

Distributed Version Control for the Enterprise

Brad Wilson

Principal Software Developer

Tier 3



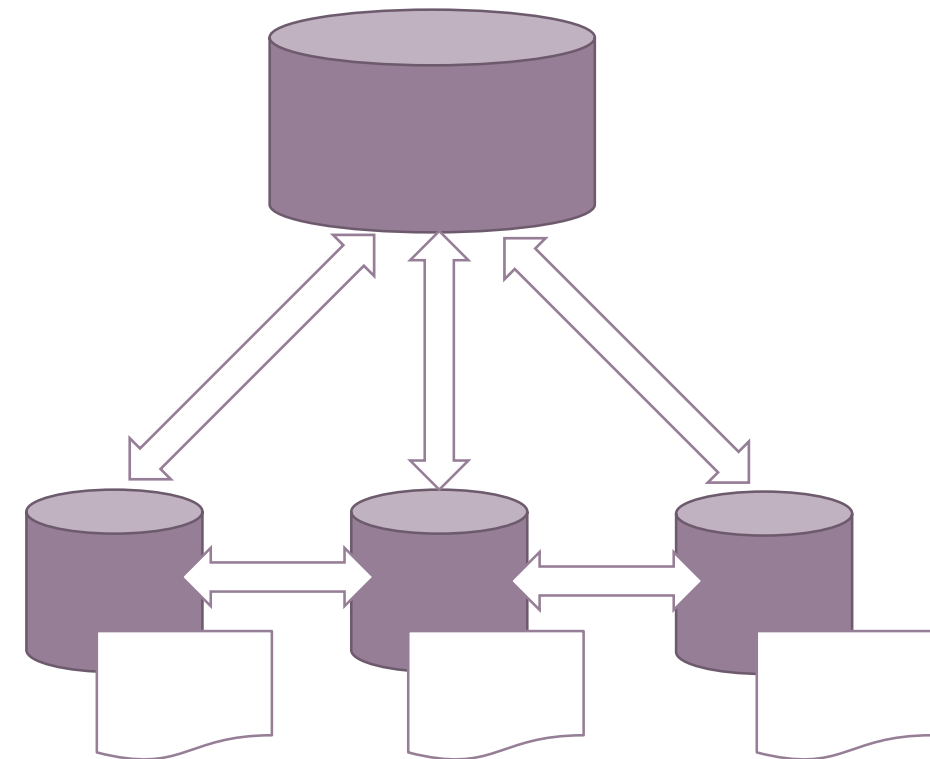
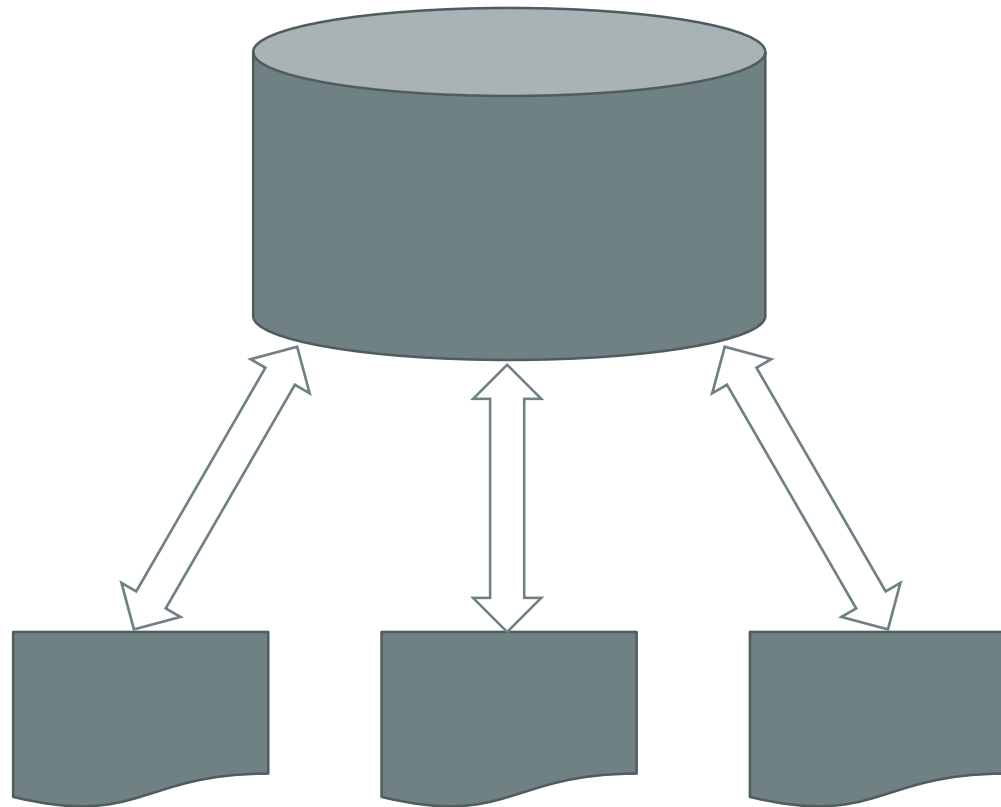
ALM Summit **3**
expertise worth sharing



DVCS and Git 101



Centralized vs. distributed



Why choose a DVCS?

