DevOps Skills for Developers with Visual Studio & TFS 2017

DEVOPS: FOCUSING ON WHAT'S IMPORTANT

Benjamin Day

TRAINER | COACH | DEVELOPER

@benday   www.benday.com

Course Summary

Mix of theoretical & practical

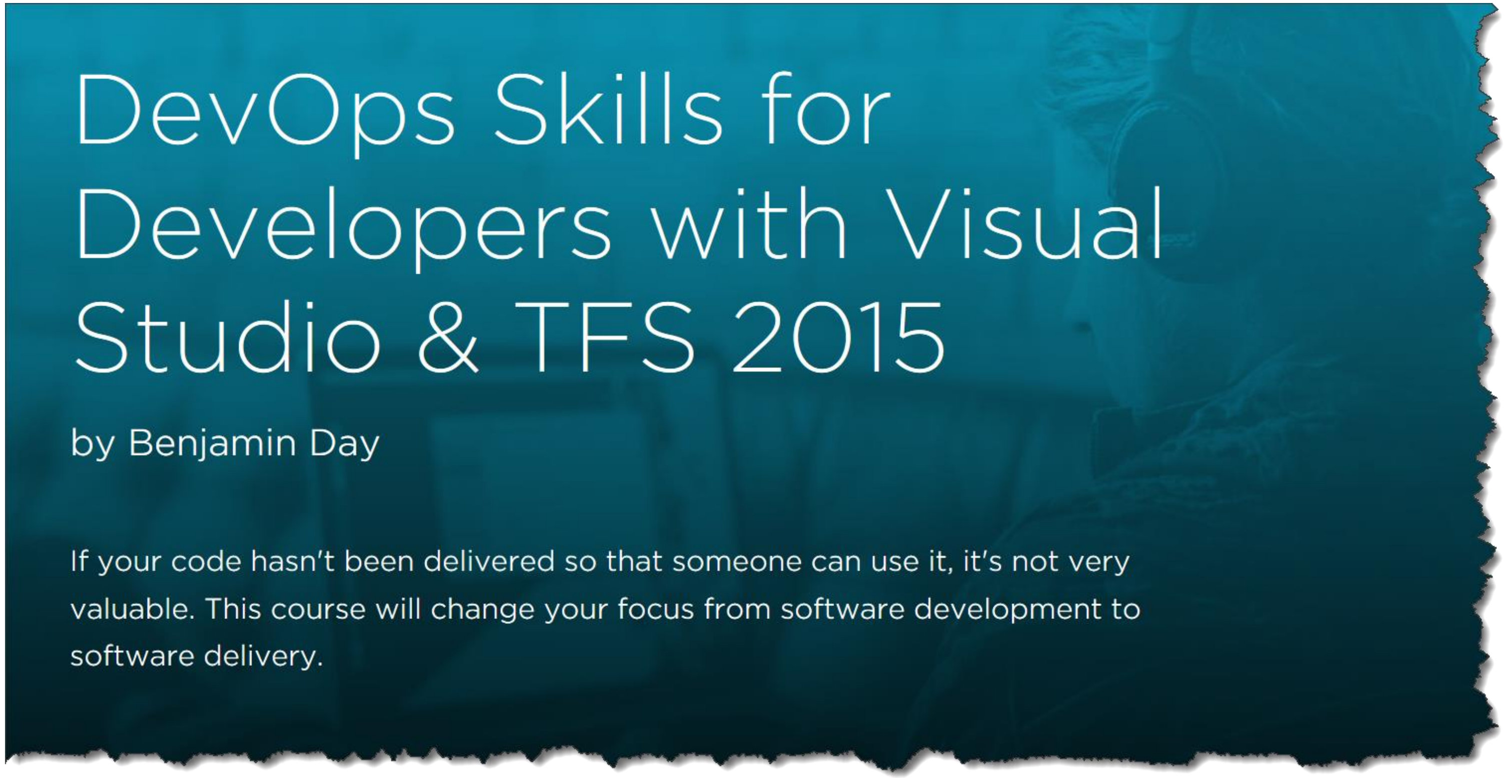
Team Foundation Server 2017

Visual Studio 2017

Primary focus is “on-premise” TFS

Almost everything should work on Visual Studio Team Services (VSTS)

How Does This Compare to the Previous Course?



