



# Applying Agile to a Web Development Agency

## A Case Study

Ognjen Bajić, Ana Roje Ivančić

Professional Scrum Trainers, DevOps Consultants

Microsoft MVPs for Developer Technologies



# Agenda

Applying Agile in a Web Agency

Making Sense of Doing UX Discovery and Development Together

Case Study

Web Development Agency Powered by Scrum and Kanban

# Speakers: Ana Roje Ivančić and Ognjen Bajić

In Software Development since 1996/1999

Helping teams improve DevOps practices since 2005

Worked as Dev, PM, Test, RM, SM, PO...

Microsoft MVP for Developer Technologies  
(ALM/DevOps) since 2006/2016

Professional Scrum Trainer (PST) for Scrum.org since 2018



**Ana Roje Ivančić**, [arojeivancic@agilist.hr](mailto:arojeivancic@agilist.hr)

**Ognjen Bajić**, [obajic@agilist.hr](mailto:obajic@agilist.hr)

Zagreb, Croatia



[www.agilitistit.com](http://www.agilitistit.com)

[trainings@agilist.hr](mailto:trainings@agilist.hr)



# Applying Agile in a Web Agency

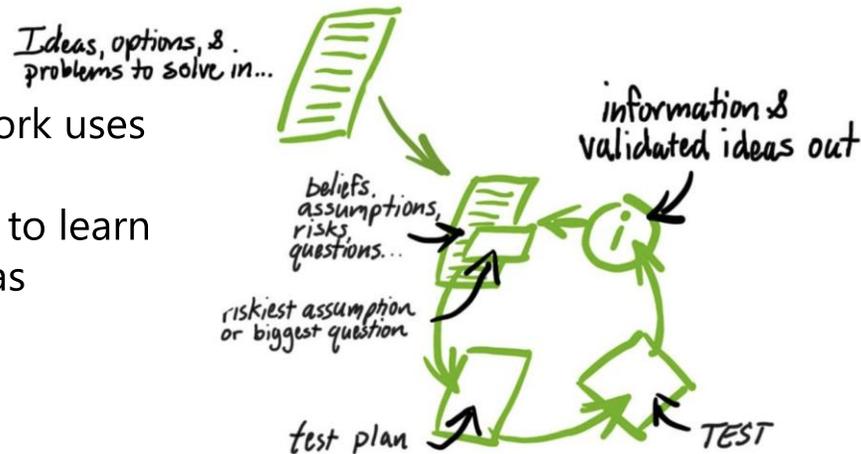


# Doing UX Discovery and Product Development Together

Product Development consists of two types of work

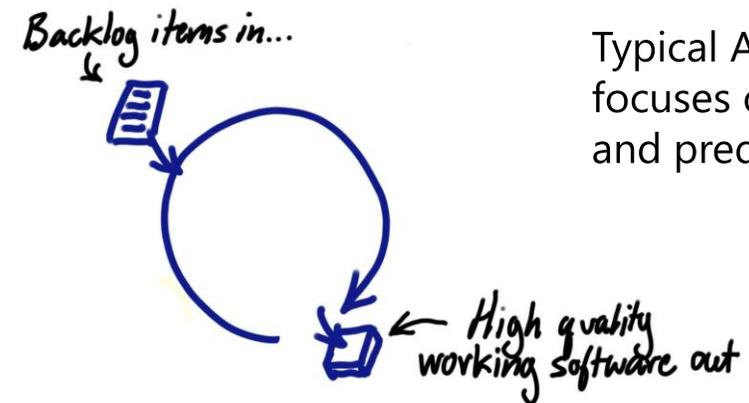
## (UX) Discovery work

Discovery work uses time-boxed experiments to learn and test ideas



## Development work

Typical Agile focuses on quality and predictability



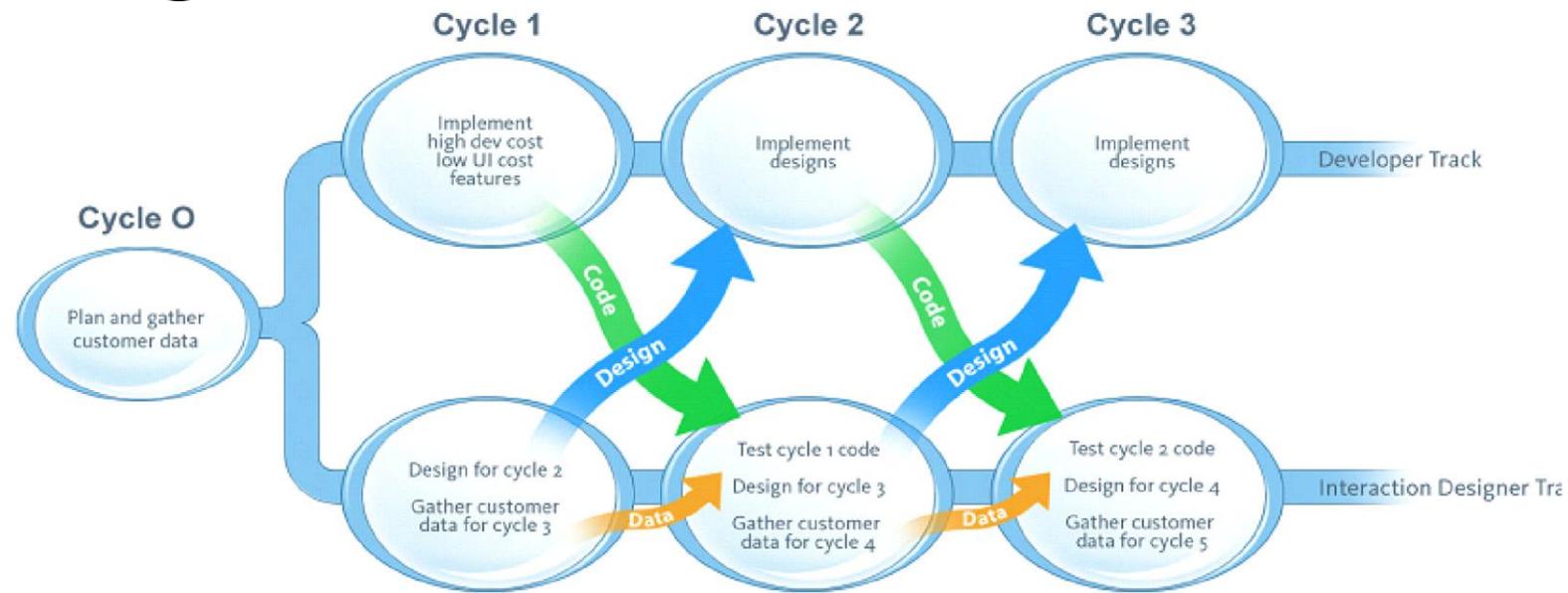
