QA Testing & Automated UI Tests with Selenium

Benjamin Day
TRAINER | COACH | DEVELOPER
@benday www.benday.com

Overview

QA / Manual Testing in a DevOps world

Test Case Management in TFS

QA Testing & Defect Tracking

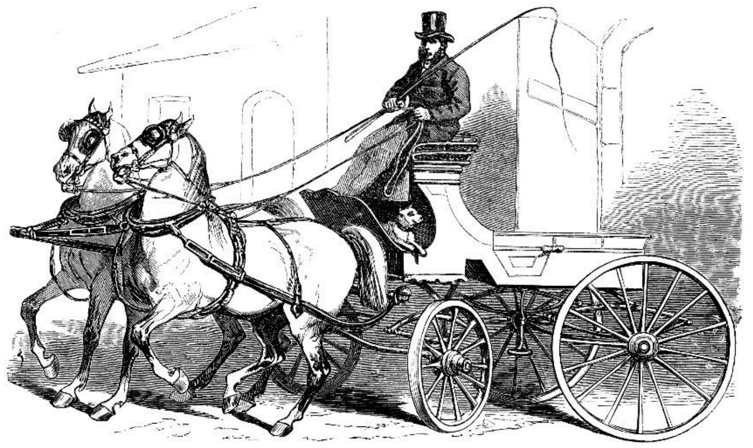
- Chrome extension

Automating Testing with Selenium

Run Selenium tests in TFS Builds

Review:
DevOps is about streamlined, automated flows.

DevOps plus Traditional QA doesn't really work.



The Traditional
QA Process

- Developers write the code
- Developers “kick it over the wall to QA”
- Testers bang on the app
- Testers send defects back to developers
- Developers fix the bugs
- Repeat
- Good enough quality → Release

DevOps needs a new kind of QA.

A New Kind of QA

QA & Development are on the same team

- No more “us vs. them”
- Developers can run QA tests, too

Focus on quality early

- No more “QA at the end”