DevOps Skills for Developers with Visual Studio & TFS 2015

by Benjamin Day

If your code hasn't been delivered so that someone can use it, it's not very valuable. This course will change your focus from software development to software delivery.

https://www.pluralsight.com/courses/devops-skills-developers-visual-studio-tfs-2015

Module Summary

What’s, why’s, & where’s of DevOps

What’s changed in the Microsoft DevOps world for the 2017 release?

Next up:
The what’s, why’s, and where’s of DevOps

What is DevOps?
Why do I care?
Where does it fit?

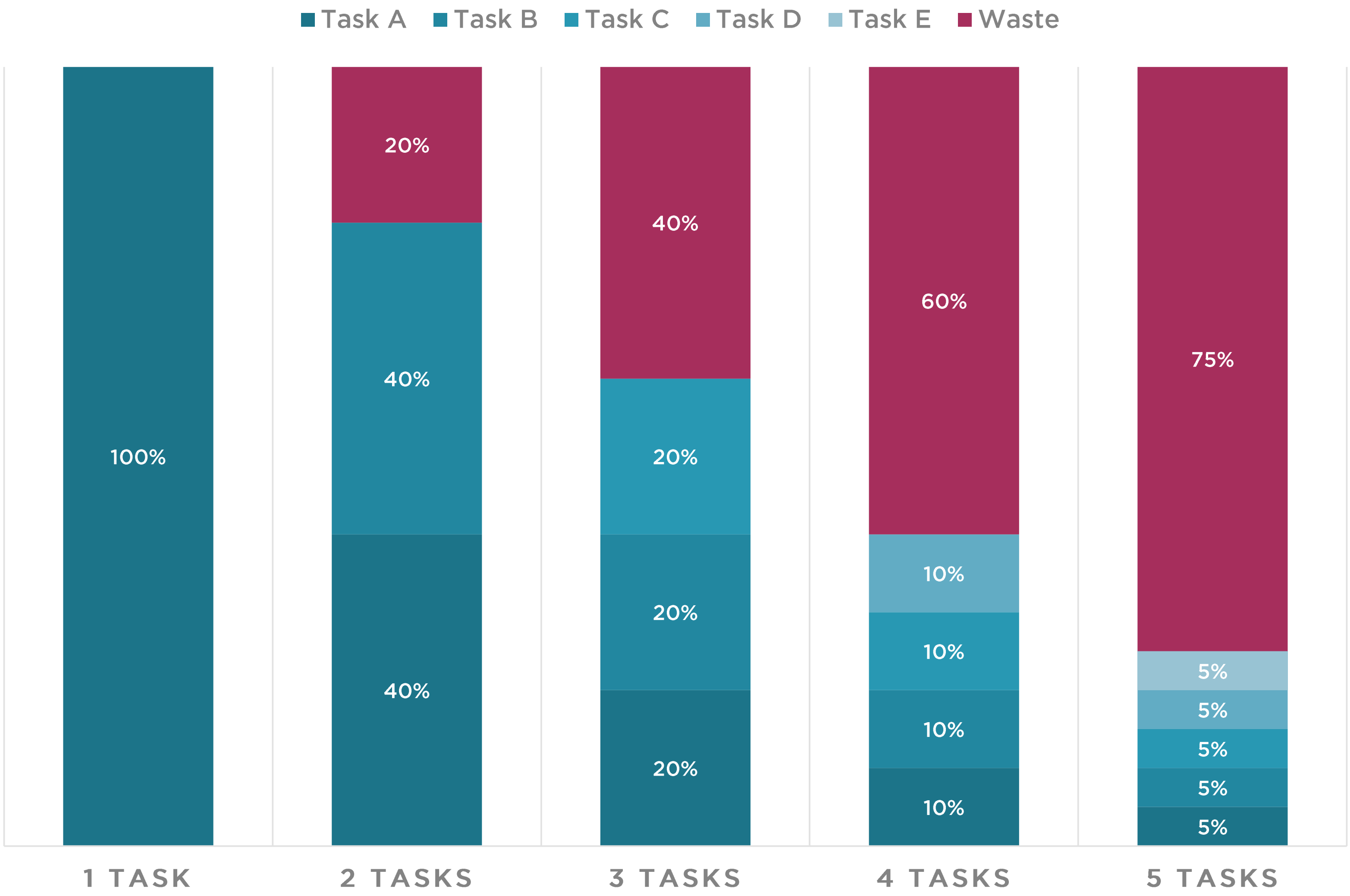
DevOps is a mindset plus a set of practices that focuses on automation.

