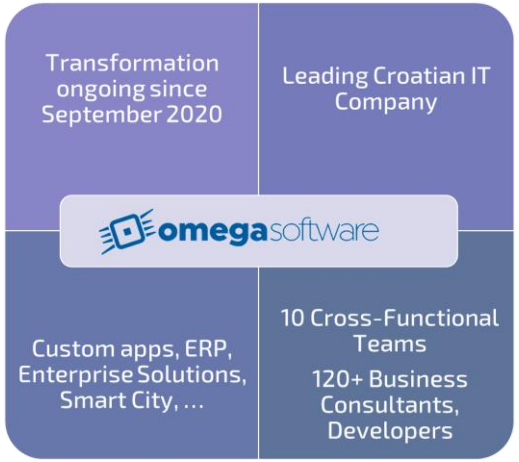
1 Team

No need for COT

External coach role

- Embedded in the production team
- Optimization process PO
- Coaching everyone

End-to end scope within the 1st year



Transformation ongoing since September 2020

Leading Croatian IT Company

omegasoftware

Custom apps, ERP, Enterprise Solutions, Smart City, ...

10 Cross-Functional Teams
120+ Business Consultants, Developers

10 Teams

COT plays a critical role

External coaches' role

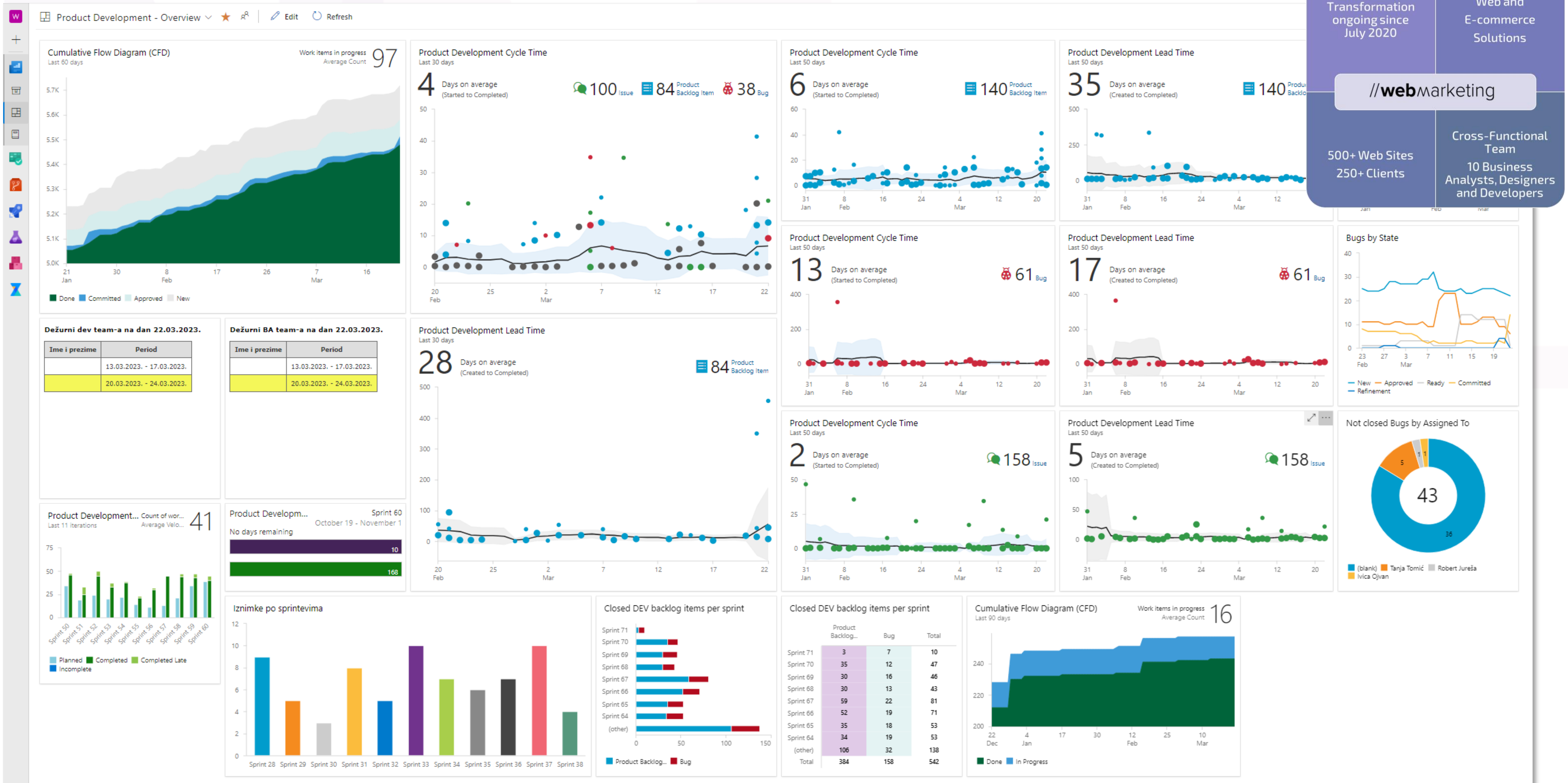
- Embedded in COT
- Consult the COT PO
- Indirectly influence production teams through COT members or COT Increments