Focus on feedback

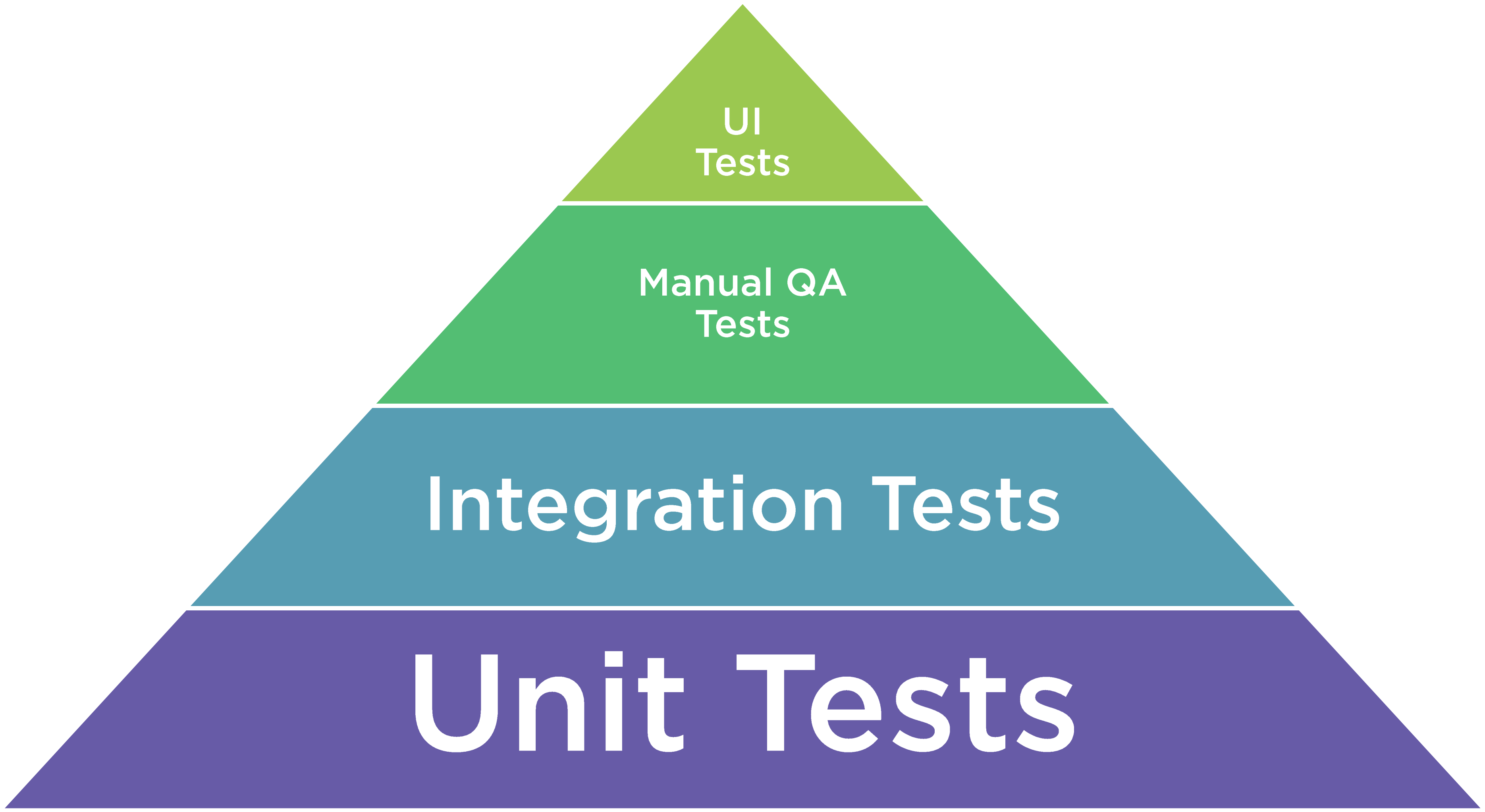
- “Is it working?” is valid feedback

Make deployment not a big deal

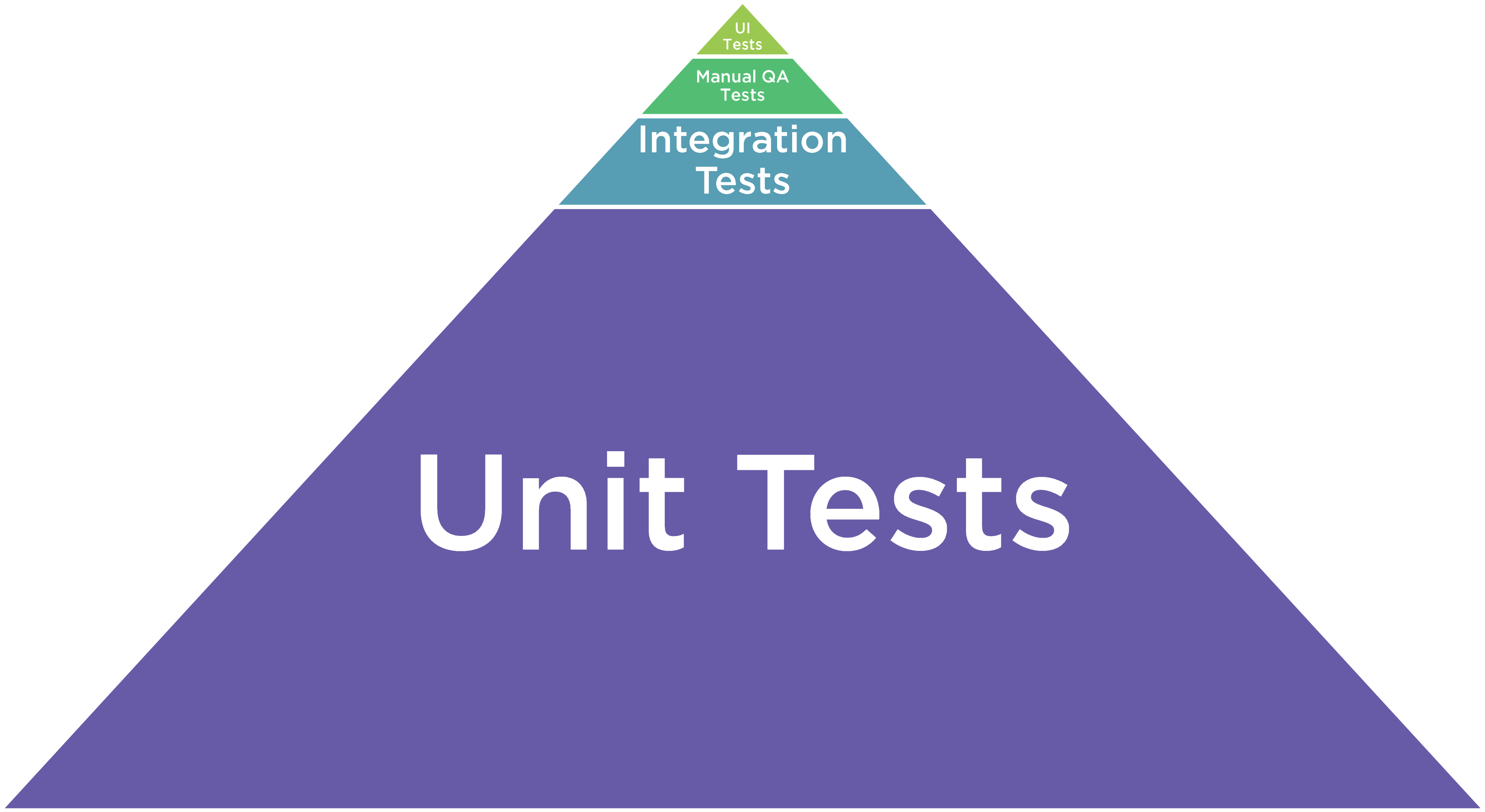
Exploratory testing rather than “is it broken?” testing