Why?

Deliver faster & more often with less work.

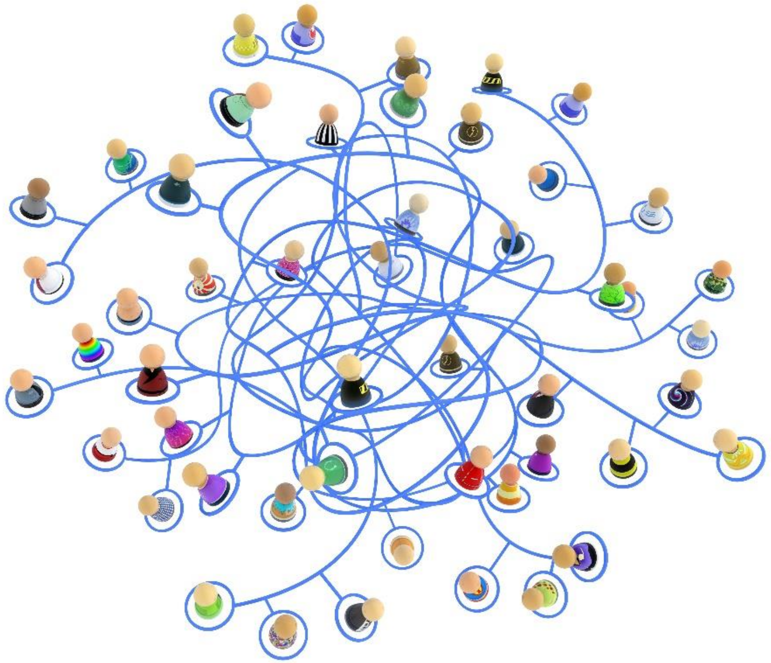
Productivity vs. Waste

0% waste



75% waste

"Quality Software Management: Vol. 1 System Thinking“, Gerald Weinberg (1992)

