

# Git Demos

Author: Nho Luong  
Skill: DevOps Engineer Lead



The image displays two web pages showing certification status. The left page is from CertMetrics, showing 'Certification status' for various AWS and Microsoft certifications, including AWS Certified Solutions Architect - Professional, AWS Certified Security - Specialty, AWS Certified DevOps Engineer - Professional, AWS Certified Solutions Architect - Associate, and AWS Certified Cloud Practitioner. The right page is from Microsoft Learn, showing 'Credentials' for Microsoft Certified: DevOps Engineer Expert, Microsoft Certified: Azure Solutions Architect Expert, Microsoft Certified: Azure Administrator Associate, and Microsoft Certified: Azure Fundamentals. Both pages show active status and expiration dates.



# What I am covering (and what I am not)

- This is a 'How to use Git from with Visual Studio' session
  - ALM Rangers Version Control Guide <http://vsarbranchingguide.codeplex.com/>
  - Map TFVC actions to Git <https://www.visualstudio.com/en-us/articles/mapping-my-tfvc-actions-to-git>
  - Can also watch <https://channel9.msdn.com/Events/Build/2015/3-746>
  - Also recommended <http://www.pluralsight.com/courses/git-visual-studio-developers> (If you have PluralSight access, remember you can get 200 free minutes as a trial)
- This is not a detailed 'deep dive' Git tutorial
  - Look at <https://www.atlassian.com/git/tutorials/setting-up-a-repository> for a good walk through of command line and theory behind Git

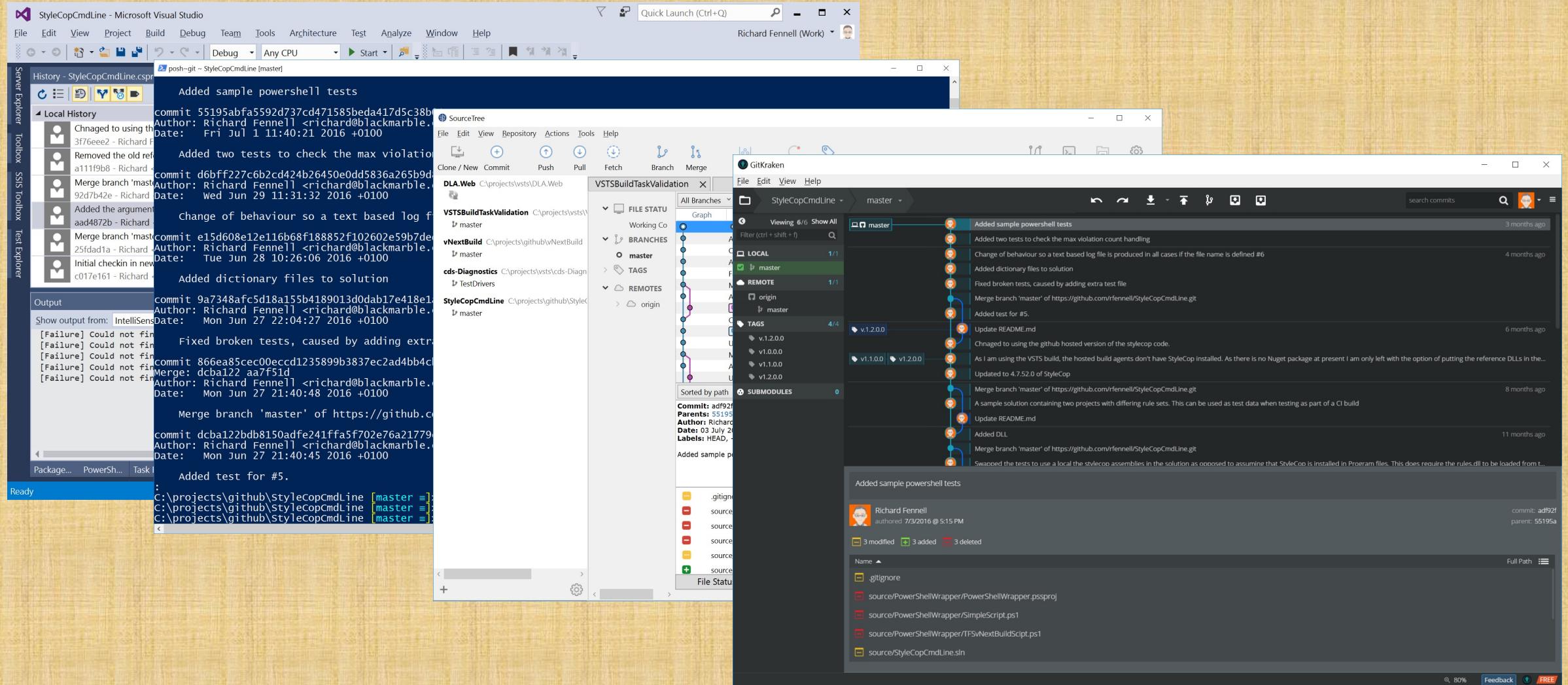
# Modern source-control approaches

		Strengths	Best for
Centralized Version Control (TFVC)	Server Workspaces	<ul style="list-style-type: none"><li>• Scales to very large codebases</li><li>• Fine level permission control</li><li>• Allows usage monitoring</li></ul>	<ul style="list-style-type: none"><li>• Large integrated codebases</li><li>• Control and auditability over source code down to the file level</li></ul>
	Local Workspaces	<ul style="list-style-type: none"><li>• Offline editing support</li><li>• Easy to edit files outside Visual Studio or Eclipse</li></ul>	<ul style="list-style-type: none"><li>• Medium-sized integrated codebases</li><li>• A balance of fine-grained control with reduced friction</li></ul>
Distributed Version Control (Git)		<ul style="list-style-type: none"><li>• Full offline experience</li><li>• Complete repository with portable history</li><li>• Simplified branching model</li></ul>	<ul style="list-style-type: none"><li>• Modular codebases</li><li>• Integrating with open source</li><li>• Highly distributed teams</li></ul>