Use UI automation sparingly / strategically

The Testing Pyramid



Ben’s Testing Pyramid



#1 mistake of
Scrum / Agile teams:

Thinking that testing
happens are the end

#2 mistake of Scrum / Agile teams:

Testing is something that "QA" does

Requirements for “New QA”

Written test plans

Test plans are co-designed by the team

Developers run manual QA tests on their code before check-in

Try: Developers & Testers testing together

- Informal tests of partially done features
- Test & Fix without creating bugs in TFS
- Ultra-fast feedback

Higher quality builds means more time for exploratory testing

Want to super-charge your Scrum / Agile team?

Create test cases during your Sprint Planning Meeting.

QA Testing & Team Foundation Server

QA Tests = Manual Tests

Most functionality → web-based "Test" hub

- Cross-platform

Chrome Extension

- Exploratory Testing
- Screenshots
- Video recordings
- Create test cases, tasks, bugs

Microsoft Test Manager (MTM)

- Still there…
- …but there's not much reason to use it

Next up:
Create, manage, and run tests using TFS

Demo

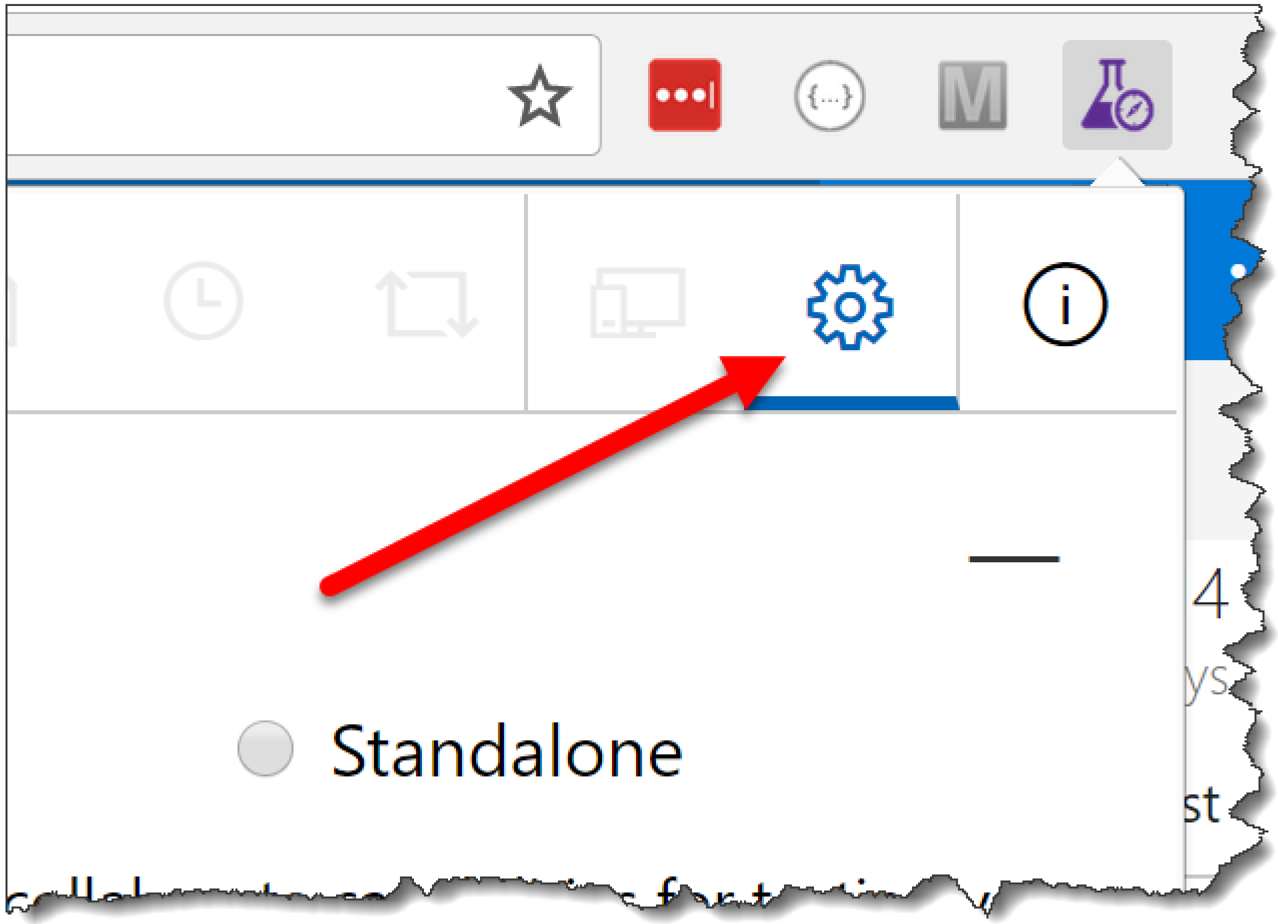
Create test cases using Team Foundation Server