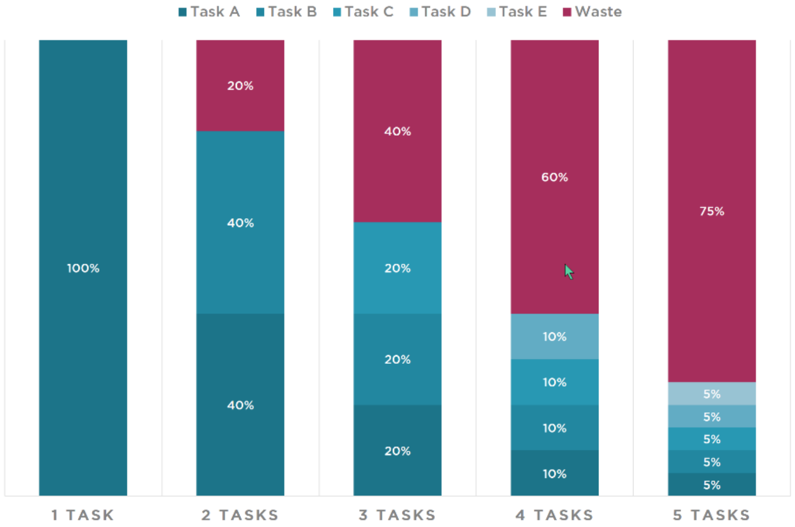


Multitasking is horrible

Multitasking sneaks up on you

Multitasking is expensive

Multitasking Is Expensive



5 tasks → 75% waste

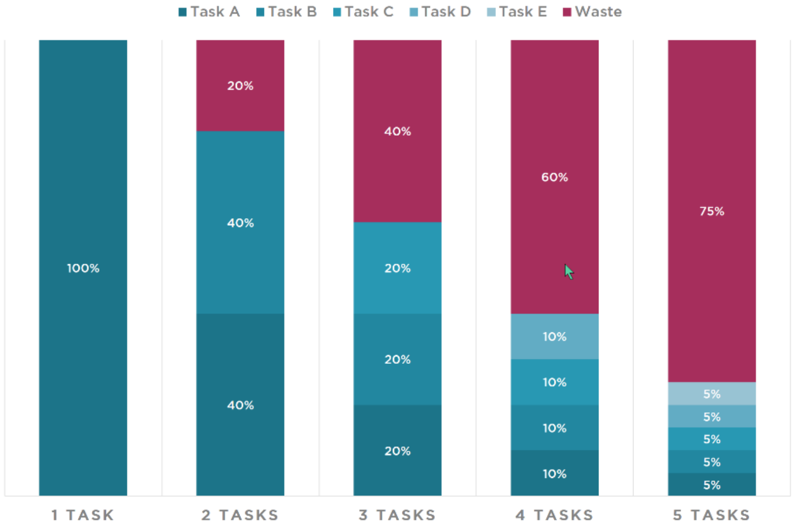
$100,000 → $25,000

40 hour week → 10 hours

Monday – Friday →

- Monday gone
- Tuesday gone
- Wednesday gone
- Thursday mostly gone
- Productive on Friday

Multitasking Sneaks up on You



You're assigned one task...