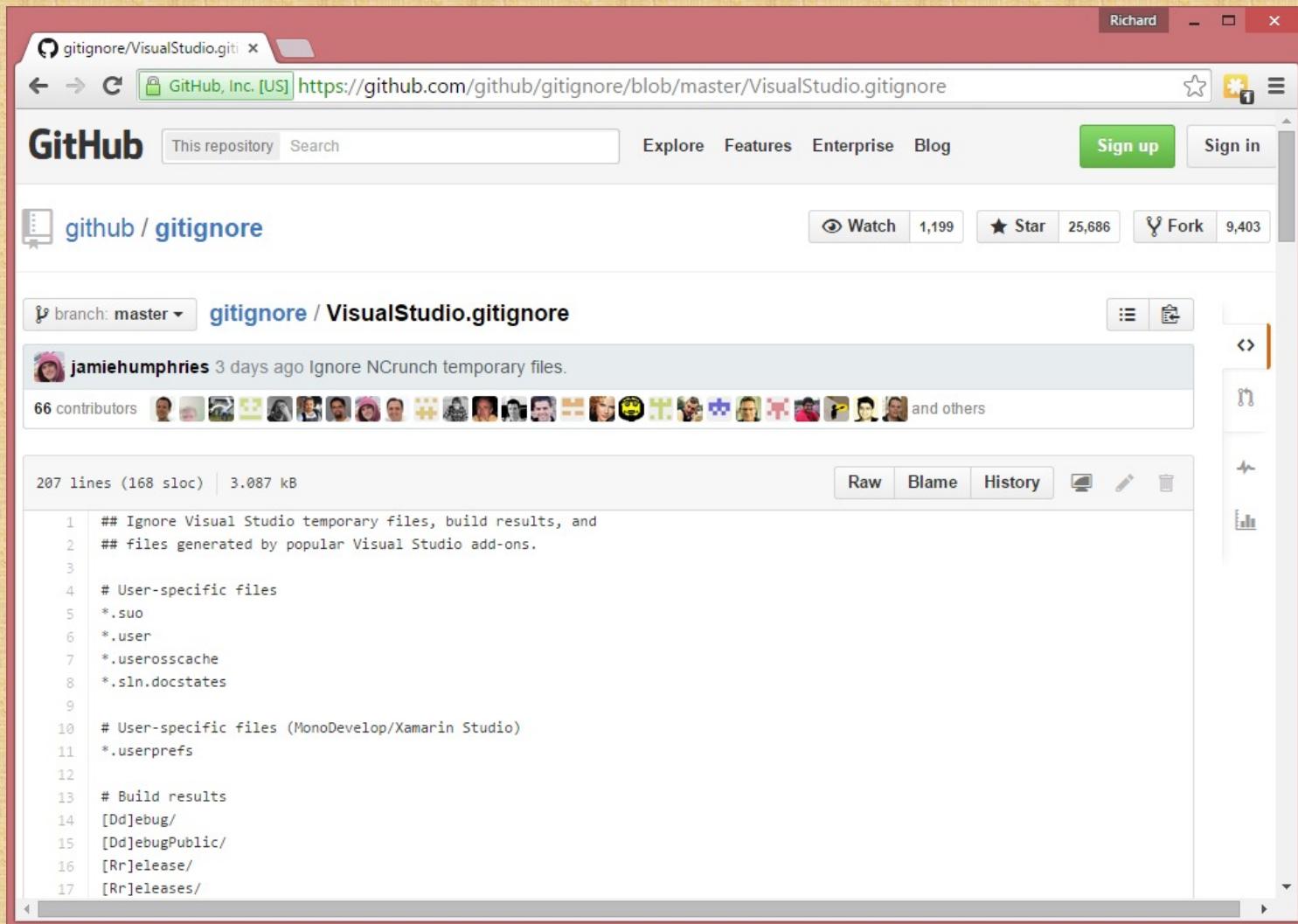
# TFVC to Git Terminology Cheat Sheet

TFS Version Control	Git
Workspace	Repository (aka. "Repo")
Get Latest (First time)	Clone
Get Latest (After first time)	Pull
Check in	Commit + Push
Check out	(just start typing)
Branch (are a set of folders)	Branch (are all in a single folder)
Merge	Merge
Code Review	"pull request"
Shelveset	Stash
Label	Tag

# There are many GIT Clients

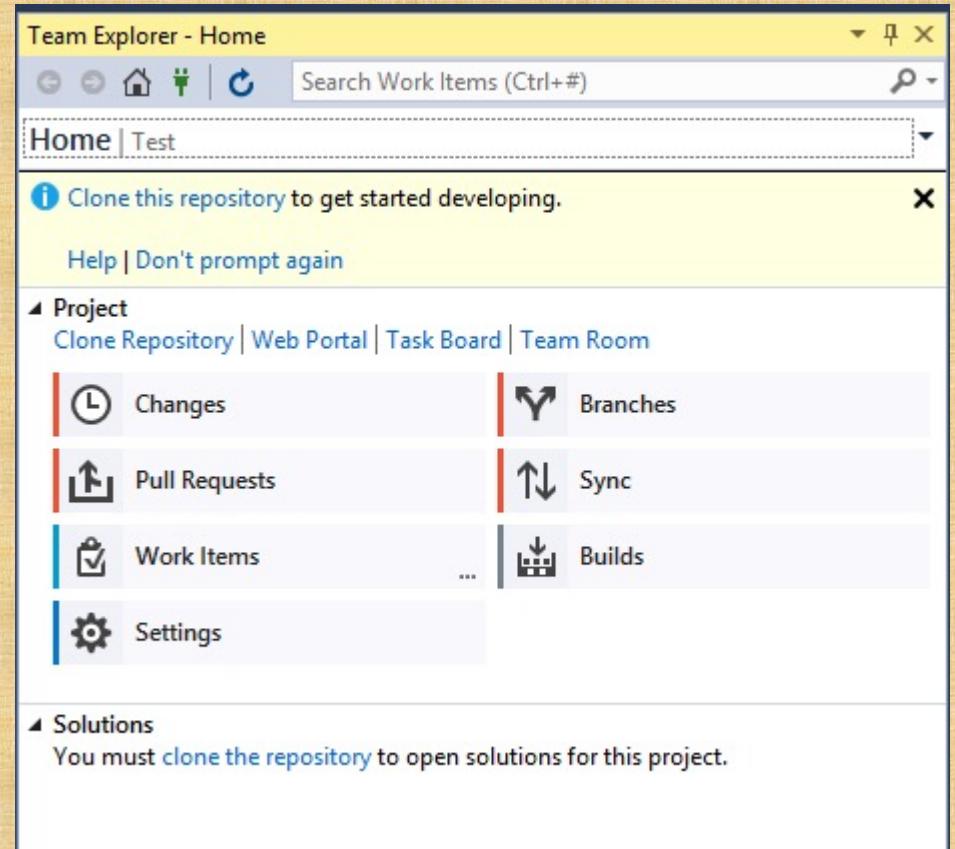


# Controlling which files to source control



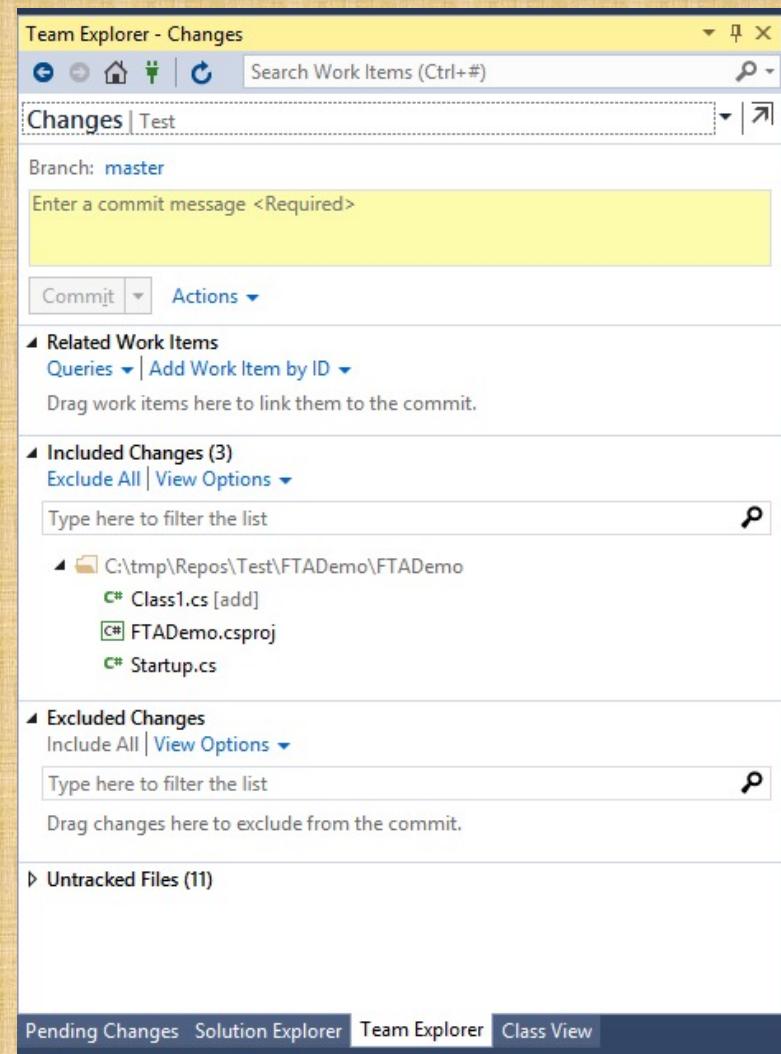
# Connecting to a Git Repo in Visual Studio

