



Visual Studio **LIVE!** | San Diego
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Get Good at DevOps: Feature Flag Deployments with ASP.NET, WebAPI, & JavaScript

Benjamin Day
www.benday.com
[@benday](#)

 Code Again for
the First Time!
[@benday](#) | www.benday.com

 **MVP**
Microsoft®
Most Valuable
Professional

 Visual Studio **25**
YEARS OF CODING INNOVATION

Benjamin Day

Brookline, MA

Consultant & Trainer

Scrum, DevOps,
Team Foundation Server,
Software Architecture & Testing

Microsoft MVP

Pluralsight Author

Scrum.org Trainer

[@benday](#)



Benjamin Day Consulting



PLURALSIGHT



Professional
Scrum Trainer
Scrum.org



[@benday](#) | www.benday.com



PLURALSIGHT

Pluralsight.com

Online courses

DevOps with TFS2017

Scrum Master Skills

DevOps Skills for Developers with Visual Studio & TFS 2015	by Benjamin Day	Intermediate	7h 7m	19 Sep 2016
Scrum Master Skills	by Benjamin Day	Advanced	4h 7m	06 Aug 2015
Load Testing with Visual Studio 2013	by Benjamin Day	Intermediate	3h 7m	05 Dec 2014
Real World Scrum With Team Foundation Server 2013	by Benjamin Day	Intermediate	5h 2m	10 Nov 2014
Load Testing with Visual Studio 2012	by Benjamin Day	Intermediate	4h 3m	26 Aug 2013
ALM for Developers with Visual Studio 2012	by Benjamin Day	Intermediate	4h 31m	23 Sep 2012
ALM with TFS 2012 Fundamentals	by Benjamin Day	Intermediate	5h 34m	04 Sep 2012



@benday | www.benday.com

Getting Started with Visual Studio Team Services (VSTS)



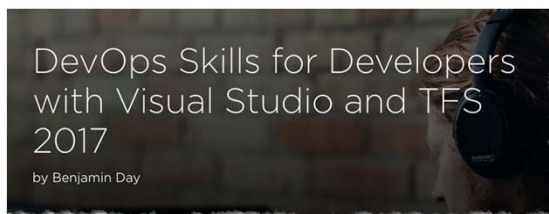
PLURALSIGHT

Architecting an ASP.NET Core MVC Application for Unit Testability

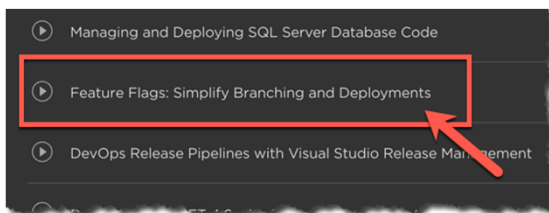
Coming in ~November 2018



Want to see this talk and code again?



DevOps Skills for Developers
with Visual Studio and TFS2017



Feature Flags: Simplify
Branching and Deployments

On with the show.

Software development is hard.

Software delivery is hard.

How do you get a feature into production?