...and a couple of bugs

...and something is weird in production

...and patches are needed on the servers

...and fix a problem in QA

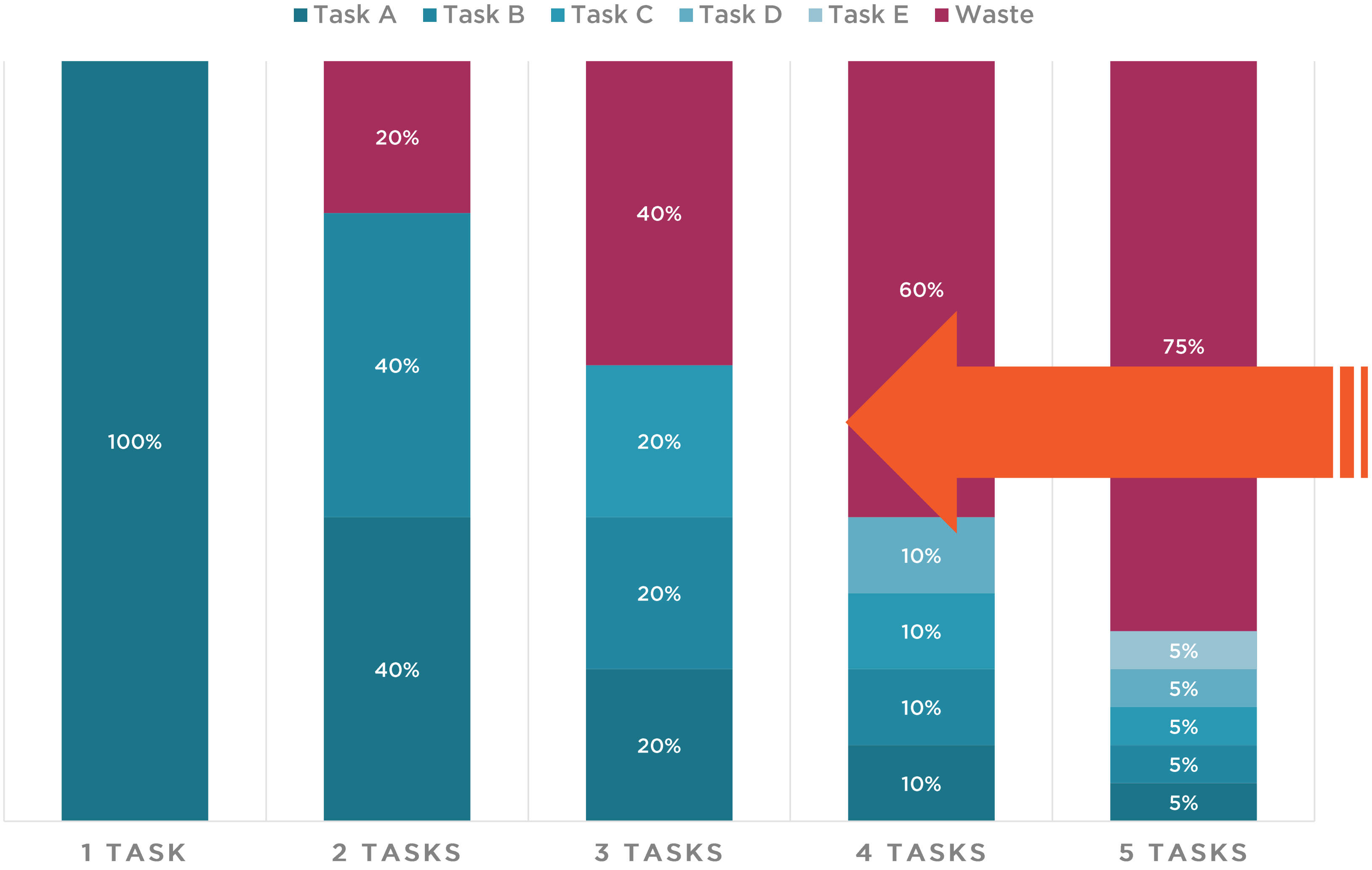
...and deploy a release

Reality Check:
You probably won’t get down to just one thing.

Work can hide in plain sight.

DevOps =
Eliminate Distractions

Productivity vs. Waste



0% waste

75% waste

"Quality Software Management: Vol. 1 System Thinking“, Gerald Weinberg (1992)

If it’s repetitive, automate it.
If it’s painful, automate it.

Software Development DevOps Opportunities

Scope	Activity	Change Frequency	DevOps Action
Project	Requirements	Avoid new distractions & multitasking	Track it
Sprint	Requirements		Track it
Sprint	Develop the software	Plug in to existing DevOps flow	Automated builds
Sprint	Test the software		Automated tests
Sprint / Project	Integration & regression testing	Already been DevOps'd	Automate
Sprint / Project	Deploy to testing environment & re-test		Automate
Sprint / Project	Deploy to production		Automate

Huge % of Time → Tedious & Disruptive Work

Integrating

AUTOMATED BUILD

Testing

AUTOMATED TEST

Deploying

AUTOMATED DEPLOY

Big Benefits

Automated Build, Test, & Deploy

Release more often

- Not scary, painful, disruptive

Faster bug fixes

Smaller, more frequent releases

- Easier to change
- Easier to test
- Easier to get feedback

Respond faster to feedback

DevOps decreases the time from ‘Feedback’ to ‘Fix’

Decreases the number of things you need to worry about.

Focus on your real job.

Where does DevOps fit?

Software Development Industry
vs.
Software Delivery Industry

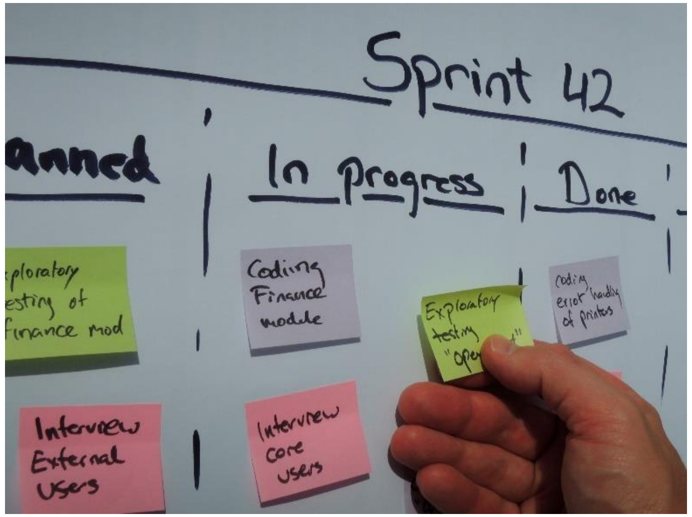
Software that’s written is interesting.