- Connecting to VSTS via Team Explorer (just like TFVC)
- Clone a repository to make a local copy



# Day to Day Operations

- Adding code (staging files)
- Commit to the local repo
- Sync to a remote repo
  - Fetch – a get from remote repo
  - Pull – a ‘fetch’ followed by a ‘merge’
  - Push – a push to a remote repo
- View History
  - Reset to bring a branch in your local repository back to the contents of a previous commit
  - Revert to undo the changes made in your commits pushed to shared branches

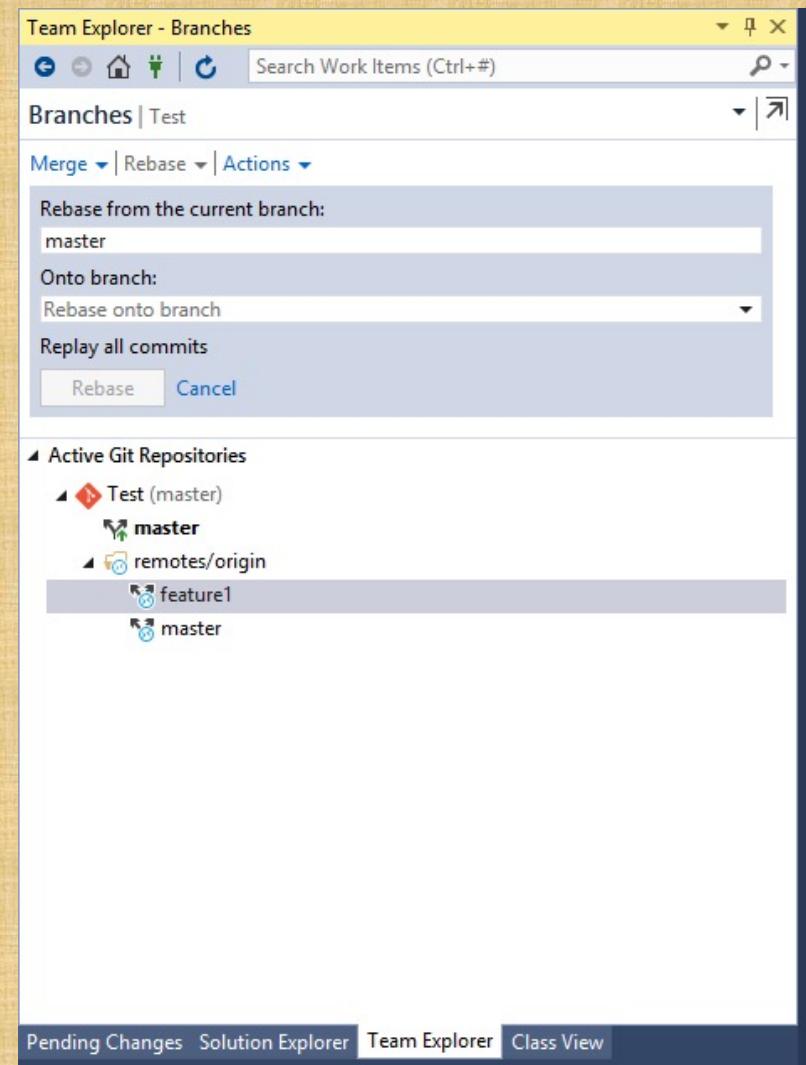


# Demo

## Basic Operations

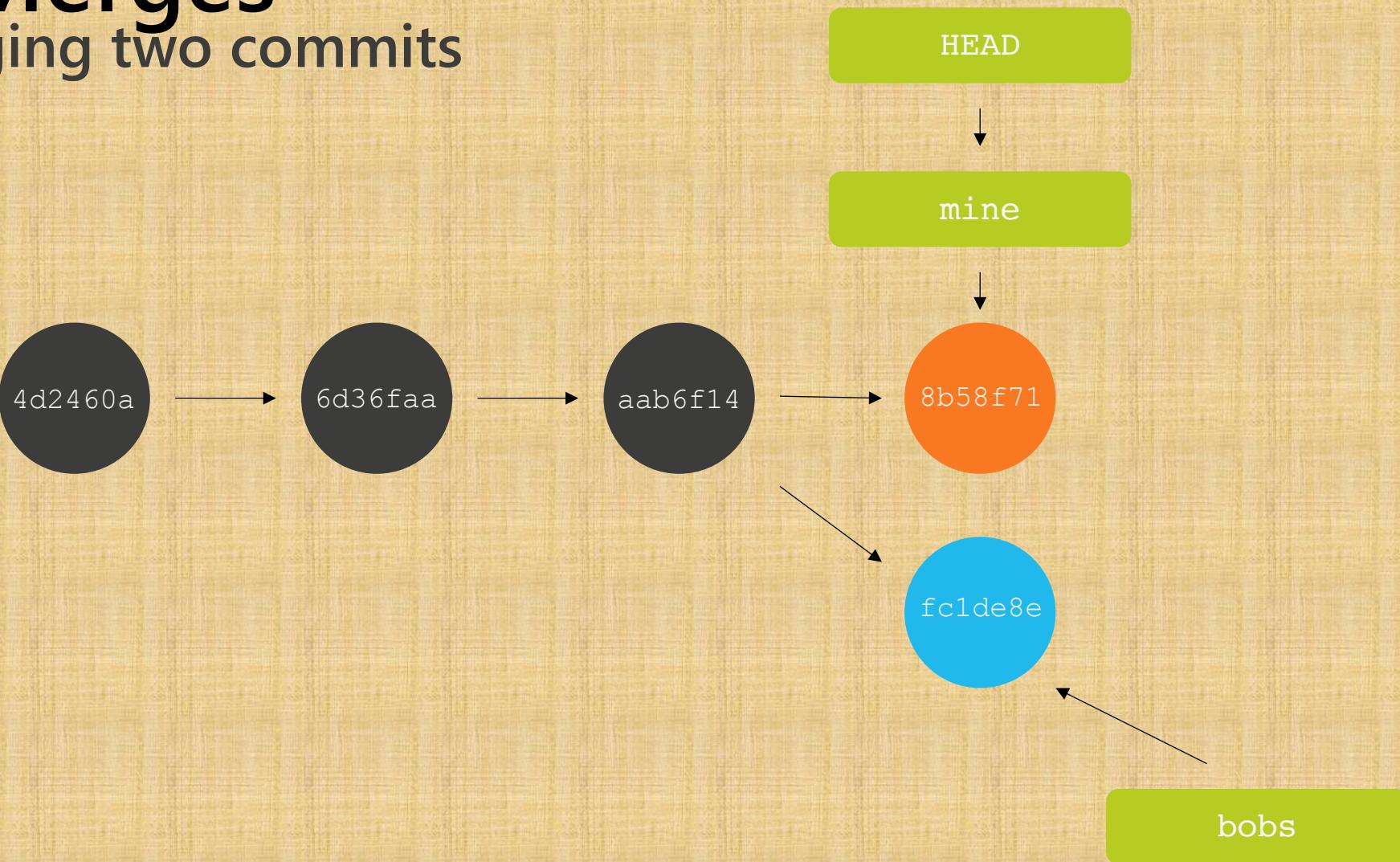
# Branching is different in GIT

- Can branch locally
- Can publish local branches to the remote 'origin' repo
- VS 2012/13 offers merge
- VS 2015/17 offers more merge/rebase options
- There are always command line tools for other scenarios