## MAXIMIZE LEARNING VELOCITY

Focus on fast learning and validation  
Rapid iterations (hours, days)

## MAXIMIZE DELIVERY VELOCITY

Focus on delivery of high-quality software  
Scrum Sprints (30 days or less)

# Dual-Track Agile



The concept of “Dual-Track Agile” is often represented with a graphic like this.

How do you think teams typically interpret and implement this?

# Busting “Dual-Track Agile” Myths

MYTHS	FACTS
Discovery is a process that precedes agile development	Discovery is a necessary part of product development. Practice it with the same agile and lean principles in mind
All work moves from discovery to development	If we’re doing discovery right, we substantially change and kill lots of ideas
Once you move from discovery to development, discovery work is done	Keep measuring and learning even after you ship
The discovery team is different than the development team	NO – keep discovery work and progress visible to the whole team

# Scrum Supports Emergent (Lean) UX

UX discovery tasks can be done simultaneously with other PBI work: coding, testing, etc.

Some UX things can be on the Definition of Done and apply universally to every PBI

E.g., adherence to the style/brand guide, all features are instrumented with customer analytics

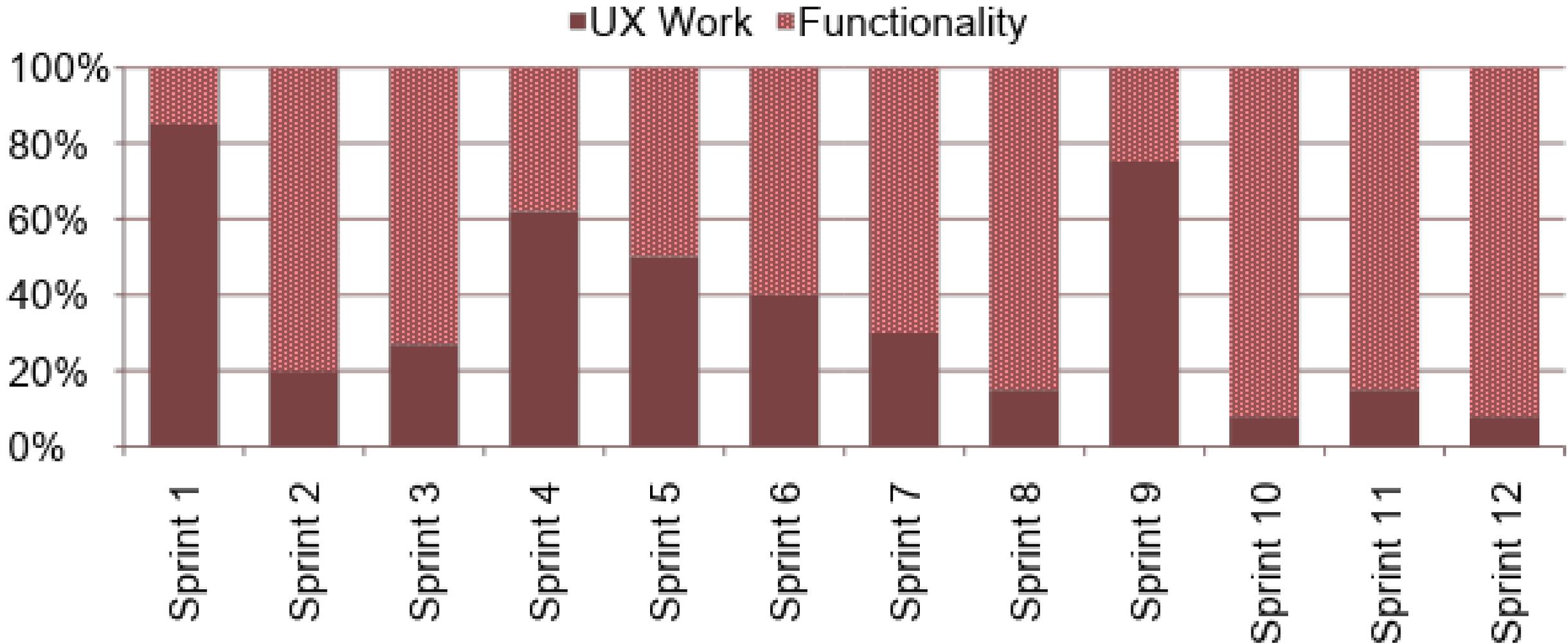
Every Sprint must still deliver at least some valuable business functionality