Software that’s *delivered* is valuable.

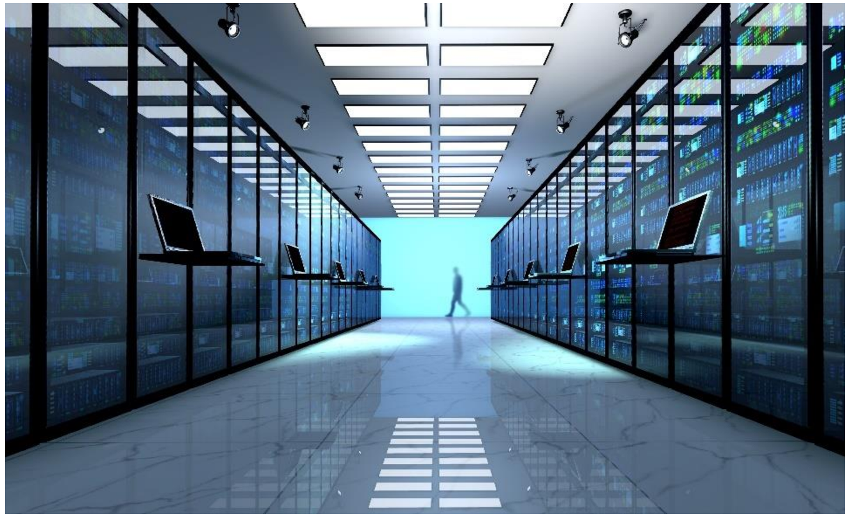
Delivered = $$$

Scrum & Agile

Almost Delivered

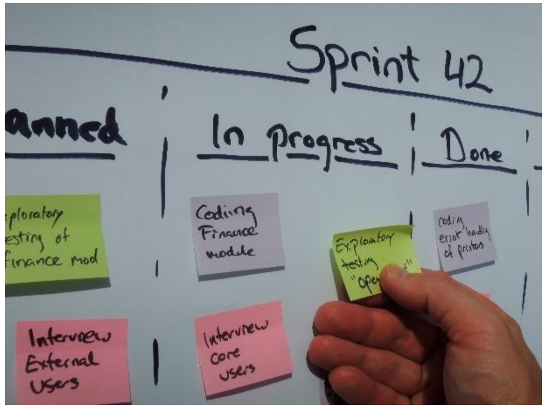


Agile / Scrum:
Good at getting ready to deliver



IT Operations:
Responsible for deploying
(aka. actually "delivering")

Almost Delivered



Agile / Scrum:
Good at getting ready to deliver

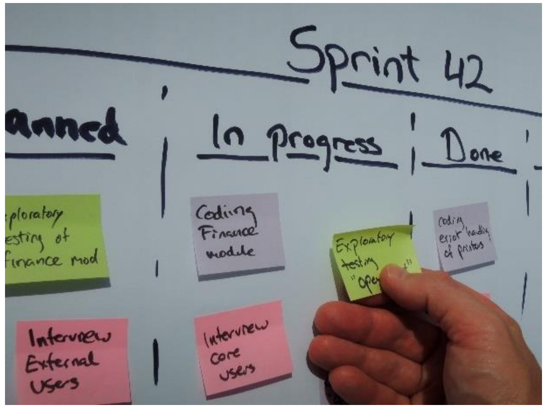
A Giant Brick Wall

“Kick it over the wall to IT.”