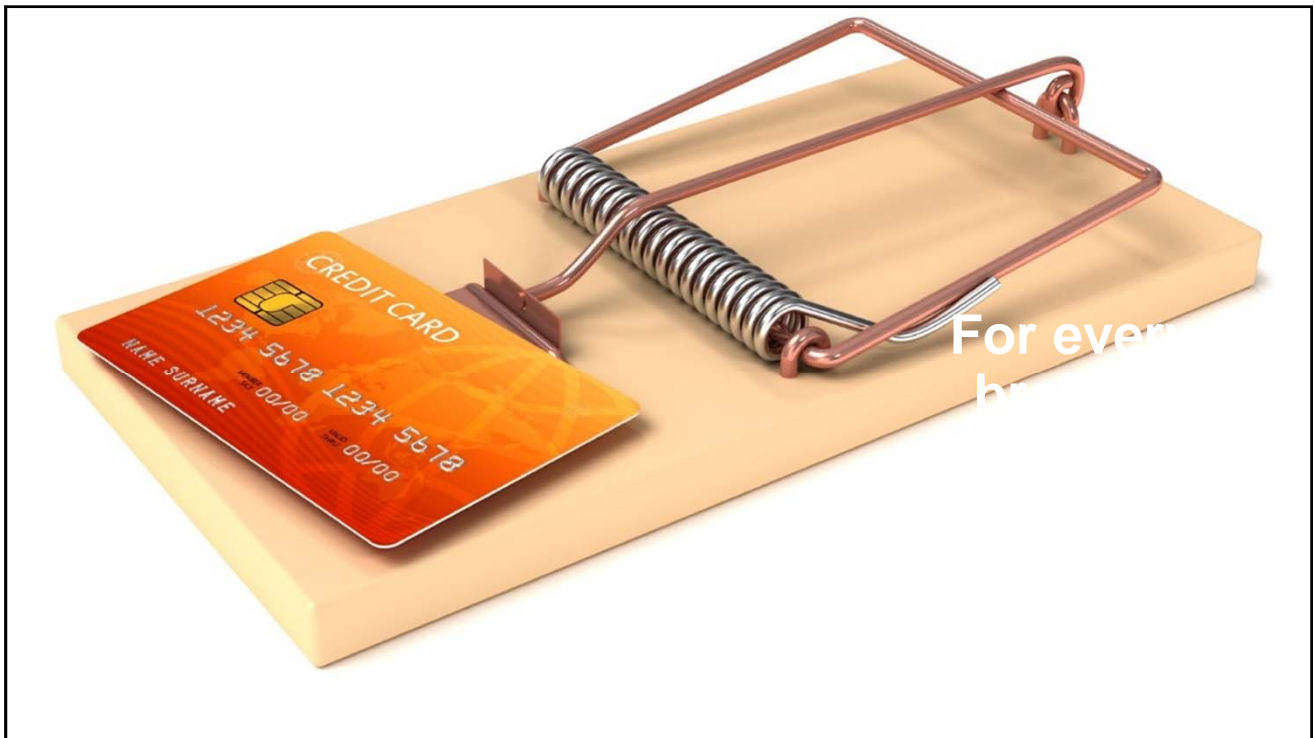
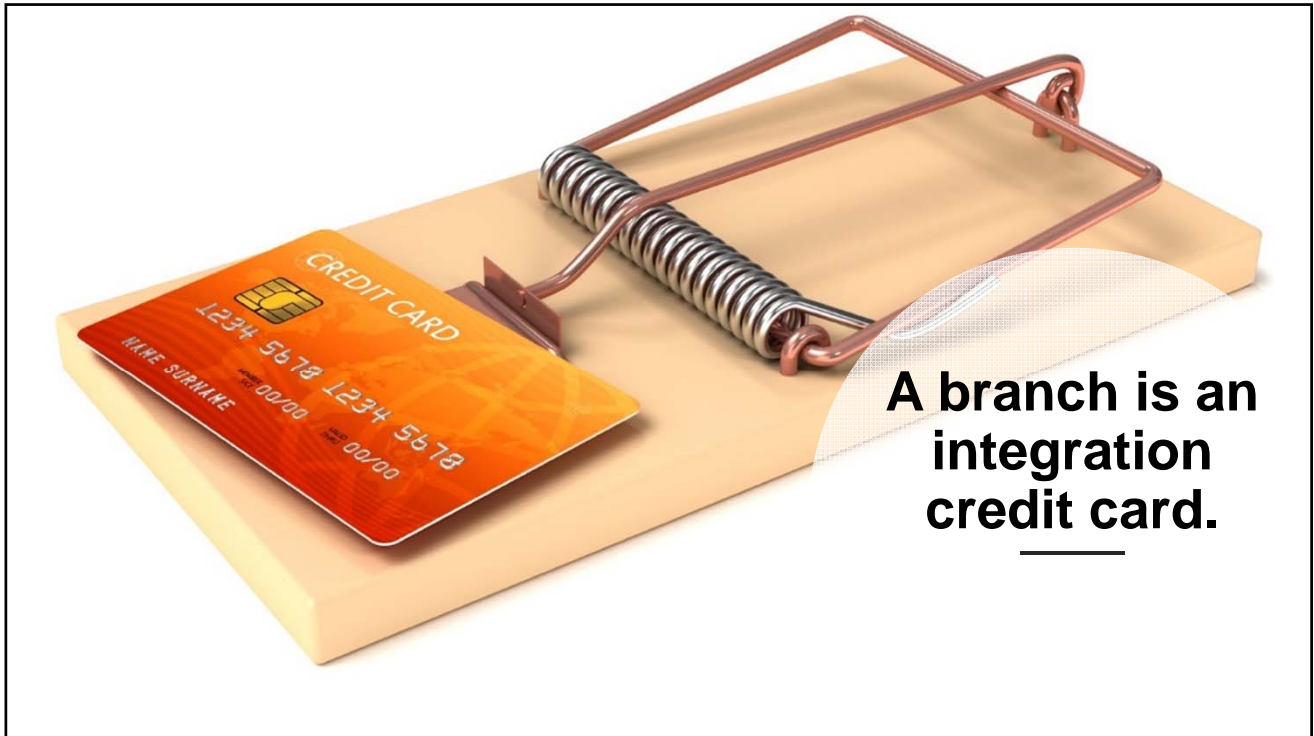
Branching & Merging