Phased approach, optimize sales in the 2nd year

# Improvements

- Scrum with clear accountabilities
- Empowered single Product Owner(s)
- End-to-end process supported by Azure DevOps
- Transparent status insight
- Improved requirements +DoD → Improved product
- Minimized impact of support and incidents on ongoing development

# Product Development Dashboard in Azure DevOps



Cumulative Flow Diagram (CFD)
Last 60 days
Work items in progress 97
Average Count

Product Development Cycle Time
Last 30 days
4 Days on average
(Started to Completed)
100 Issue 84 Product Backlog Item 38 Bug

Product Development Cycle Time
Last 50 days
6 Days on average
(Started to Completed)
140 Product Backlog Item

Product Development Lead Time
Last 50 days
35 Days on average
(Created to Completed)
140 Product Backlog Item

**Dežurni dev team-a na dan 22.03.2023.**

| Ime i prezime | Period |
|---|---|
| | 13.03.2023. - 17.03.2023. |
| | 20.03.2023. - 24.03.2023. |

**Dežurni BA team-a na dan 22.03.2023.**

| Ime i prezime | Period |
|---|---|
| | 13.03.2023. - 17.03.2023. |
| | 20.03.2023. - 24.03.2023. |

Product Development Lead Time
Last 30 days
28 Days on average
(Created to Completed)
84 Product Backlog Item

Product Development Cycle Time
Last 50 days
13 Days on average
(Started to Completed)
61 Bug

Product Development Lead Time
Last 50 days
17 Days on average
(Created to Completed)
61 Bug

Bugs by State

Product Development... Count of wor... 41
Last 11 iterations
Average Velo...

Product Developm... Sprint 60
October 19 - November 1
No days remaining
10
168

Product Development Cycle Time
Last 50 days
2 Days on average
(Started to Completed)
158 Issue

Product Development Lead Time
Last 50 days
5 Days on average
(Created to Completed)
158 Issue

Not closed Bugs by Assigned To
43

Iznimke po sprintevima

Closed DEV backlog items per sprint

Closed DEV backlog items per sprint

| | Product Backlog... | Bug | Total |
|---|---|---|---|
| Sprint 71 | 3 | 7 | 10 |
| Sprint 70 | 35 | 12 | 47 |
| Sprint 69 | 30 | 16 | 46 |
| Sprint 68 | 30 | 13 | 43 |
| Sprint 67 | 59 | 22 | 81 |
| Sprint 66 | 52 | 19 | 71 |
| Sprint 65 | 35 | 18 | 53 |
| Sprint 64 | 34 | 19 | 53 |
| (other) | 106 | 32 | 138 |
| Total | 384 | 158 | 542 |