To prove that the user experience is as you expected

To prove to the customer that work they care about is taking place

As a basis for forecasting

# Emergent (Lean) UX



# How do You Manage UX Work in Scrum?



There are 3 approaches:

1. Separate backlogs for UX and Development
2. Single backlog with PBIs specific to UX or Development
3. Single backlog UX and Development in each PBI

# Patterns of Managing UX Work: Pros and Cons

## Separate Backlogs

*Mindset: UX work doesn't have to interact in the Scrum process*

### PROS/CONS?

- Clear separation of types of work
- May lead to silos in team, phased waterfall development, or worse: separate teams
- PO must order multiple backlogs and balance work between each, or worse: multiple "PO's" each ordering different backlogs

## One Product Backlog UX Work **AS** PBI's

*Mindset: UX work can be an Increment that gets inspected at the Sprint Review*

### PROS/CONS?

- Feels natural to most teams
- May lead to silos in team, mini-waterfall behavior
- Must manage dependencies between PBI's when discovery work is required first or validation work is required after delivering the Increment.
- All PBI's no longer equal, some produce learning, some produce "Done" Product Increments

## One Product Backlog UX Work **IN** PBI's

*Mindset: Only "Done" working Product is the Increment, and everything else is just work done to get there*

### PROS/CONS?

- Focus on value delivery
- UX work thought of as Refinement?
- Team swarming
- May lead to high WIP and work that spans multiple Sprints as teams do discovery-development-validation on each individual PBI

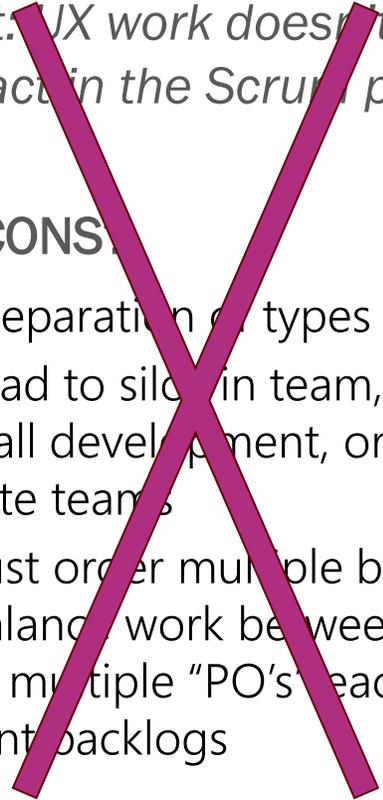
# Patterns of Managing UX Work: Our Recommendations

## Separate Backlogs

*Mindset: UX work doesn't have to interact in the Scrum process*

### PROS/CONS

- Clear separation of types of work
- May lead to silo in team, phased waterfall development, or worse: separate teams
- PO must order multiple backlogs and balance work between each, or worse: multiple "PO's" each ordering different backlogs

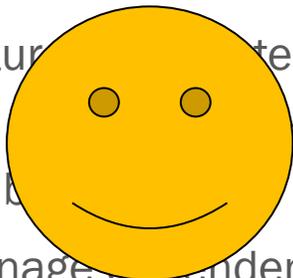


## One Product Backlog UX Work **AS** PBI's

*Mindset: UX work can be an Increment that gets inspected at the Sprint Review*

### PROS/CONS?

- Feels natural to teams
- May lead to waterfall thinking, mini-waterfall development
- Must manage dependencies between PBI's when discovery work is required first or validation work is required after delivering the Increment.
- All PBI's no longer equal, some produce learning, some produce "Done" Product Increments

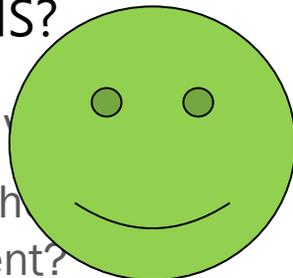


## One Product Backlog UX Work **IN** PBI's

*Mindset: Only "Done" working Product is the Increment, and everything else is just work done to get there*

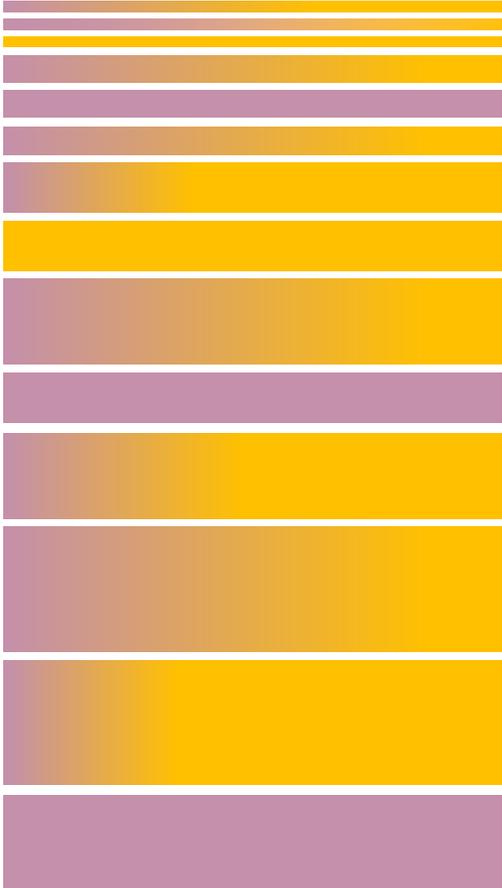
### PROS/CONS?