Demo

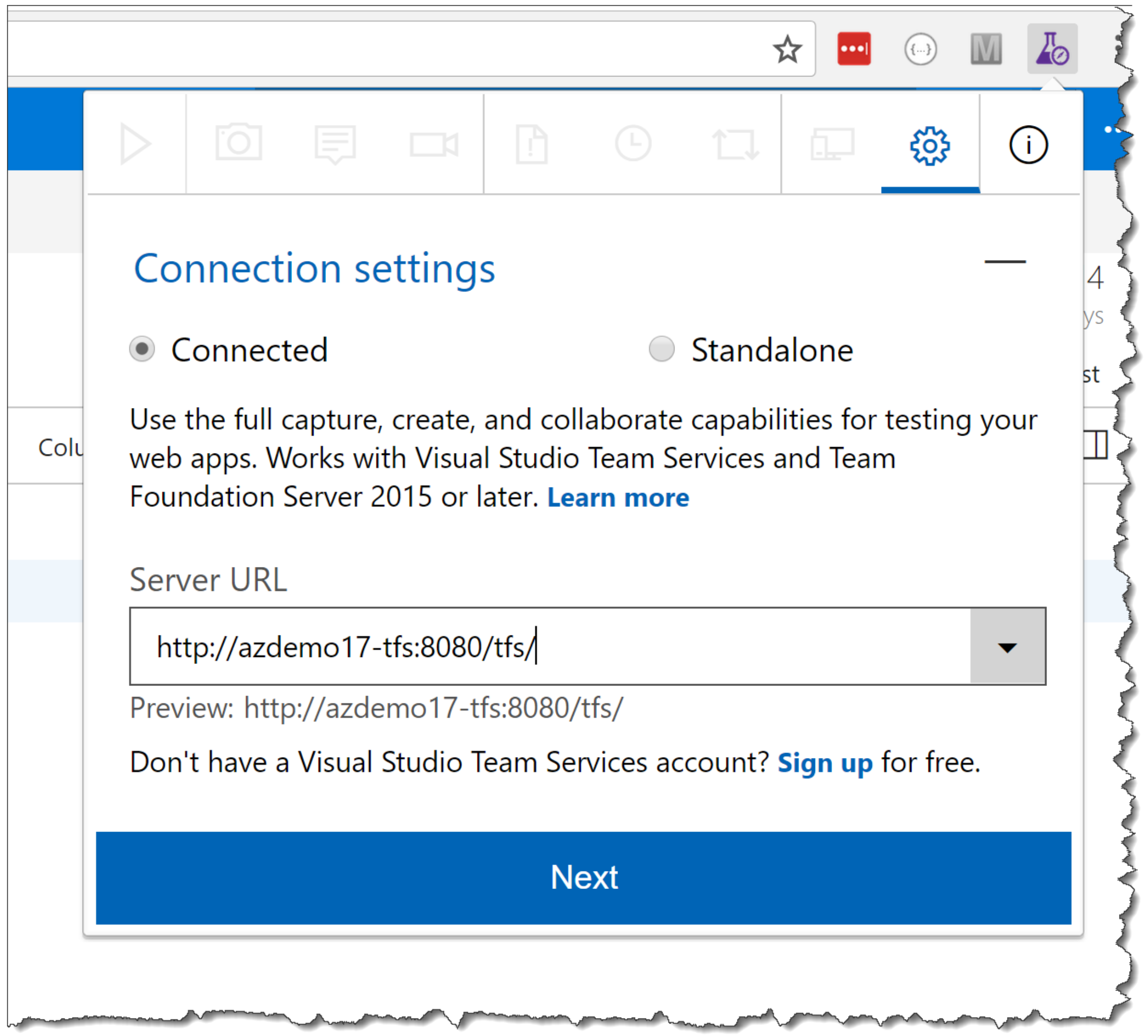
Run test cases

Test extension for Chrome

Step 1: Click the Gear Icon



Step 2: Enter Your TFS URL



Connection settings

● Connected ○ Standalone

Use the full capture, create, and collaborate capabilities for testing your web apps. Works