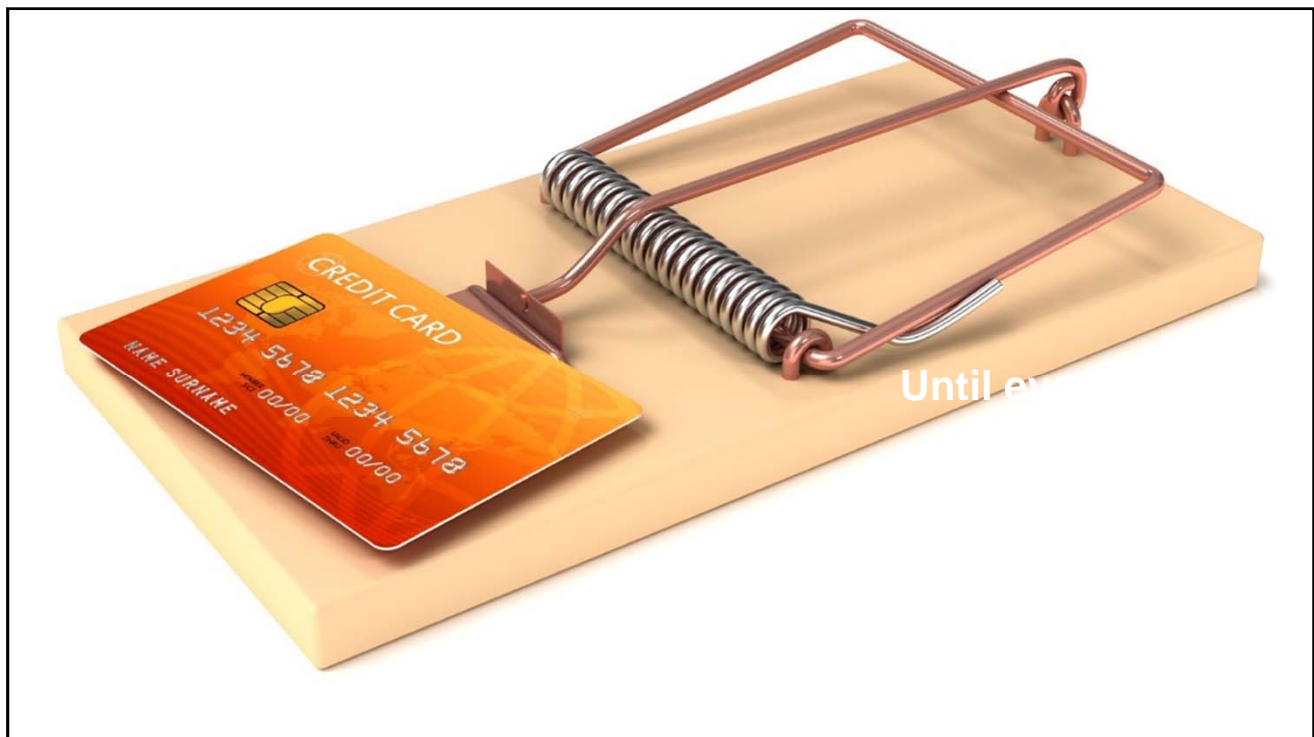
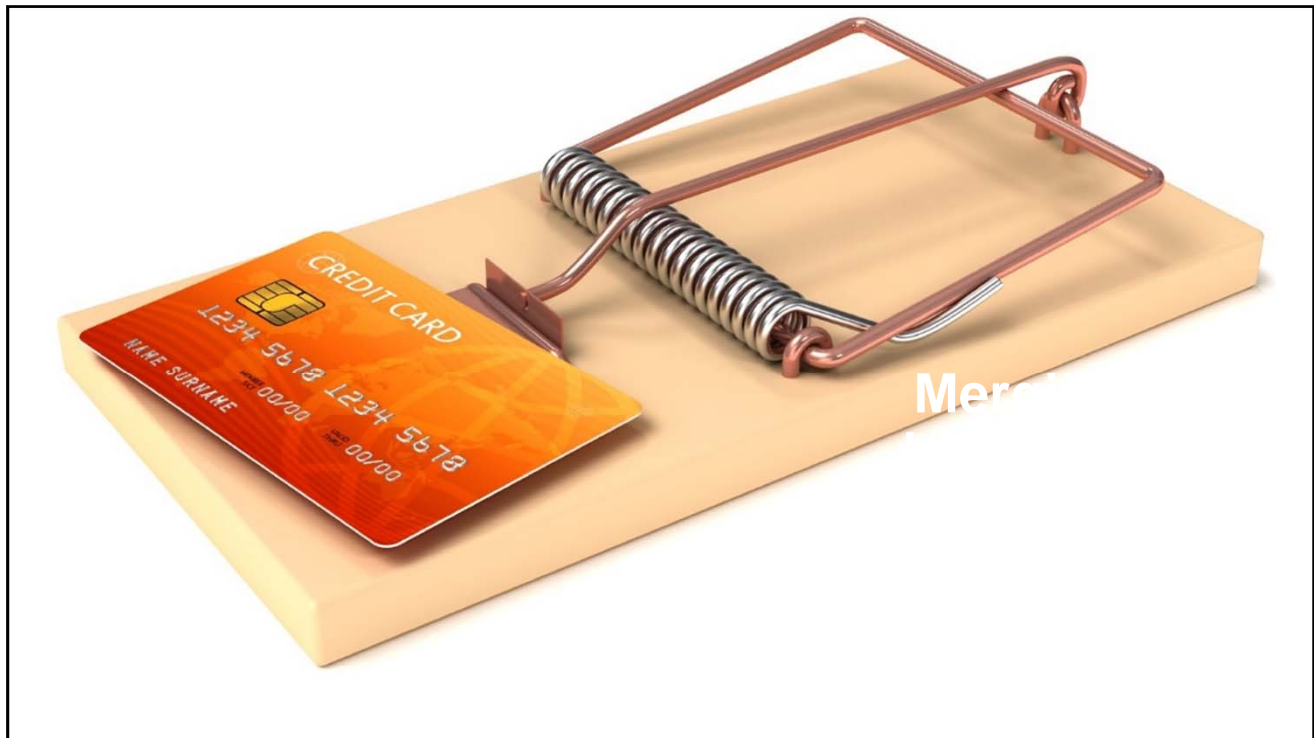
Branching lets you work on similar, related code in isolation at the same time.