Cumulative Flow Diagram (CFD)
Last 90 days
Work items in progress 16
Average Count

Transformation ongoing since July 2020

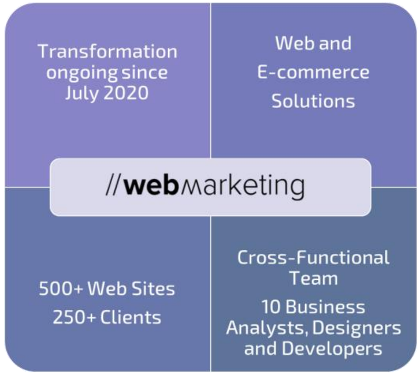
Web and E-commerce Solutions

//webmarketing

500+ Web Sites
250+ Clients

Cross-Functional Team
10 Business Analysts, Designers and Developers

# Metrics: Example of Changes 2021-2022



Transformation ongoing since July 2020

Web and E-commerce Solutions

//webmarketing

500+ Web Sites
250+ Clients

Cross-Functional Team
10 Business Analysts, Designers and Developers

Much improved Development process!

4x improved cycle time for Bugs

7x improved cycle time for Incidents (issues)

10% improved cycle time for PBI

2,3x fewer open Bugs

Need more Development team capacity!

2,5x longer lead time for PBIs

# Metrics: Bug Cycle Time Improvements 2021-2023

Transformation ongoing since July 2020

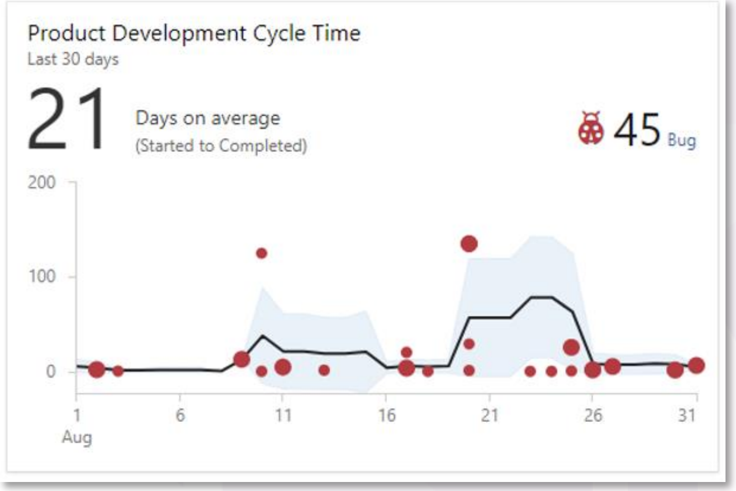
Web and E-commerce Solutions

//webmarketing

500+ Web Sites
250+ Clients

Cross-Functional Team
10 Business Analysts, Designers and Developers

## August 2021

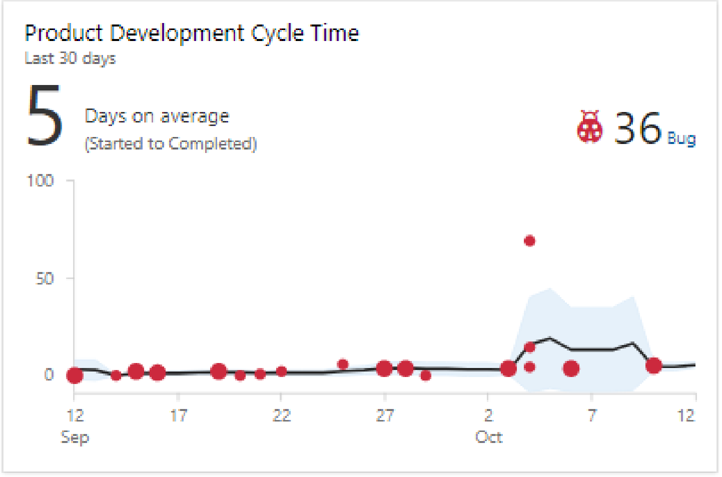


Product Development Cycle Time
Last 30 days

21 Days on average
(Started to Completed)

45 Bug

## October 2022

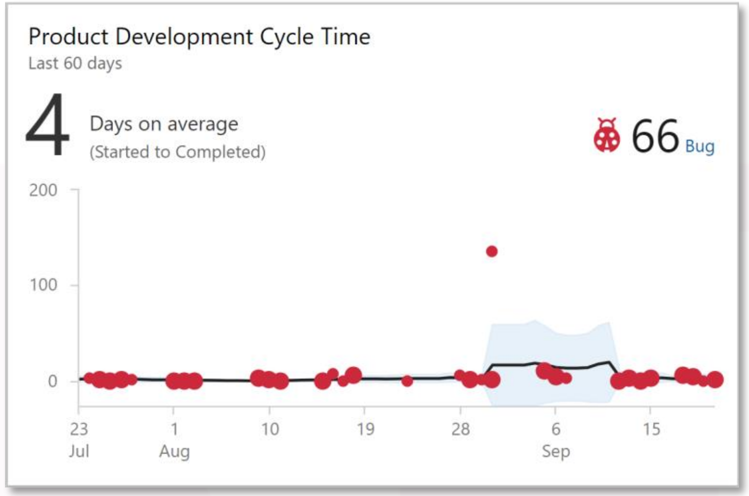


Product Development Cycle Time
Last 30 days

5 Days on average
(Started to Completed)

36 Bug

## September 2023



Product Development Cycle Time
Last 60 days

4 Days on average
(Started to Completed)

66 Bug

# Scrum.org: Case Study on Omega Software Agile Transformation

OMEGA SOFTWARE USES SCRUM TO IMPLEMENT SCRUM AND TO DRIVE ITS AGILE TRANSFORMATION

INCREASING CLIENT SATISFACTION, TEAM COLLABORATION, MORALE AND MOTIVATION

September 2023

Scrum.org

CASE STUDY

omegasoftware

THE COMPANY

Omega Software ltd. prides themselves on their commitment to quality and customer satisfaction.

For more than 20 years, Omega Software Ltd. has been one of the leading providers of ICT solutions in Croatia, specializing in developing efficient business solutions for both the public and private sectors. Their mission is to help their clients digitize and modernize their business value streams through innovative, tailor-made software solutions, modern ERP systems, and futuristic Internet of Things (IoT) solutions that transform the way people interact with cities. They are at the forefront of modern digital trends developing ICT systems that have already improved business operations for more than 300,000 users. They pride themselves on their commitment to quality and customer satisfaction.