with Visual Studio Team Services and Team Foundation Server 2015 or later. **Learn more**

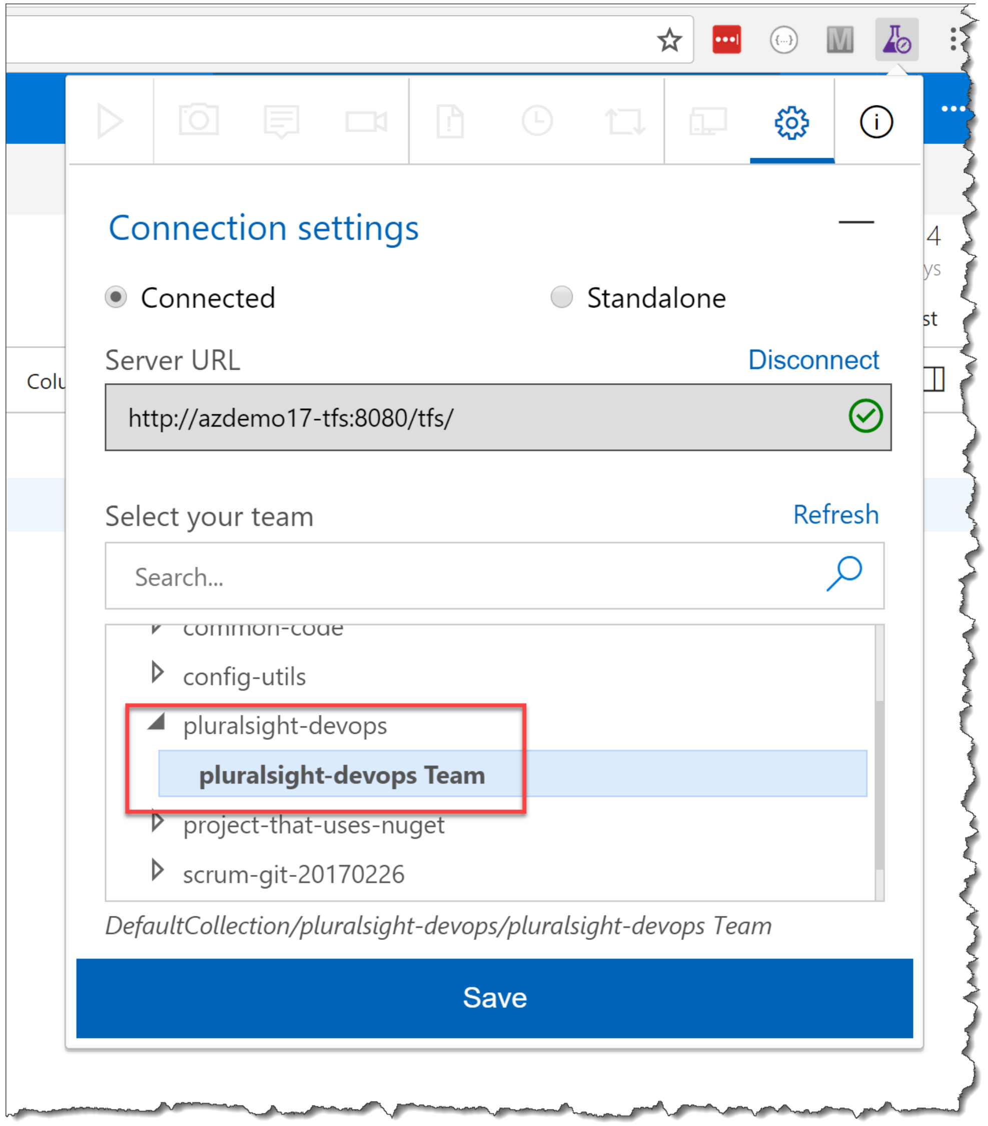
Server URL

http://azdemo17-tfs:8080/tfs/

Preview: http://azdemo17-tfs:8080/tfs/

Don't have a Visual Studio Team Services account? **Sign up** for free.