**“So, I can go nuts
and create
250 zillion
branches
and it’s a good
idea?”**

No.





Keep it simple.

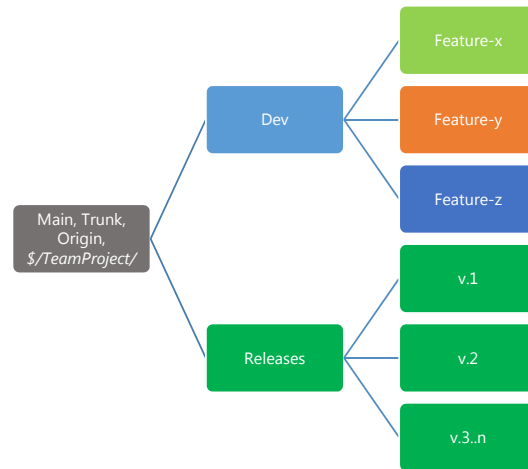
Best Practice: Don't Branch *

* - unless you absolutely have to

Integrate often.

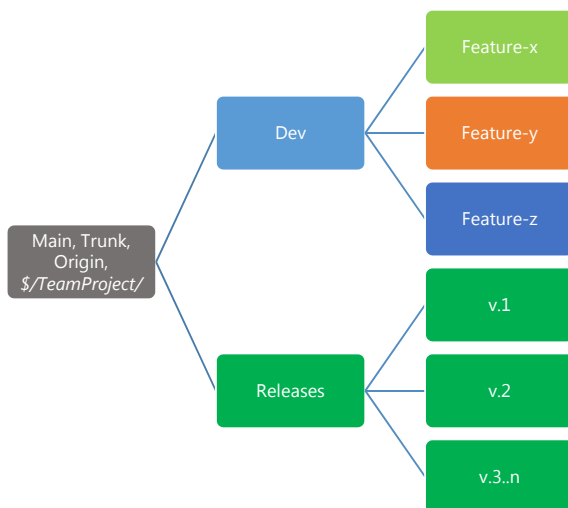
**The smaller the integration,
the easier it is.**

Typical Branching Structure



BDC @benday | www.benday.com

Typical Branching Structure



Main

- Beginning of everything
- Ultimate integration point

Dev

- Integration point for a release

Feature-*

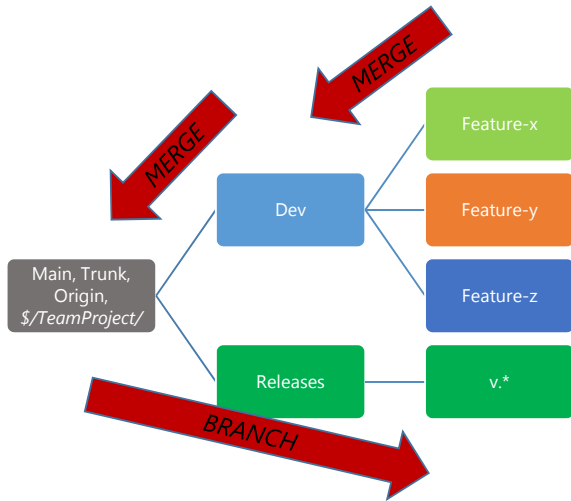
- Features are developed here

Releases (v.*)