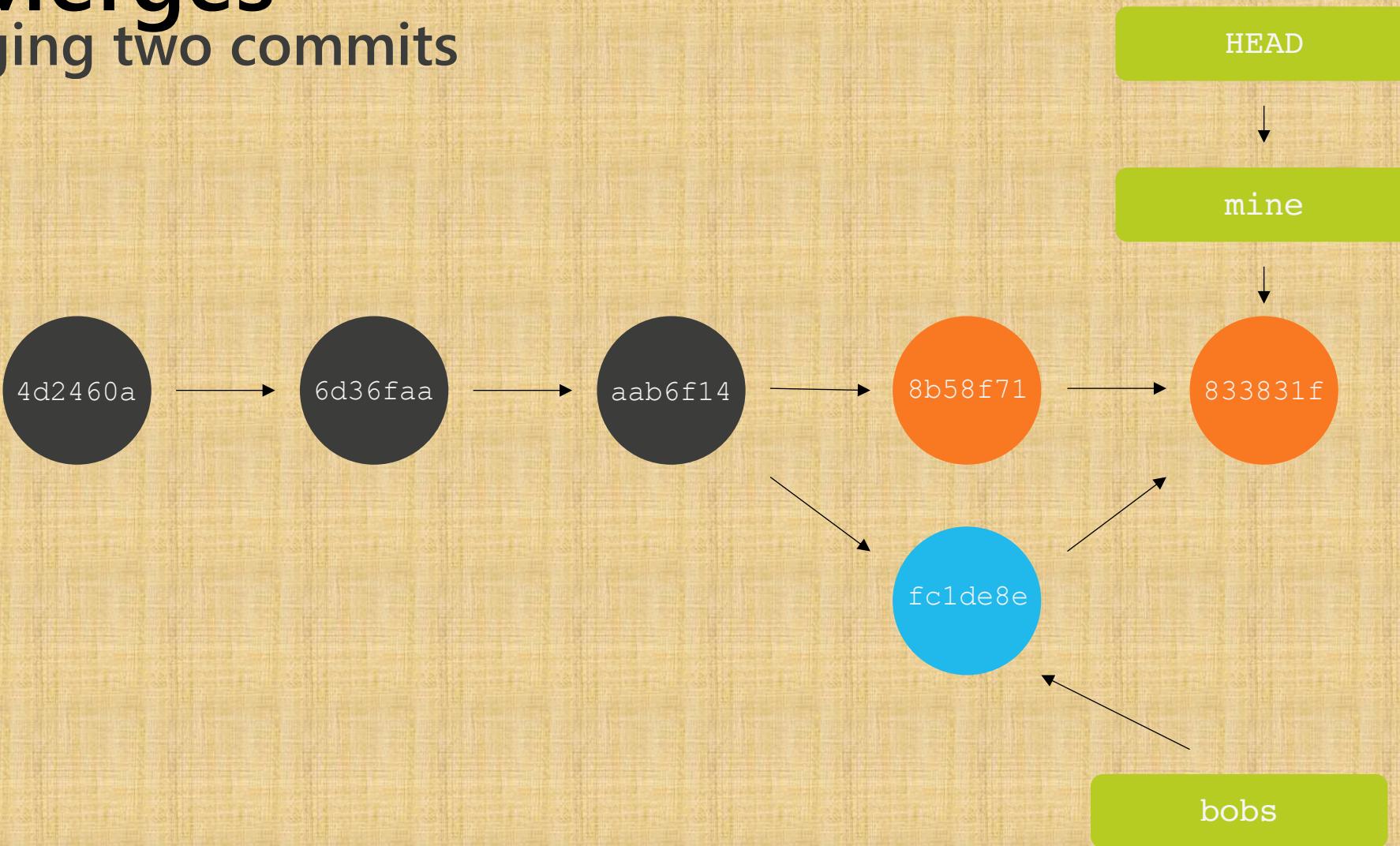
# Merges

## Merging two commits



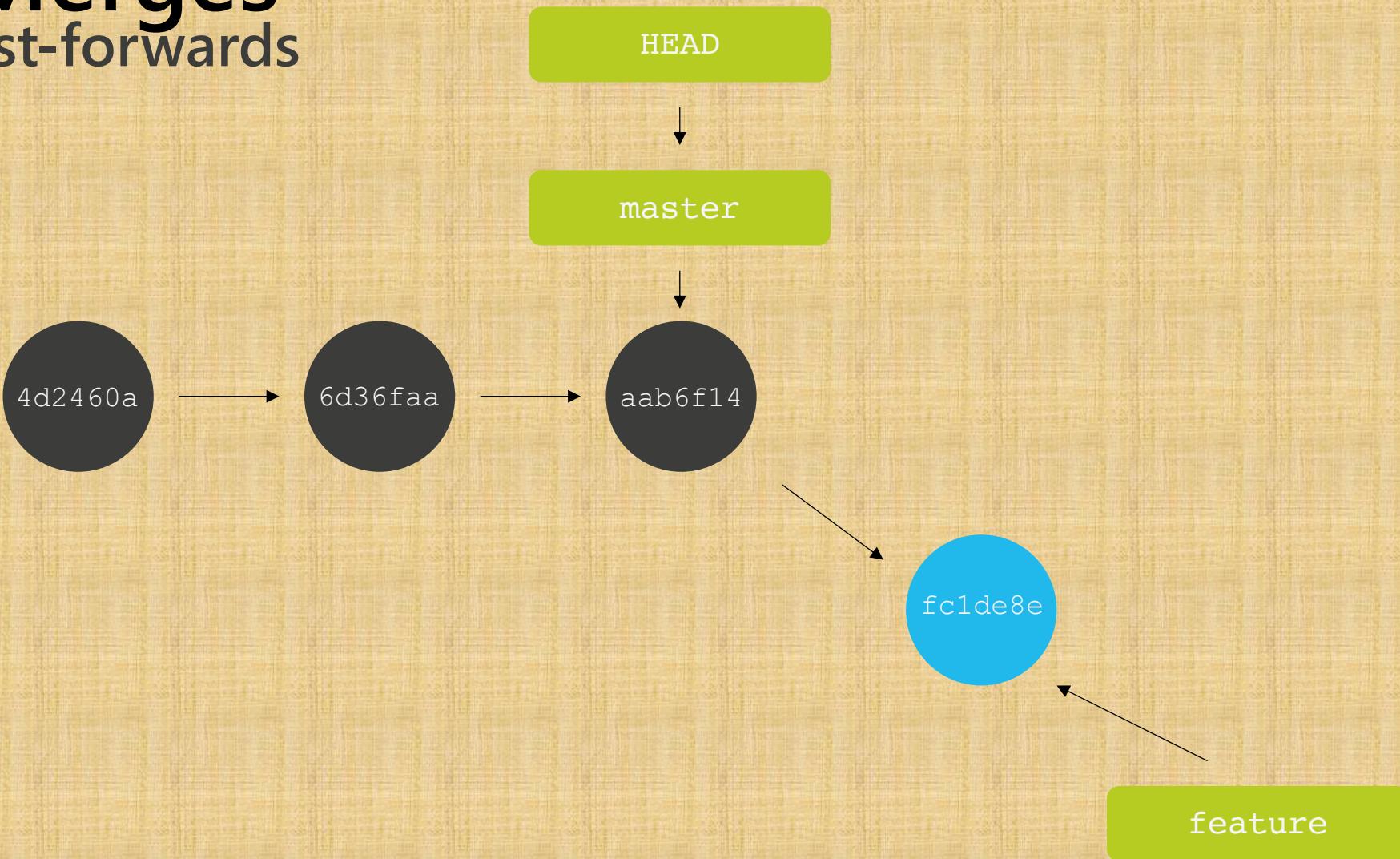
# Merges

## Merging two commits