- Focus on validation
- UX work that is not "Done" Refinement?
- Team swarming
- May lead to high WIP and work that spans multiple Sprints as teams do discovery-development-validation on each individual PBI



# No One-Size-Fits-All Answer: Most UX Work IN PBI's, Some UX Work AS PBI's

UX  
DEVELOPMENT



**Case Study:**

**A Web Development Agency  
Powered by Scrum and  
Kanban**



# Case Study (July 2020 - Today)



# 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

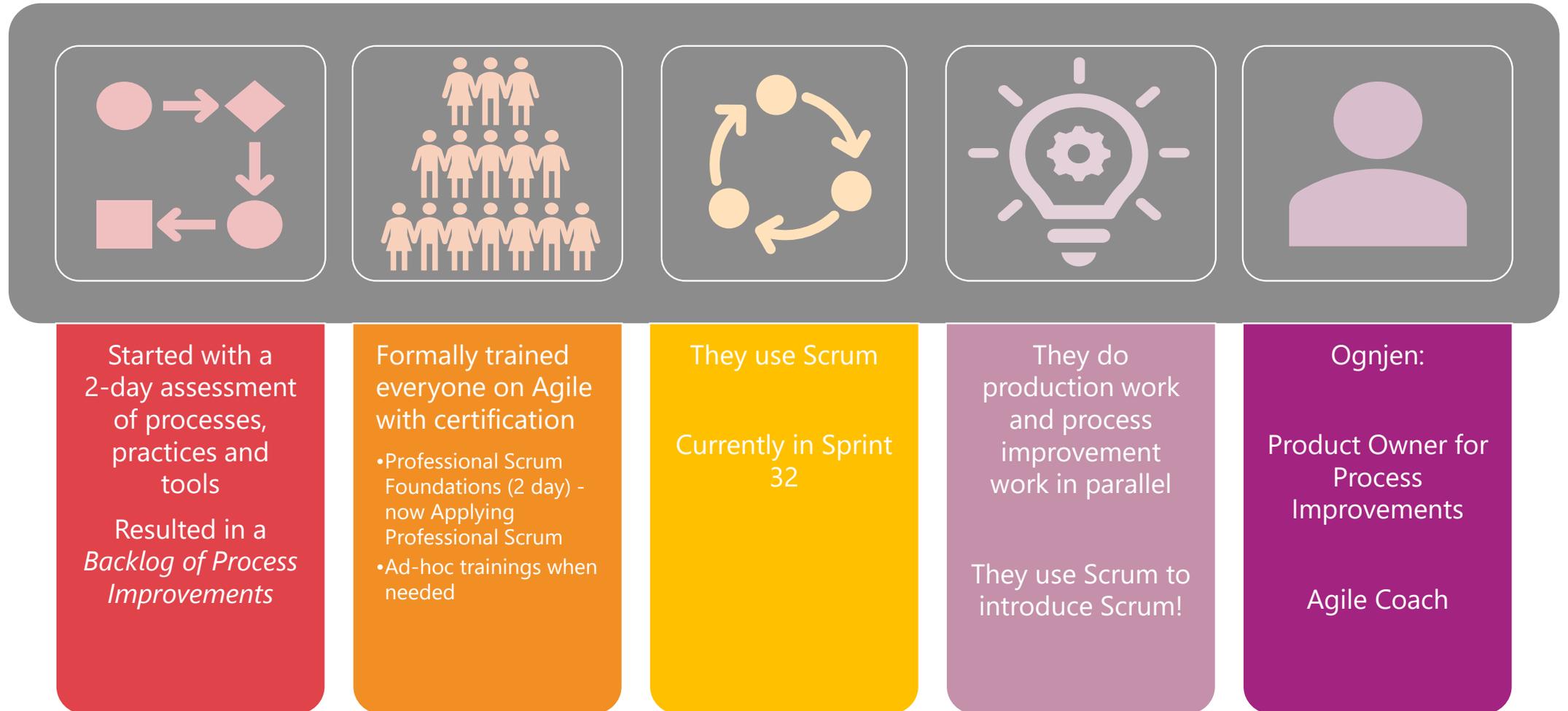
## Quality Issues

Bugs, unfinished features

## Unclear Accountabilities

Micromanagement enforced by one of the founders

# Our Approach



# Process Improvements (Part 1)

## Scrum

Unified hierarchical backlog for all projects managed by a single Product Owner  
Sprints with clearly defined and "*defended*" Sprint Backlogs

## Empowered Single Product Owner

Fully balances work on different ongoing projects  
Has full control over priorities, plan and roadmap