Step 3: Choose Your Project & Team



Connection settings

[✓ Connected] Standalone

Server URL: http://azdemo17-tfs:8080/tfs/

Disconnect

Select your team

Refresh

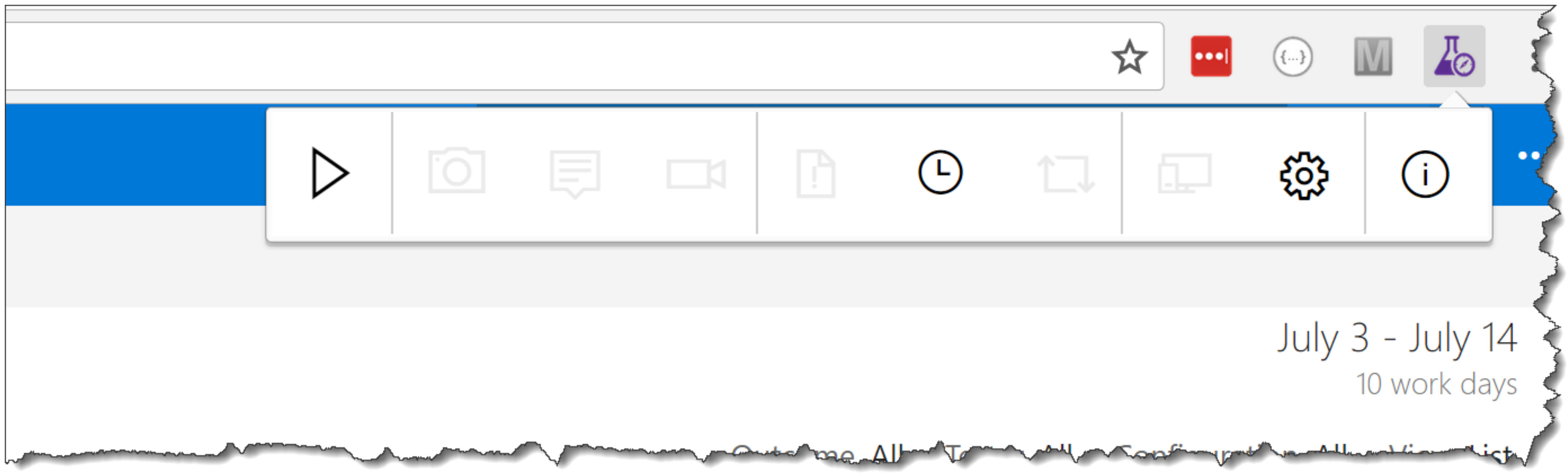
Search...

- common-code
- config-utils
- pluralsight-devops
  - **pluralsight-devops Team**
- project-that-uses-nuget
- scrum-git-20170226

DefaultCollection/pluralsight-devops/pluralsight-devops Team

Save

Step 4: Enjoy



Demo

View data for a test execution

Demo

Run test cases via the web

Record a bug

Demo

Shared steps

Shared parameters

Demo

Test configurations

Demo

Assign test cases

Demo

Exploratory testing on a “requirement”

Next up:
Automated UI Tests

User Interface Automation Tests
(UI Automation Tests)

Automated UI tests can help speed up DevOps QA.

UI Automation Tests in Visual Studio

Coded UI Tests

- Web applications
- Windows
- WPF / XAML

Web Performance Tests & Load Tests

- (Out of scope for this course)