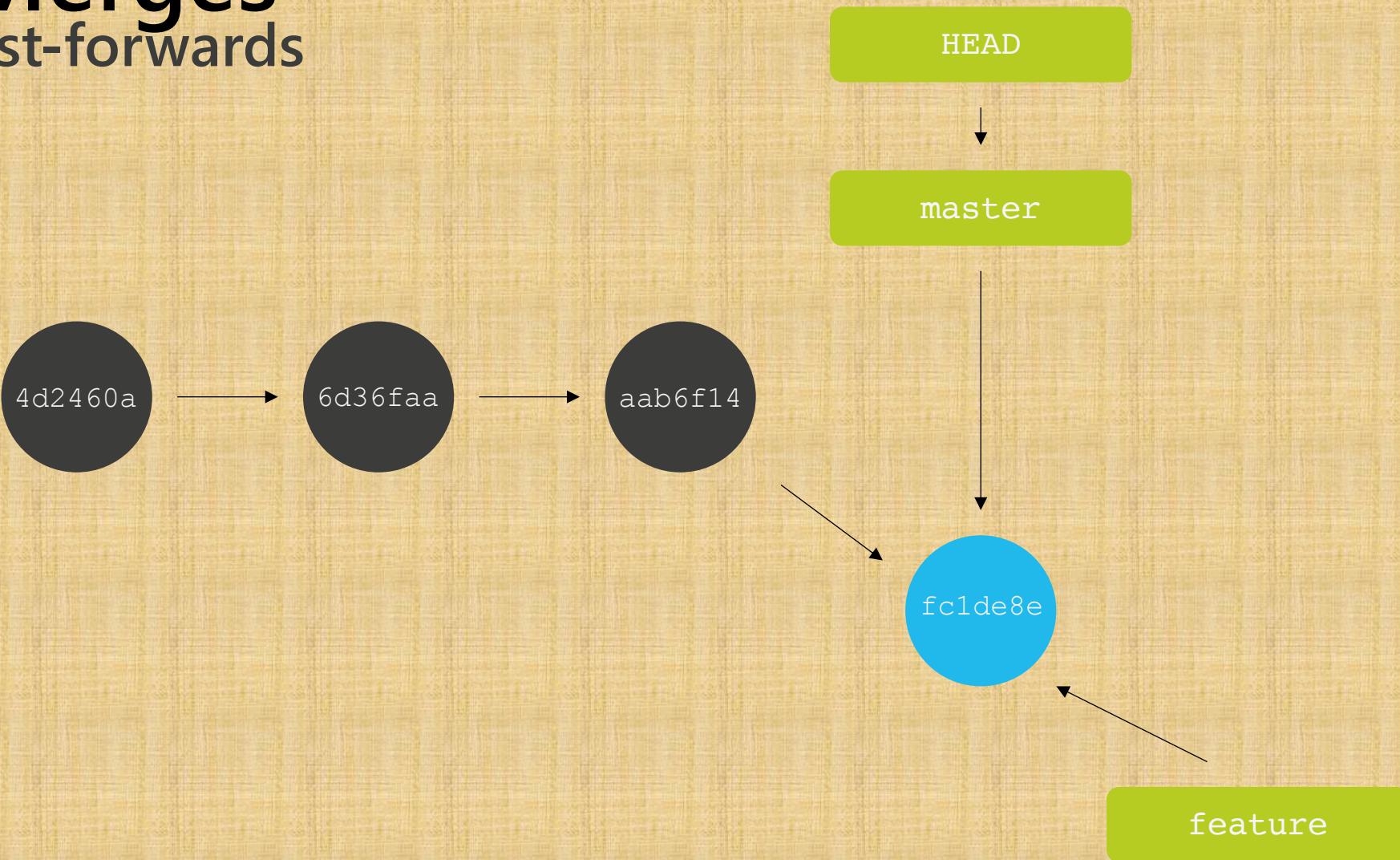
# Merges

## Fast-forwards



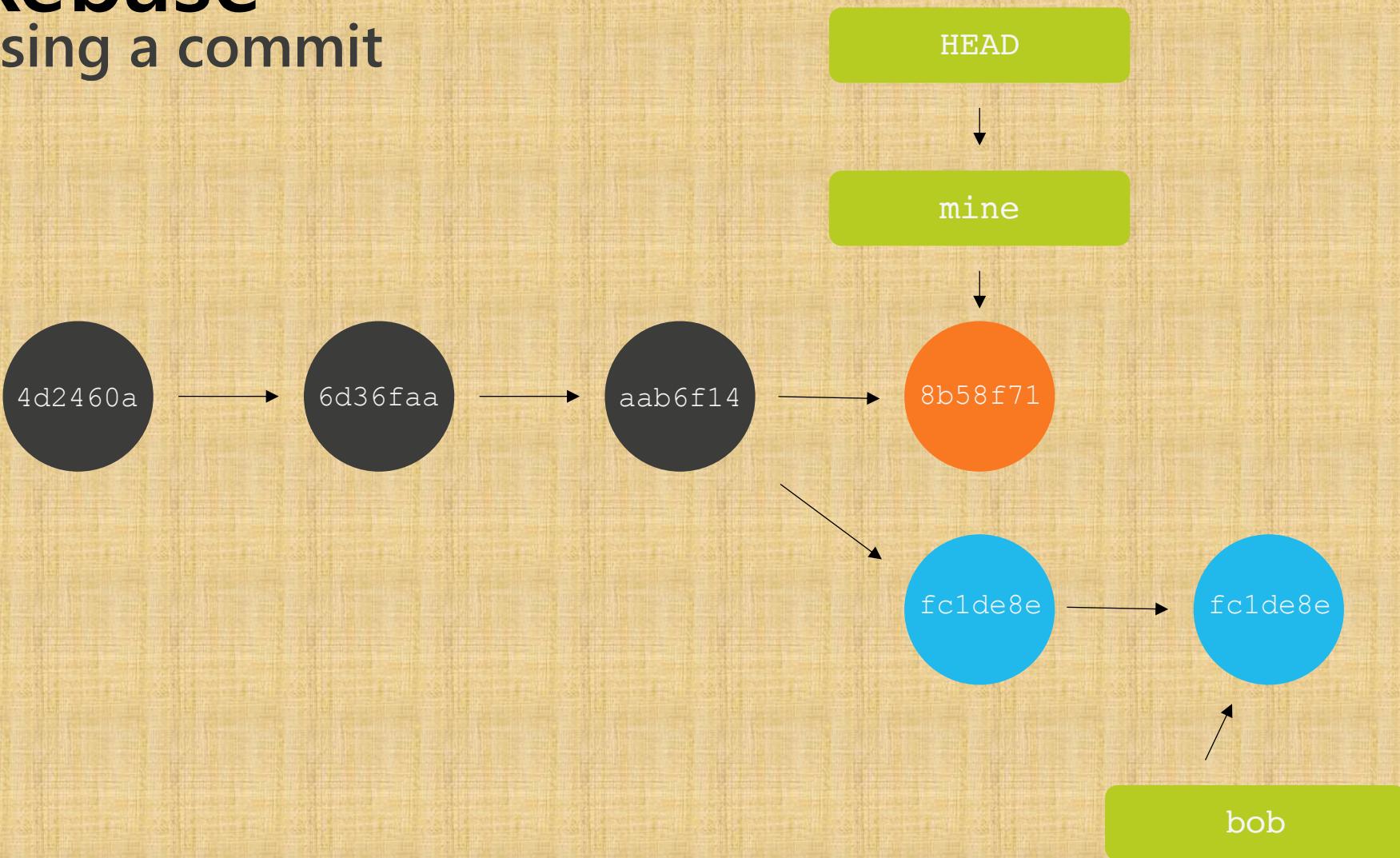
# Merges

## Fast-forwards



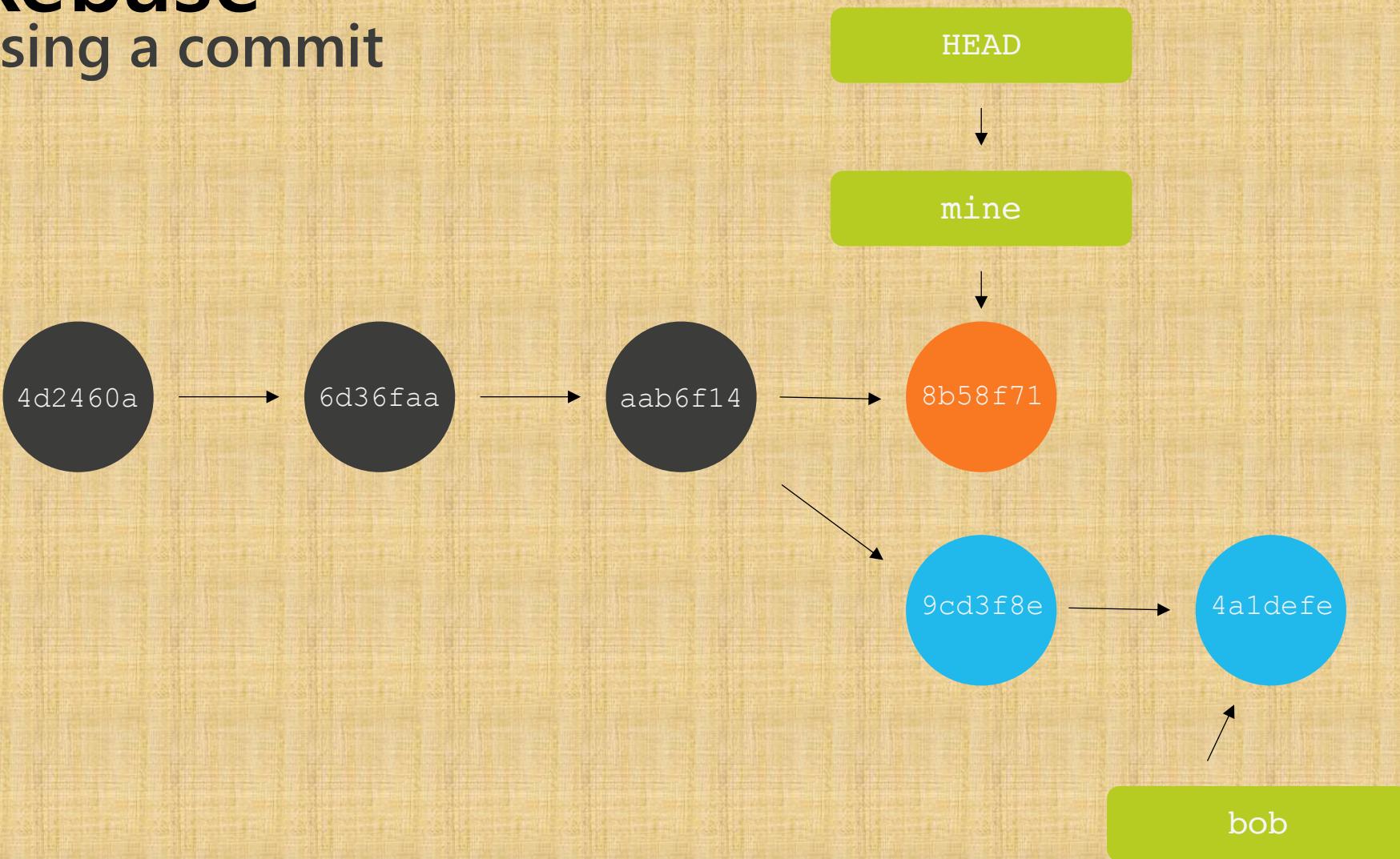