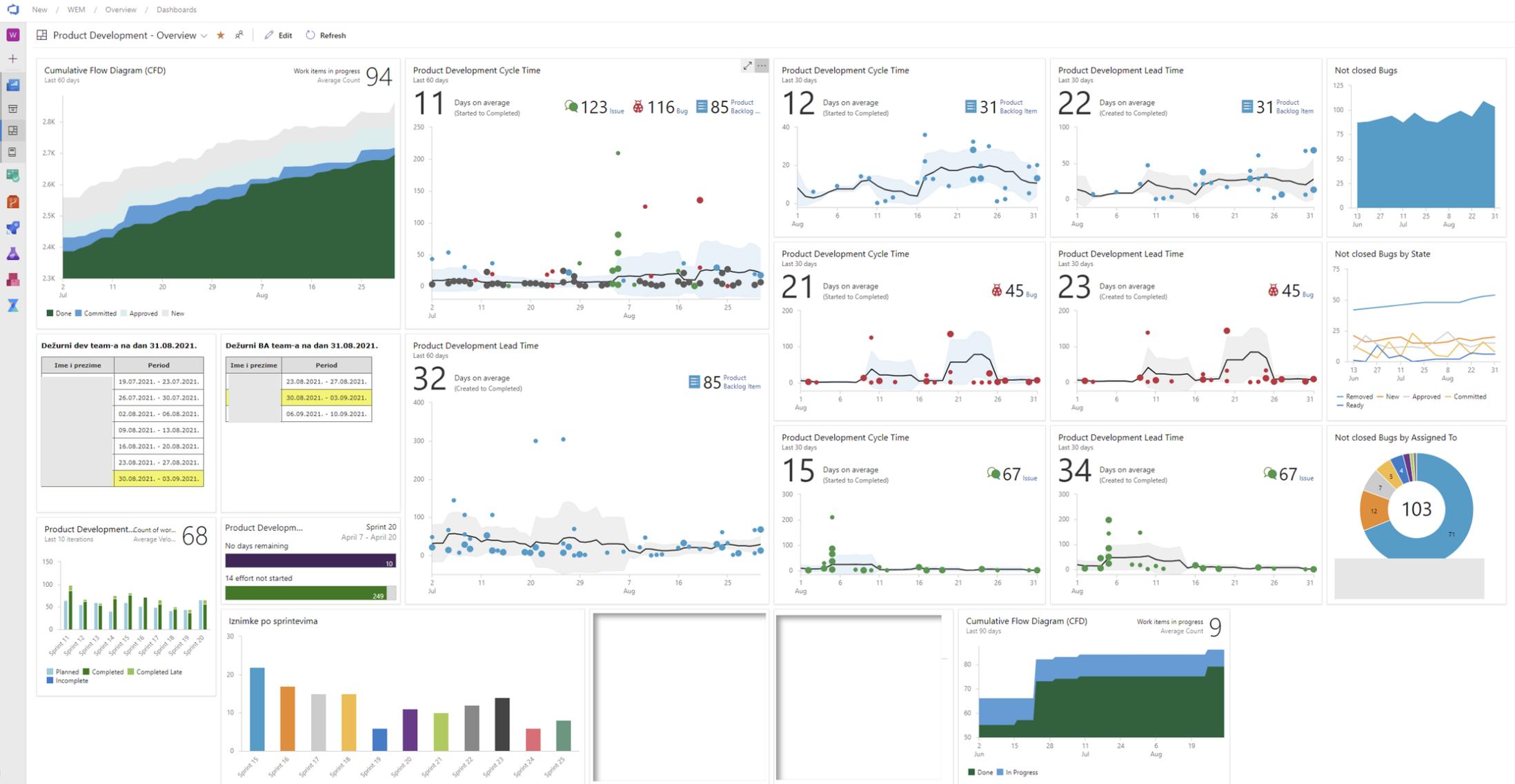
## Ability to Deliver Entire Modules Within Each Sprint

Main Page, Web Shop, Catalog, News, etc.