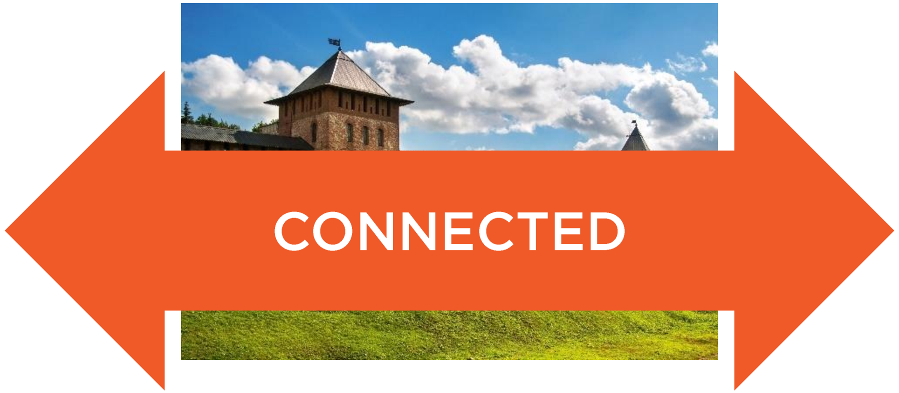
IT Operations:
Responsible for Deploying

(aka. actually “delivering”)

Dev + Ops = DevOps



Agile / Scrum:
Good at getting ready to deliver



A Giant Brick Wall

"Kick it over the wall to IT."

IT Operations:
Responsible for deploying

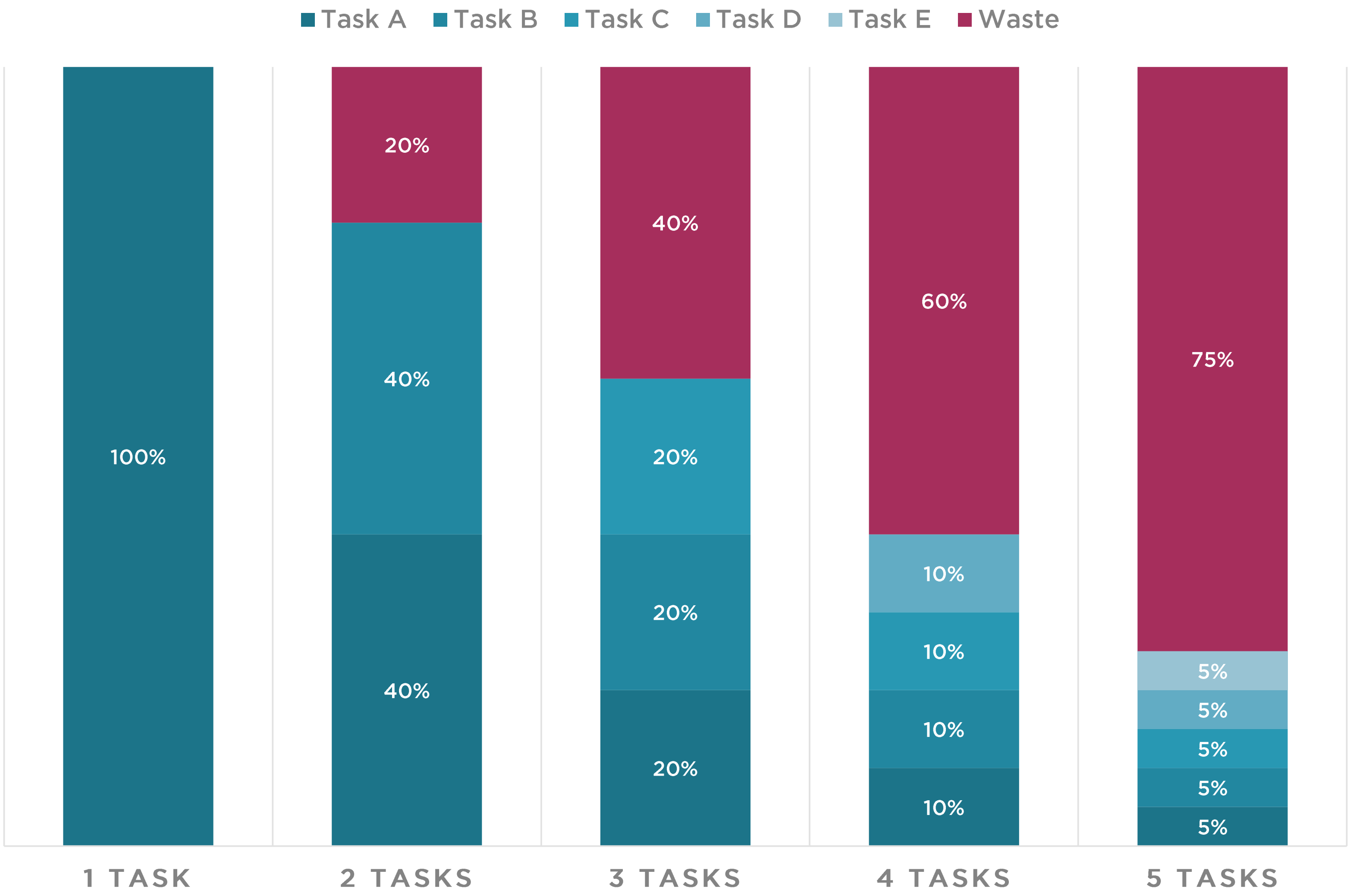
(aka. actually "delivering")