# Rebase

## Rebasing a commit



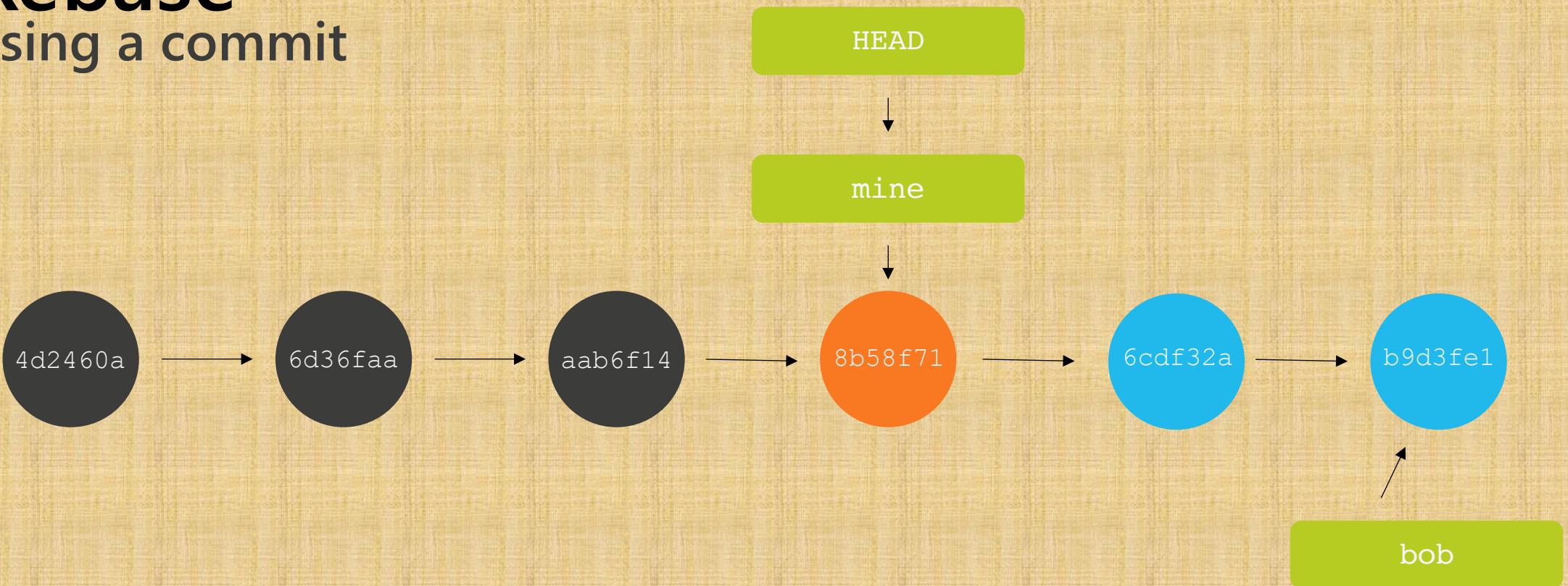
# Rebase

## Rebasing a commit



# Rebase

## Rebasing a commit