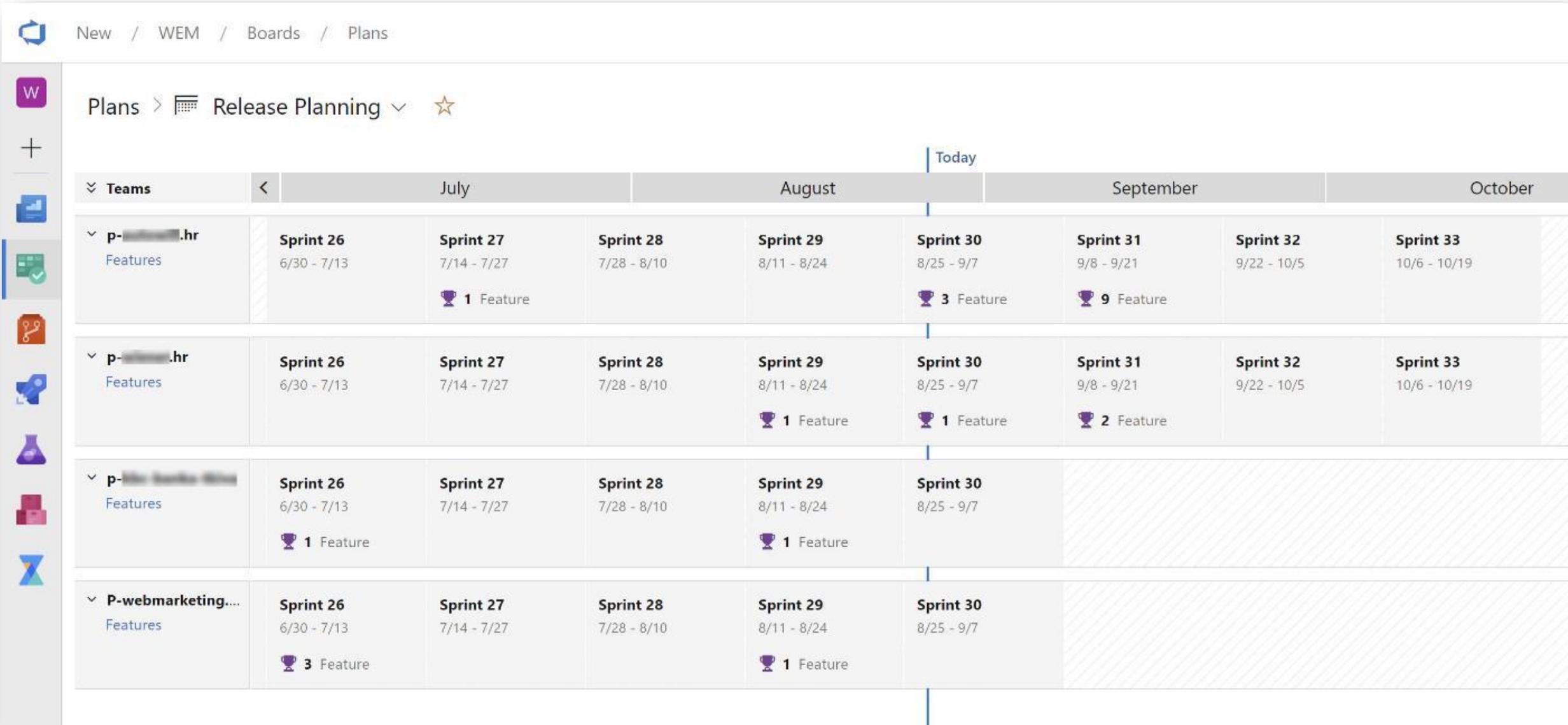
## Fewer/no Delays for Customer Approvals

UX is split in smaller units of work  
Design UX workshops followed by rapid development  
Design, development and delivery within the same sprint

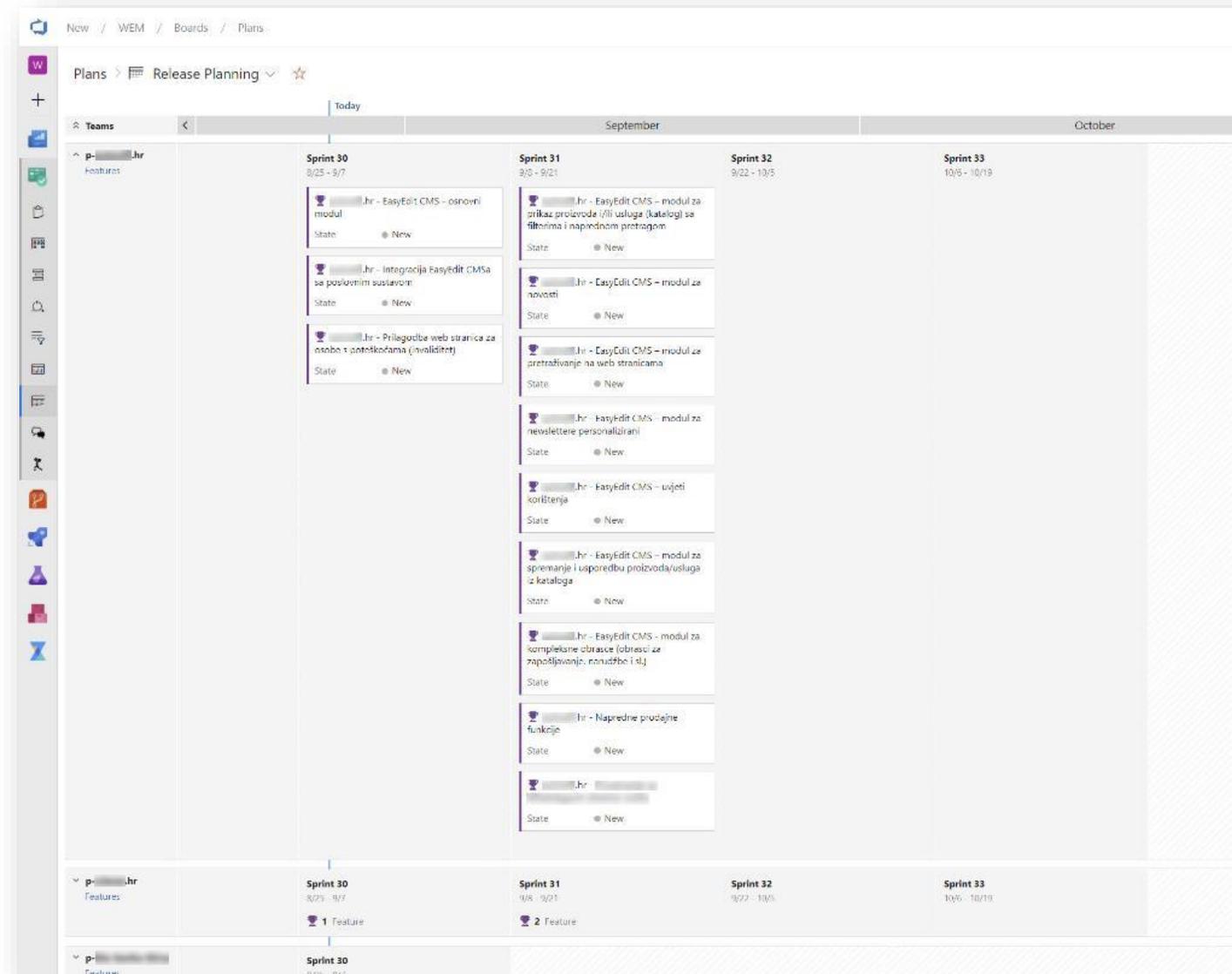
# Product Development Dashboard in Azure DevOps