- Snapshot of what goes to production
- Source of hot fixes

BDC @benday | www.benday.com

Typical Release Flow



Merge Features to Dev

Verify / fix whatever is in Dev

Merge to Main

Create new Release branch

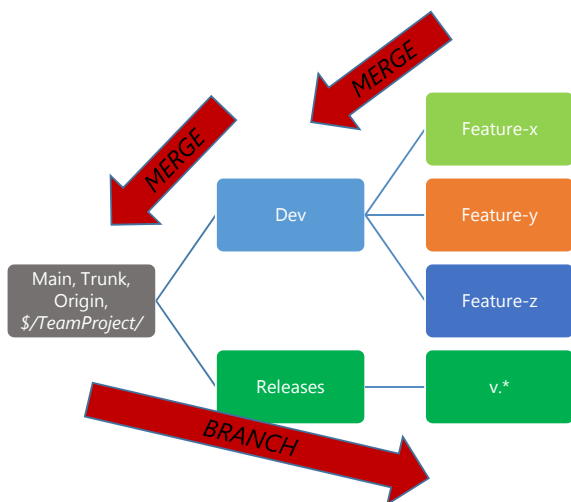
- V.1123414
- 2016.03.15

Deploy to production servers



@benday | www.benday.com

Typical Release Flow



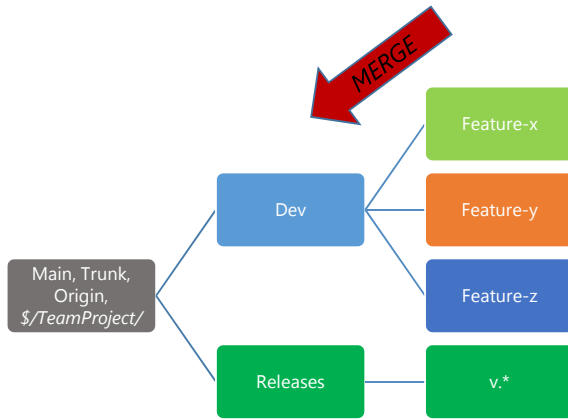
Whatever gets merged to Main gets released

Code-based release model



@benday | www.benday.com

Typical Release Flow: Bad News



Merge Feature X
- Easy

Commit the merge

Merge Feature Y
- Ok

Commit the merge

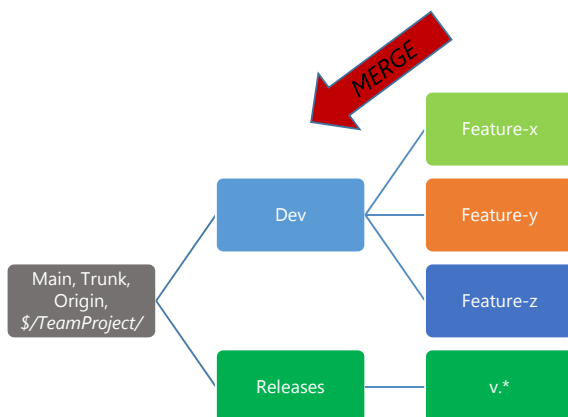
Merge Feature Z
- HARD
- REALLY REALLY HARD

Turns out that Feature X isn't done

Major surgery to get this undone

BDC @benday | www.benday.com

Typical Release Flow: Bad News



Feature branch integrations
are not going well

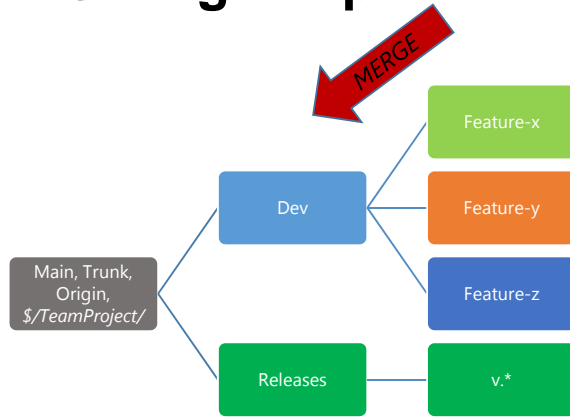
Getting merged code to
"play nice" means manually
un-doing work that's already
been committed