Most Omega Software products and services are designed for public procurement projects, which typically have a fixed scope, budget, and completion date. Core offerings include digital transformation systems such as Document and Case Management (DMS) and filing systems, enterprise resource planning (ERP) systems, and IoT solutions. They also provide custom development services to meet the unique needs of their clients. Omega Software DMS and ERP products are primarily tailored for the public sector, while their IoT solutions cater to both the public and private sectors. Centrix is a modern, comprehensive, and fully integrated office business management solution that is trusted by over 10,000 satisfied users to streamline their daily work. It combines a robust case and document management system that is fully aligned with the latest regulations for office operations and accompanying legislation. eVisitor is an innovative and comprehensive information system designed for tourist registration and de-registration, connecting all tourist boards across the Republic of Croatia.

© 2023 Scrum.org All Rights Reserved | Page 1

Omega Software

CASE STUDY | SCRUM

## The Challenge

Omega Software previously used Waterfall-like processes to develop most products for public procurement projects. They had many challenges related to client satisfaction, team composition, requirement clarity, backlog management, production support issues, and a turbulent development process. In practice, they never complied with strict waterfall process rules. They tended to adjust requirements along the way and needed to approach the later project stages iteratively and apply some agile principles. In most cases, they assembled new teams for each new project, transferring team members from other project teams whose workloads were decreasing according to their plans. This way of working was unsustainable in the mid-and long-term. A simple slippage of one project caused a ”butterfly effect” on the entire project portfolio, and constant replanning was necessary. Project teams were short-lived and had no clear guidance in processes, practices, and usage of tools. They were often just a group working on the same project with a narrow focus on delivery. The teams were not self-managed, and work was often assigned to specific people.