Coded UI Tests + Selenium

- Web apps
- Cross-browser testing

Selenium

- Probably best to use it by itself

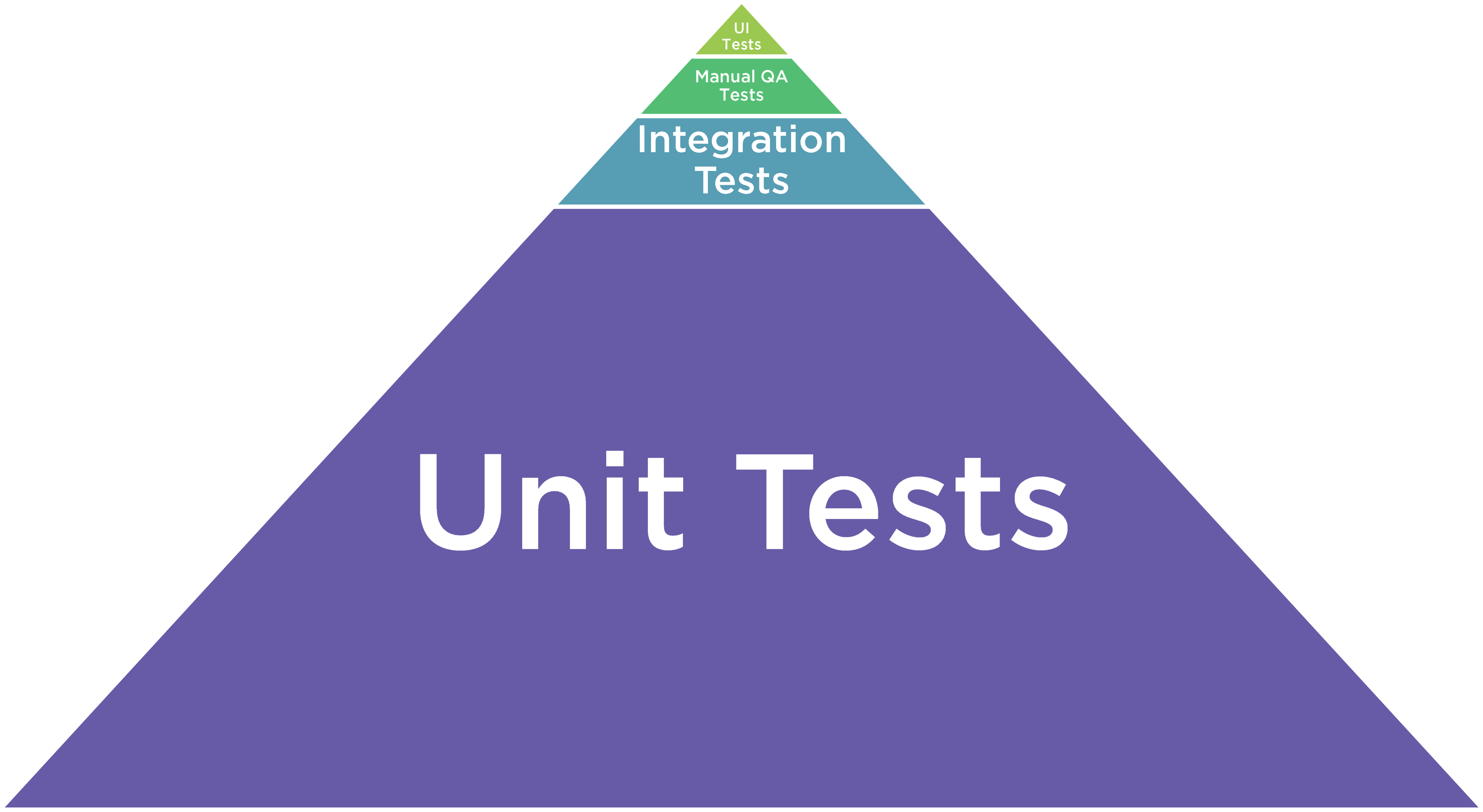
Web Performance Tests & Load Tests in the Pluralsight Library

Load Testing With Visual Studio 2013 by Benjamin Day	**3 hours**

I’m conflicted and skeptical about UI automation testing.