Recap:

DevOps is a mindset plus a set of practices that focuses on automation.

Use DevOps to Reduce Waste

0% waste



75% waste

"Quality Software Management: Vol. 1 System Thinking“, Gerald Weinberg (1992)

Next up:
What’s changed for DevOps with TFS2017 & VS2017?

DevOps: What’s changed in TFS2017 & VS2017?

Not much...

…and also a whole lot.

The world has changed and Team Foundation Server 2017 had to change with it.

The Microsoft Development World Before Now

Visual Studio

.NET Framework

ASP.NET

SQL Server

ADO.NET & Entity Framework

Windows Server

Internet Information Server (IIS)

Deploy an Application in Microsoft World

Servers have names

Web server

Database server

“Deploy to IIS”

“Deploy to SQL Server”

All Windows, all the time.

More and More in Microsoft Land…

Decreased supremacy of Windows

Multi-OS company

Cloud computing

Containers

.NET Core & ASP.NET Core

Microsoft Is Cross-platform

.NET Core, ASP.NET Core, Entity Framework Core

Run on Windows, Linux, MacOS

SQL Server *for Linux*

All this stuff actually works

The world is changing.

Coming Soon!

The end of the corporate data center

The end of servers with names

The end of patching servers

- OS updates
- Deploying application updates

Need a new version of your app?

- Stop the old server
- Start a new one
- Deploy on to the new server

DevOps with TFS2017: More cloud, more cross-platform

Cloud Computing

Microsoft Azure

Infrastructure as Code (IaC)

Azure Resource Manager (ARM)

Containers

Docker and/or Windows Containers

Cross-platform

.NET Core on Windows, Mac, Linux

DevOps isn’t just about automating deployment.

Cloud-based DevOps: Automated provisioning of execution environment followed by application deployment

DevOps with TFS 2017

Version Control

- Code, Tests
- Deployment & Configurations
- Execution environment

Automated Build

Automated Test

Automated Deploy to Azure and/or Containers

- Provision “servers”
- Deploy the application

Big Philosophy Change:
Don’t update existing servers, create a new ones

Team Foundation Server 2017
has a lot of new stuff to support this
new way of thinking & working.

But what about specific features?

Top 4 New Features in TFS2017

Package Management

- Private NuGet Server

Git Improvements

- Pull Requests & Branching Web UI

QA Testing Improvements

- Microsoft Test Manager features are nearly all web-based now

Build & Release Improvements

- Task Groups
- Variable Groups
- Improved Azure & Docker support

Watch for these things throughout the rest of the course.

Summary

What's, why's, where's of DevOps

What's changed in the Microsoft DevOps world for the 2017 release?

Next up:
Version Control