The lack of clear accountabilities for backlog management and refinement resulted in unclear requirements that weren’t ready for development and caused frequent rework. Teams complained about unclear priorities and scope and had issues with estimating and planning.

They practiced the ”big design upfront” approach and ”late testing”. Such an approach did not leave much room for updating the backlog based on stakeholder feedback. Teams were stressed by feedback instead of welcoming it to better satisfy clients’ needs. This decreased process predictability and negatively affected the development process flow.

Another significant negative impact resulted from the need to support existing clients and their production environments. Teams that previously worked on these solutions did not exist anymore, and they were forced to reassemble them fully or partially on occasion. This negatively impacted the performance of existing teams whose team members were (temporarily) reassigned to their previous teams. As a result, their plans and delivery schedules were seriously challenged. They allowed constant unplanned interruptions from different channels, like support and sales. This negatively influenced flow and created additional stress.

Regarding reporting and metrics, they manually reported on project progress and tracked only the working hours. As a result, they lacked a transparent insight into the project portfolio state. They did not have relevant, real-time data for making informed project and product portfolio management-related decisions.

Microsoft Azure DevOps was already their DevOps platform of choice, but its usage was not standardized across teams. Due to lacking knowledge and experience with this platform, departments, and teams used different patterns and practices. Teams were scattered across separate Azure DevOps organizations and team projects. Many useful platform features were completely ignored, like area paths, Backlog hierarchies, ordering of Backlog items, capacity planning, Kanban boards, team hierarchies, portfolio management, analytics and charts, dashboards, Wiki pages, and Test plans.

### Beginning their Agile Transformation

Omega Software began an Agile transformation initiative in September 2020, building on the foundation of a few teams called ”agile pioneer teams” that were already using some Agile approaches. Prior to this, they experimented with structured teams of 10+ people working on multiple projects and solutions for various clients. While their composition was relatively stable and they had some agile practices in place, such as daily meetings, the reality was that team members were divided into sub-teams focused on specific solutions and dedicated to individual projects. This made the overall team collaboration rather inefficient. Therefore, they decided to transition to project-based teams.

In line with the new approach, they would create a team for a specific project purpose and dissolve it upon project completion. In most cases, this group of people lacked time to mature and evolve into an efficient unit. A few long-term project teams were exceptions to this rule, with enough time to

Scrum.org

© 2023 Scrum.org All Rights Reserved | Page 2

https://aglst.com/OmegaCaseStudy-PPTX

#ntk23

# Closing

# Call to Action

Don’t delay your transformation any longer!

Consider an iterative and incremental approach

Parallel processes: Scrum to introduce Agility, Scrum for Production

Introduce agile values and principles, a lean mindset

Organizational structure and leadership as an enabler

Three Ways of DevOps, Metrics

Beware of Zombie Scrum

Experiment, measure/get feedback, continue or pivot

Let us know about your successes and challenges

Agilist IT
INFORMATION TECHNOLOGIES

Ognjen Bajić
obajic@agilist.hr
Ana Roje Ivančić
arojeivancic@agilist.hr
www.agilitistit.com
training@agilist.hr

APPLYING PROFESSIONAL SCRUM
Scrum.org
APS
PSM I

APPLYING PROFESSIONAL SCRUM
Scrum.org
APS SOFTWARE
PSD I

Conference attendees get 20% discount on course prices. Contact us at training@agilist.hr

NT KONF
NT KONFERENCA

PST Professional Scrum Trainer
Scrum.org

MVP Microsoft® Most Valuable Professional