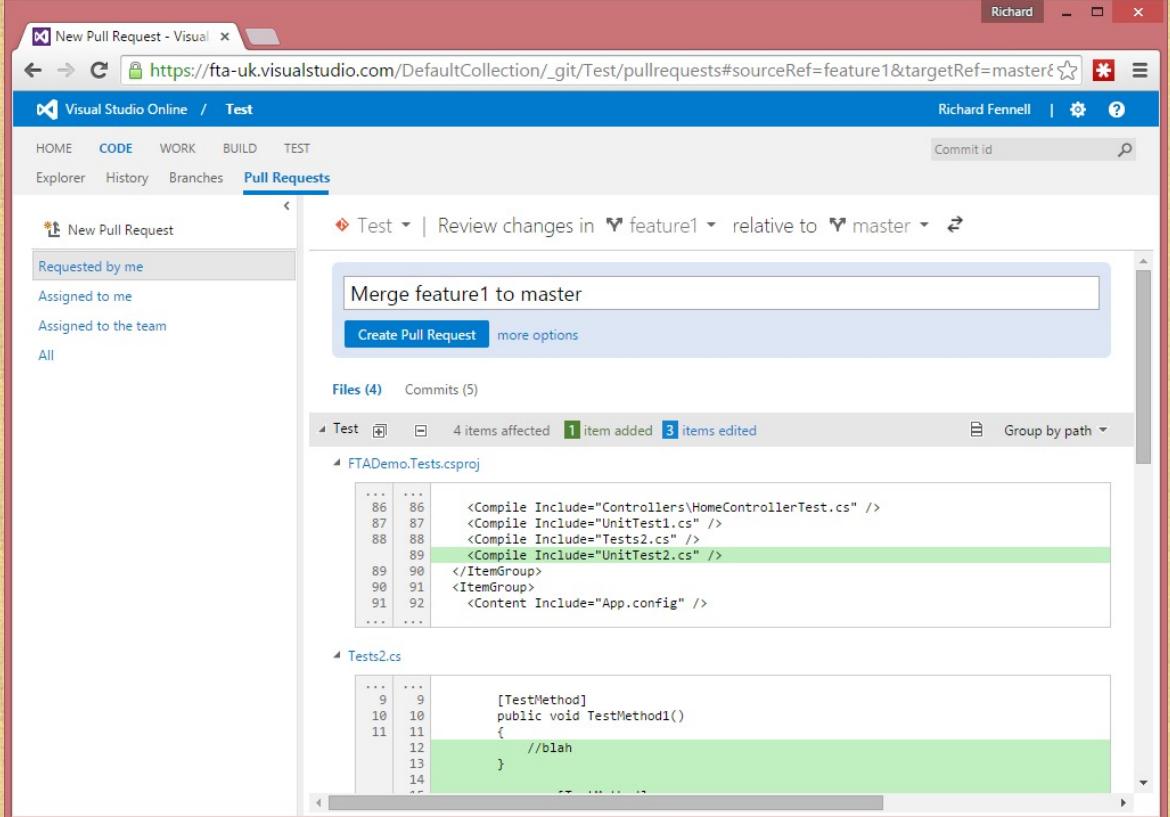


# Demo

Branch Operations

# Pull Requests

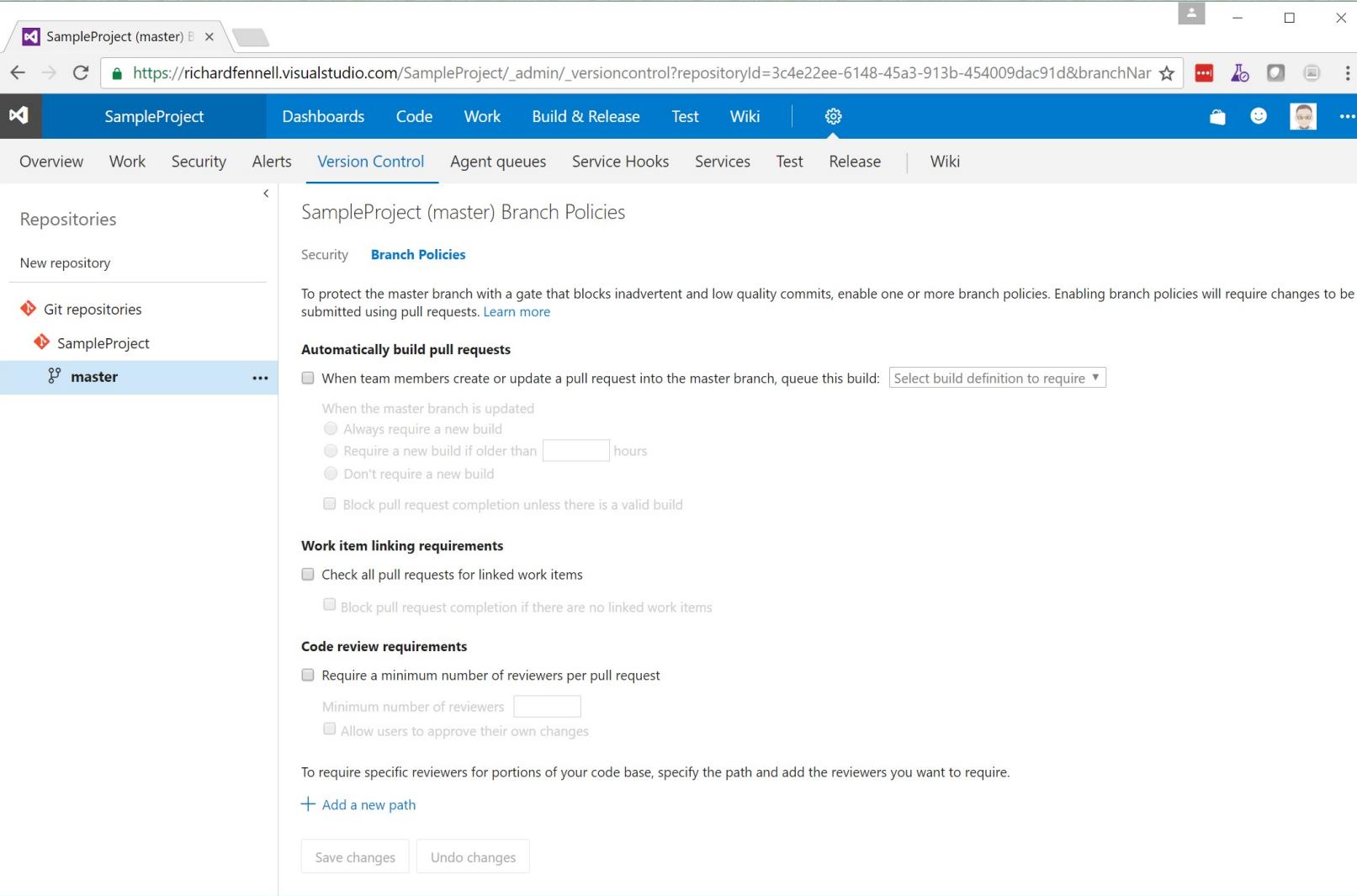
- Controls merging
- Allows code reviews