# Delivery Plans – Condensed View - Release Plan



# Delivery Plans – Detailed View - Release Plan



# Process Improvements (Part 2)

## Clear Accountabilities

Self-managing Team

PO, SM, Devs (BAs, Designers and Devs)

Strict adherence to accountabilities

Less task switching, faster decision making

## Management has transparent insight into the status

Through Product Backlog, Delivery Plans, Dashboards

## End-to-end process supported by Azure DevOps

Kanban: Sales, Refinement and Support

Scrum: Design and Development

## Support process visually managed on a Kanban board

# Chained Processes

End-to end process modelled using Azure DevOps and CRM

Chained processes:

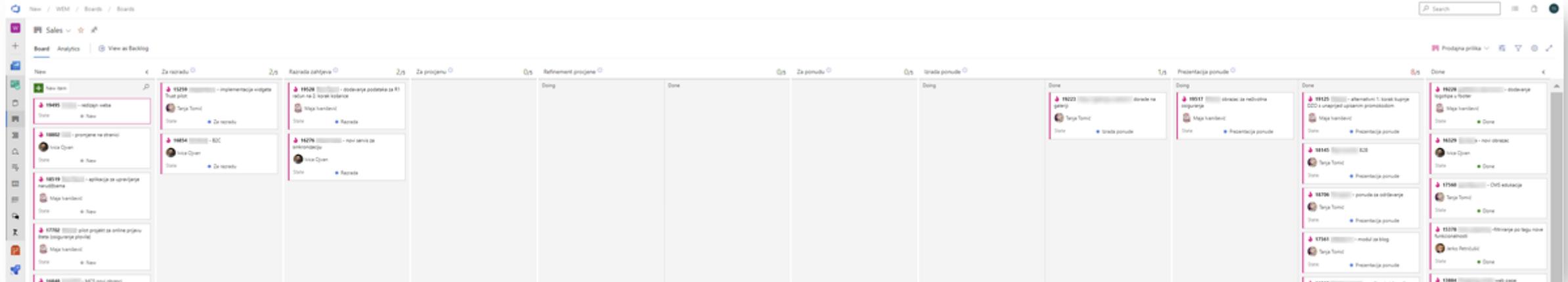


Sales process managed in Azure DevOps, driven by CRM

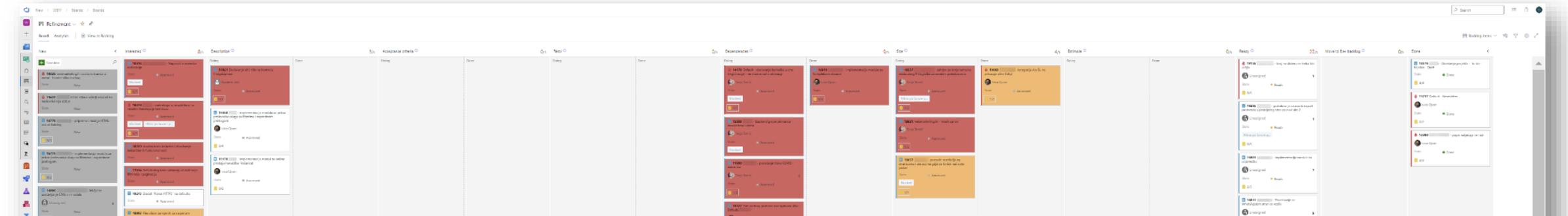
Work flows from one process to another

# Chained Processes (Contd.)

## Sales Kanban



## Refinement Kanban



<b>New</b>	<b>Interested</b>	<b>Description</b>	<b>Acceptance Criteria</b>	<b>Tests</b>	<b>Dependencies</b>	<b>Size</b>	<b>Estimate</b>	<b>Ready</b>	<b>Moved to Dev Backlog</b>	<b>Done</b>
------------	-------------------	--------------------	----------------------------	--------------	---------------------	-------------	-----------------	--------------	-----------------------------	-------------

# Quality Improvements

## Improved Requirements

Clear Definition of Ready (DoR) supported by a Kanban Board  
Explicit acceptance criteria for each PBI  
Requirement refinement process supported by Kanban  
Separate DoR for design and development

## Improved Product

Clear Definition of Done (DoD) for design and development  
Pull Requests with Peer Reviews and CI build

## Specified and Exploratory Testing

Test Cases for each acceptance criteria  
Test Cases, Test Plans, Test Configurations, Exploratory tests  
Testing tools create rich bugs (easy to reproduce, understand, and resolve)



# Effectiveness Improvements

## Minimized impact of support and incidents on ongoing development

Introduced BA and Dev "On Duty"