- No single point of failure
- Developers can be productive while offline
- Offline operations are extremely fast... almost everything is offline
- No infrastructure required for skunk-works projects
- Admin in whatever model suits you best:
benevolent dictator, Lieutenant, or multiple committers
- Branching “just works”



A (very) brief history of Git

- Created by Linus Torvalds (~9 months)
Maintained by Junio Hamano
- Source control for Linux since v2.6 in April, 2005
- Open source (<http://github.com/git/git>)
- TFS Online, GitHub, CodePlex, Bitbucket, and countless startups and enterprises

“Nothing else was good enough or fast enough.”

Microsoft

facebook

NETFLIX

Google

LinkedIn



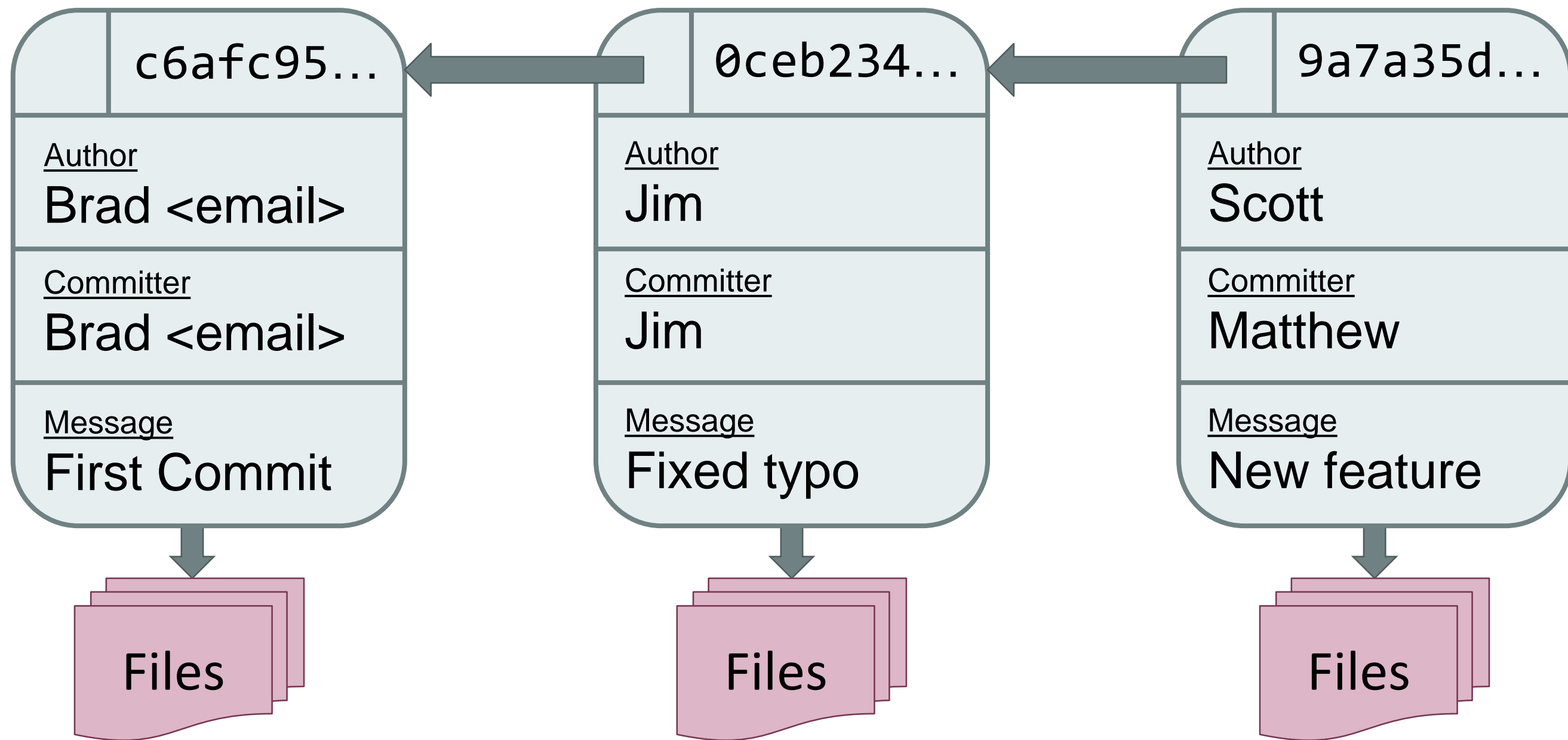
Why Git?