The screenshot shows a browser window for 'New Pull Request - Visual Studio Online' at the URL [https://fta-uk.visualstudio.com/DefaultCollection/\\_git/Test/pullrequests#sourceRef=feature1&targetRef=master](https://fta-uk.visualstudio.com/DefaultCollection/_git/Test/pullrequests#sourceRef=feature1&targetRef=master). The page is titled 'Merge feature1 to master'. On the left, there's a sidebar with 'Requested by me', 'Assigned to me', 'Assigned to the team', and 'All' filters. The main area shows a diff view with two files: 'FTADemo.Tests.csproj' and 'Tests2.cs'. The 'FTADemo.Tests.csproj' file has several changes highlighted in green, including additions to the 'ItemGroup' section. The 'Tests2.cs' file has one change highlighted in green, showing a new test method 'TestMethod1' with a comment '//blah'.

```
... ...
86 86 <Compile Include="Controllers\HomeControllerTest.cs" />
87 87 <Compile Include="UnitTest1.cs" />
88 88 <Compile Include="Tests2.cs" />
89 89 <Compile Include="UnitTest2.cs" />
90 90 </ItemGroup>
91 91 <ItemGroup>
92 92 <Content Include="App.config" />
...
...
9 9 [TestMethod]
10 10 public void TestMethod1()
11 11 {
12 12     //blah
13 13 }
14 14
15 15
```

# VSTS Branch Policies



The screenshot shows the 'Branch Policies' section of the VSTS interface for the 'SampleProject (master)' repository. The left sidebar lists repositories, showing 'Git repositories' and 'SampleProject' under it, with 'master' selected. The main content area is titled 'SampleProject (master) Branch Policies' and includes sections for 'Automatically build pull requests', 'Work item linking requirements', and 'Code review requirements'. It also provides instructions for specifying specific reviewers for code base portions.

SampleProject (master)

[https://richardfennell.visualstudio.com/SampleProject/\\_admin/\\_versioncontrol?repositoryId=3c4e22ee-6148-45a3-913b-454009dac91d&branchName=master](https://richardfennell.visualstudio.com/SampleProject/_admin/_versioncontrol?repositoryId=3c4e22ee-6148-45a3-913b-454009dac91d&branchName=master)

SampleProject

Dashboard Code Work Build & Release Test Wiki

Overview Work Security Alerts Version Control Agent queues Service Hooks Services Test Release Wiki

Repositories

New repository

Git repositories

SampleProject

master

SampleProject (master) Branch Policies

Security Branch Policies

To protect the master branch with a gate that blocks inadvertent and low quality commits, enable one or more branch policies. Enabling branch policies will require changes to be submitted using pull requests. [Learn more](#)

**Automatically build pull requests**

When team members create or update a pull request into the master branch, queue this build:

When the master branch is updated

Always require a new build

Require a new build if older than  hours

Don't require a new build

Block pull request completion unless there is a valid build

**Work item linking requirements**

Check all pull requests for linked work items

Block pull request completion if there are no linked work items

**Code review requirements**

Require a minimum number of reviewers per pull request

Minimum number of reviewers

Allow users to approve their own changes

To require specific reviewers for portions of your code base, specify the path and add the reviewers you want to require.

+ Add a new path

Save changes Undo changes

# Demo

## Pull Requests

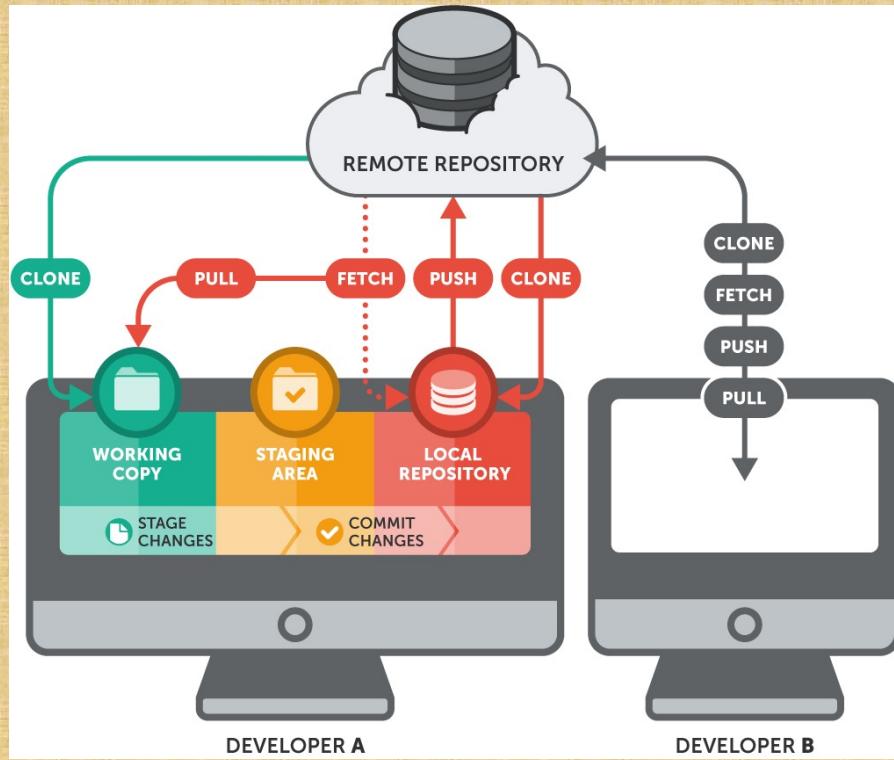
# Gotcha's

- Weird Errors in Visual Studio Git tools
  - Use the git command line tools
- Authentication outside of Visual Studio
  - Use SSH
    - <https://www.visualstudio.com/da-dk/docs/git/use-ssh-keys-to-authenticate>
  - (last resort) Use alternate credentials
    - <https://www.visualstudio.com/en-us/integrate/get-started/auth/overview>

# What if I am using another IDE?

- Most modern IDEs support Git today
- For older ones look at the GIT extensions for Visual Studio
  - <https://code.google.com/p/gitextensions/>
  - Supports VS 2005 and later
  - Also installs command line tools (using MsysGit)
- If doing a lot of command line look at POSH Git
  - <https://github.com/nholuongut/posh-git>
  - TIP: Install it using <https://chocolatey.org/> 'choco install poshgit'

# Quick Reference Summary



TFS Version Control	Git
Workspace	Repository (aka. "Repo")
Get Latest (First time)	Clone
Get Latest (After first time)	Pull
Check in	Commit + Push
Check out	(just start typing)
Branch (are a set of folders)	Branch (are all in a single folder)
Merge	Merge
Code Review	"pull request"
Shelveset	Stash
Label	Tag

Image Source: <https://www.git-tower.com/learn/git/ebook/en/command-line/remote-repositories/introduction>



Thank You