They deal with all production issues and interruptions

The rest of the team is focused on production work

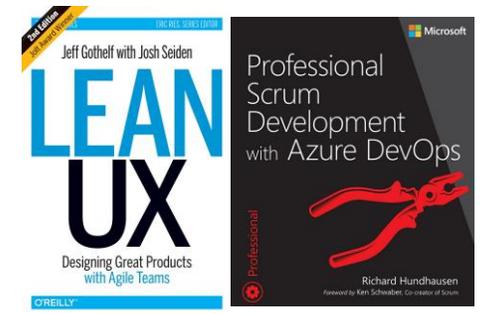
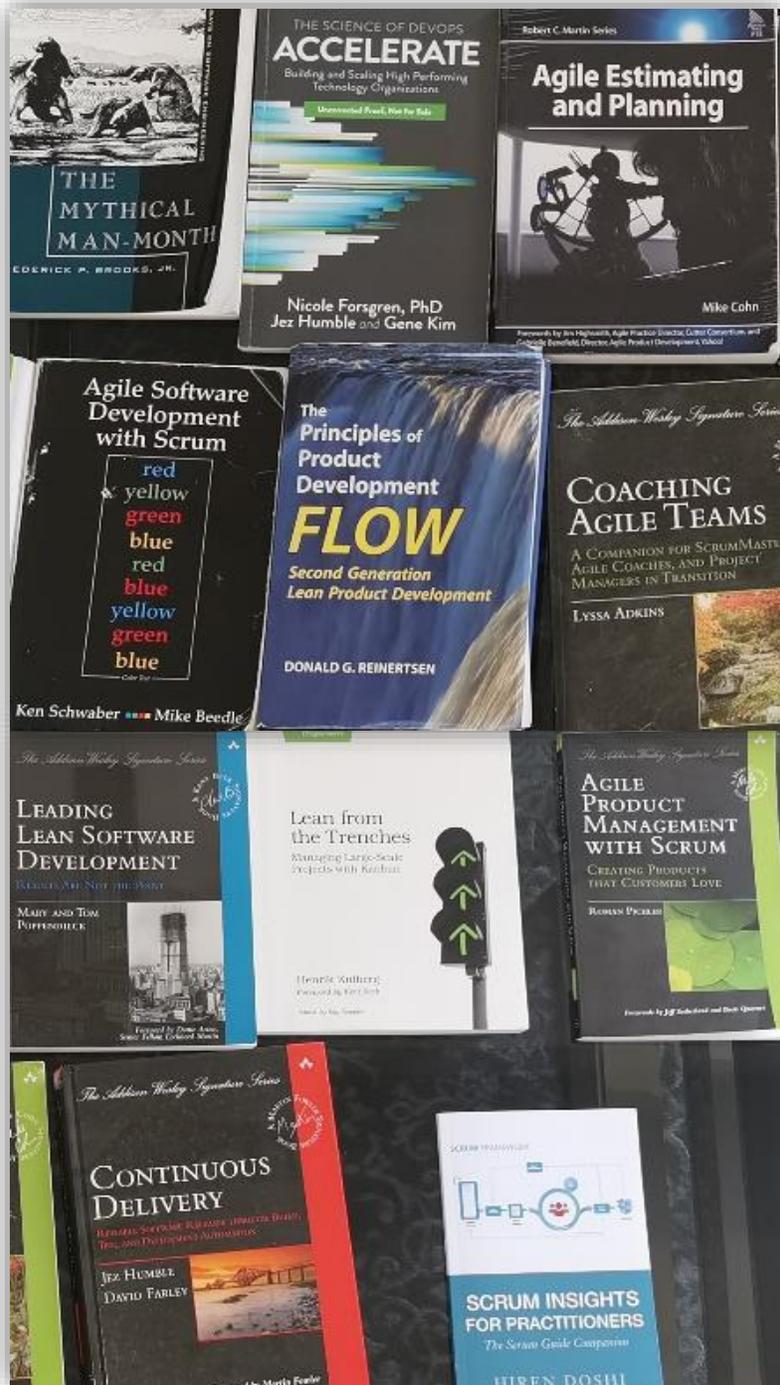
## Automated CI and CD pipelines

No more manual deliveries

All developers can deliver from Day 1

# Next Steps





# More Info

- [Lean UX: Designing Great Products with Agile Teams](#)
- [Professional Scrum Development with Azure DevOps](#)
- Agilists' Suggested Reading for Scrum Practitioners
  - <https://aglst.com/ScrumResourcesAndBooks>
- Agilists' Selection of Books for Agile Practitioners
  - <https://aglst.com/APS-ReadingList>
  - <https://aglst.com/APS-SD-ReadingList>
- The Scrum Guide 2020
  - <https://scrumguides.org/index.html>
- Vodič kroz Scrum 2020 – Croatian Translation
  - <https://aglst.com/ScrumGuide2020-HR-PPTX>
- Kanban Guide for Scrum Teams
  - <https://www.scrum.org/resources/kanban-guide-scrum-teams>

# Agile in a Web Development Agency

UX and Development work intertwined

We prefer: One Product Backlog with UX Work In PBI's

Case Study: Web Marketing

## Questions?

**Ana Roje Ivančić**, [arojeivancic@agilist.hr](mailto:arojeivancic@agilist.hr)

**Ognjen Bajić**, [obajic@agilist.hr](mailto:obajic@agilist.hr)

Zagreb, Croatia

[www.agilitistit.com](http://www.agilitistit.com)

[trainings@agilist.hr](mailto:trainings@agilist.hr)



**Get 20% discount  
for conference  
attendees**

**Contact us at**

**[trainings@agilist.hr](mailto:trainings@agilist.hr)**