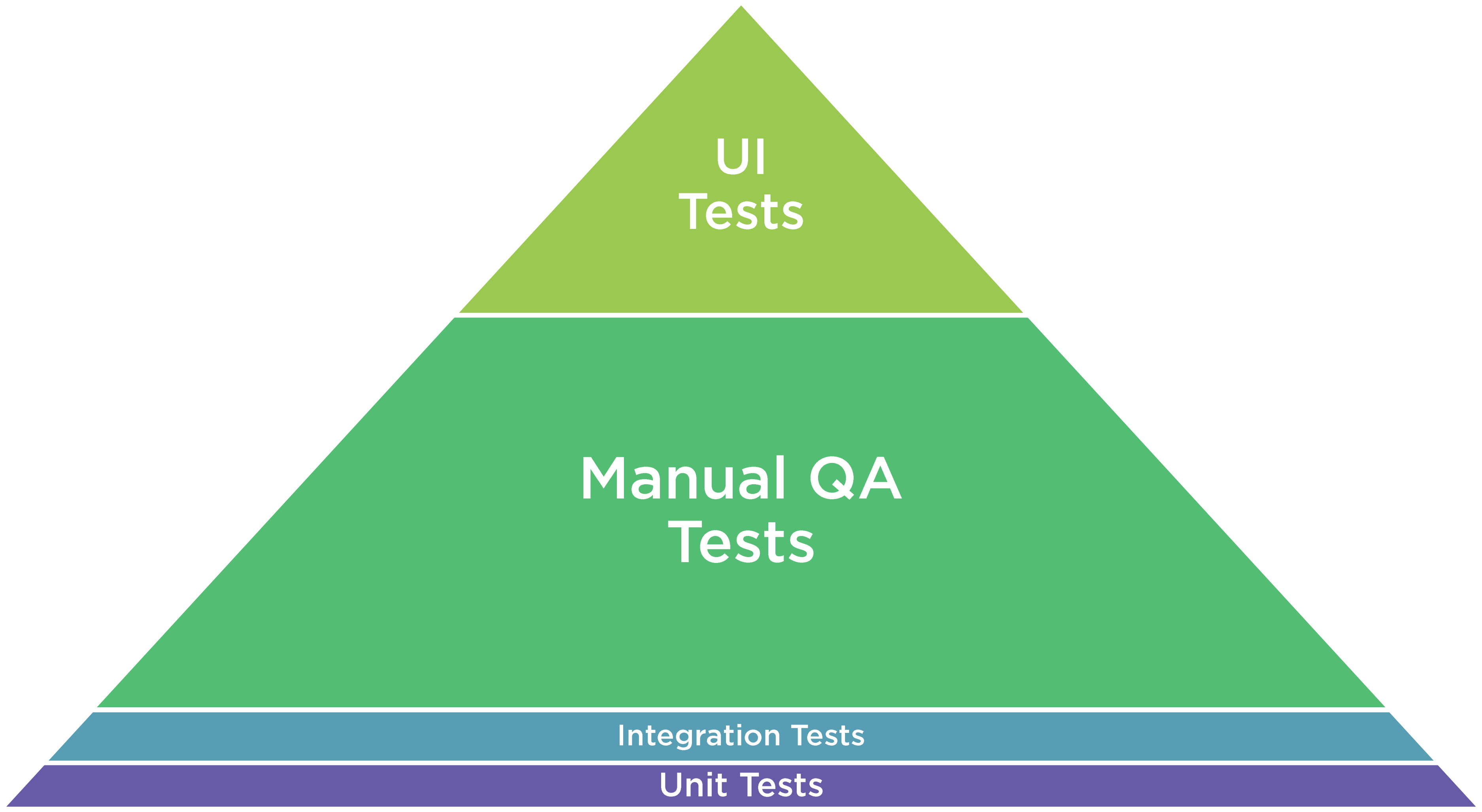
Teams tend to over-invest in UI automation tests.

Ben's Testing Pyramid



If you only have UI tests, you’re doing it wrong.

This is common and not great for DevOps.



Use UI Automation Sparingly

UI automation tests are not perfect

- You're teaching a computer to see

Hard to write

Hard to run

- Application must be deployed
- Integration tests
- (PhantomJS helps)

Break easily

- Maintenance "black hole"

Unit tests have a much better ROI

- ROI = Return on Investment

UI automation tests are a pain to maintain. Choose wisely to maximize ROI.

Choose Wisely: Good Candidates for UI Tests

Maximize ROI

- Maintenance cost vs. usefulness

Well-understood & stable

Flows that are simple to tests

- Tedious
- Basic Functionality
- Simple searches
- Simple administration

Flows that are slow to test by a human

- Tedious, Arduous
- Lots of data value comparisons

Critical flows

Choose Wisely: Bad Candidates for UI Tests

User interfaces that are constantly changing

Flows that are hard to script

- Dynamically generated UIs

Flows that require human eyeballs to verify

- Charts
- Graphs

Good Uses of UI Tests

Smoke Tests

Verify that the application works

Small number of tests

Core functionality

Common Ways to Run UI Tests

Manually with Visual Studio

Automated Build

Automated Release Management

Next up:
Selenium Tests

Demo

Getting started with Selenium

Create a basic test

Run using Chrome

Run using PhantomJS

Demo

Run Selenium tests from TFS Build

Start Presidents app in Docker

PhantomJS

- Does not require an "interactive" build agent

Summary

QA / Manual Testing in a DevOps world

Test Case Management in TFS

QA Testing & Defect Tracking

- Chrome extension

Automating Testing with Selenium

Run Selenium tests in TFS Builds

Next up:
Simplifying Deployments using Feature Flags