Cherry picking

DELAYS

BDC @benday | www.benday.com

Typical Release Flow: Bad News, Change of plans



Merge Features to Dev

Commit the merge

Boss wants only Feature X and Z to go live

Merge Feature X and Z to Main
- Painful, annoying, & tedious

Commit the merge



@benday | www.benday.com

**Releases based on source control
can be a real drag.**

Releases based on Source Control

Whatever is in the branch goes to production

Whatever is in production is live

Pros:

- Simple to understand

Cons

- Coordination can be a real headache



@benday | www.benday.com

Feature flags

Feature Flags

"Feature toggles"

Deploy whatever doesn't cause problems

Configuration-based releases

Turn on whatever is ready

Define configuration values for features

- This is a feature flag
- On / Off

Much less emphasis on branching & merging

Wrap feature code inside of checks for feature flag

- If it's on, it runs

Easier to schedule

- The features are already deployed
- Turn them on when you're ready



@benday | www.benday.com

Feature Flag: Benefits

New levels of flexibility

A/B testing

- Which version do people prefer?

Deploy multiple versions of the same feature side-by-side

- New user interfaces and old user interfaces at same time

Rollback

- Not working?
- Performance problems?

Private betas

- New features for only certain users

Great for Scrum & DevOps



@benday | www.benday.com

Feature Flag: Drawbacks

Adds complexity to the code

Can get confusing

Technical debt

Should be removed at some point

- For example, running two version of a feature should be temporary

Sometimes turns into an excuse to ship garbage

- Lack of focus on being truly "Done"



@benday | www.benday.com

DevOps?

DevOps

How are you developing, delivering, deploying, & monitoring?

"What is the process of getting a developer's changeset from check-in to production? How long does that take?"

If you want to go fast, start automating

- Automation removes hard / tedious stuff
- Automation provides reliability
- Automation helps you do more and go faster

Stability, Reliability, Repeatability, Release-ability



@benday | www.benday.com

DevOps increases the speed of feedback.

DevOps increases the speed of mistakes.

DevOps increases the speed of feedback.

Product Ownership in an Agile world is all about experimentation

Each new feature is an experiment

Build as little code as possible → Minimum Viable Product (MVP)

Ship it & gather feedback
- That's the experiment

You don't always want to commit to these experiments.



@benday | www.benday.com

***“Go big or go
home”***

VS.

**“Go small and quietly and go home
if it doesn’t work”**

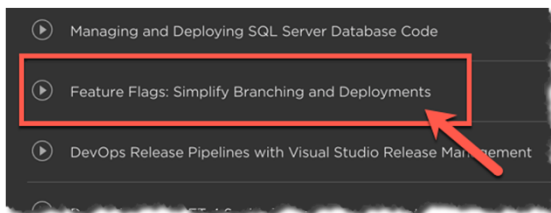
Technical Stuff

Want to see this talk and code again?



PLURALSIGHT

DevOps Skills for Developers
with Visual Studio and TFS2017



Feature Flags: Simplify
Branching and Deployments



@benday | www.benday.com

Implementation Tips

Create an IFeatureManager interface

Each feature flag should be a Boolean property
- NO MAGIC STRINGS!

Use Dependency Injection for your IFeatureManager

Makes it very easy to find and remove later



@benday | www.benday.com

DEMOS

Basic feature flag
- Search

Two features side-by-side

Undeploy a feature that's getting bad reviews

Private beta



@benday | www.benday.com

**A feature flag is (probably)
technical debt.**

**Any specific feature flag should eventually
be removed.**

Your feature flag system is forever.

**The individual feature flags
are temporary.**

Watch out for technical debt.

Any last questions?

Thanks.

benday@benday.com | www.benday.com