- Easy and fast branching and merging
- Snapshots of your working folder, not diffs
- Extremely fast with large repositories
- Cryptographic tamper-proofing of history
- Automatic support for renames

...it won in open source



Snapshots



Staging & committing

- `git add` puts copies of files (stages) into the “index”
 - *If you make further modifications to a file, you must re-stage it*
- `git commit` converts the index into a commit
 - `git commit -a` *does both ‘add’ (for mods/deletes) and ‘commit’*
 - *Still need to use `git add` for new files*
- `git status` shows you what’s staged and unstaged
- Use built-in diff to show you differences in files
 - `git diff` *shows you what you could stage*
 - `git diff --staged` *shows you would you could commit*



Sample: git status (unstaged)

```
# On branch master
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:    deleted.txt
#       modified:   foo.txt
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       new-file.txt
no changes added to commit (use "git add" and/or "git commit -a")
```



Sample: git status (staged)

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       deleted:    deleted.txt
#       modified:   foo.txt
#       new file:   new-file.txt
```



Sample: git diff

```
diff --git a/foo.txt b/foo.txt
index 3c619de..65d9d4f 100644
--- a/foo.txt
+++ b/foo.txt
@@ -1,4 +1,4 @@
  This is a new file.
+Inserted a line here
  And a new edit.
-This line I'm planning to delete later.
  Fixed!
```



gitk: xUnit.net

File Edit View Help

2.0 remotes/origin/2.0 Added more tests for xunit.runner.tdnet

- Tests for ResultVisitor in TdNet runner
- Deleted ExecutorTests (commented out)
- Rename Reflector2 to Reflector
- Ported Assert (& exceptions) to xunit2
- Ported over Assert, exceptions, and tests
- Checkpoint, implementing TdNet runner
- Add support for BeforeAfterTestAttribute on test methods and test classes
- Add IDisposable support to XunitTestCase
- Remove ITestObserver
- More cleanup around test running
- Started implementing the test running for Xunit2Controller
- Created skeleton Xunit1Controller based on code from ExecutorWrapper
- Removed Executor2 in favor of directly using ITestFramework instances
- Replaced ExecutorWrapper with a controller pattern
- Added source line information to test discovery
- Made Executor2 more testable. Moved some tests from Executor2Wrapper to Executor2. Introduced a
- Added TraitAttribute and support for it to XunitTestCase
- More API iteration
- New string assertion that encodes, shows partial values, and show difference points.
- More iteration on the design of the test discovery system
- Moved IClassTestCase and IMethodTestCase to xunit.abstractions
- Introduced xunit.abstractions, and started exploring the shape of multi-framework support
- Update to 2.0.0-alpha
- Added missing CodeAnalysisDictionary.xml
- Remove reflection-based Task discovery
- Fix for issue #9799
- Moved DoesNotThrow tests out of ThrowsTests.cs
- Fix for issue #9797
- Improved console runner message when running test assembly with no tests
- Fix for issue #9775
- Upgrade all runners to .NET 4.5 (except Visual Studio)
- Using statement cleanup
- The big conversion to v2.0
- master remotes/origin/master Mistakenly left in debugging MessageBox.Show calls
- Fixing more issues with DiaSession and async test methods
- Fix the issues with DiaSession with async test methods and derived test methods
- Add support for traits in VS2012 runner
- Updated xunit.msbuild to support SemVer package versions
- Pre-caching environment and test framework strings to reduce the time of AssemblyResultToXml()

SHA1 ID: 27bfe53d3aee1a03c932b270df22d92c9ccaba8c Row 2 / 66

Find next prev commit containing: Exact All fields

Brad Wilson <dotnetguy@gmail.com>	2012-12-31 15:44:35
Brad Wilson <dotnetguy@gmail.com>	2012-12-15 18:02:56
Brad Wilson <dotnetguy@gmail.com>	2012-12-15 17:28:40
Brad Wilson <dotnetguy@gmail.com>	2012-12-15 17:25:20
Brad Wilson <dotnetguy@gmail.com>	2012-12-15 15:09:03
Brad Wilson <dotnetguy@gmail.com>	2012-12-15 12:44:37
Brad Wilson <dotnetguy@gmail.com>	2012-12-15 12:07:38
Brad Wilson <dotnetguy@gmail.com>	2012-12-09 17:09:51
Brad Wilson <dotnetguy@gmail.com>	2012-12-09 13:45:49
Brad Wilson <dotnetguy@gmail.com>	2012-12-15 10:24:09
Brad Wilson <dotnetguy@gmail.com>	2012-12-01 17:04:10
Brad Wilson <dotnetguy@gmail.com>	2012-12-01 13:24:24
Brad Wilson <dotnetguy@gmail.com>	2012-12-01 11:04:28
Brad Wilson <dotnetguy@gmail.com>	2012-11-24 16:07:59
Brad Wilson <dotnetguy@gmail.com>	2012-11-23 17:22:47
Brad Wilson <dotnetguy@gmail.com>	2012-11-23 13:50:57
Brad Wilson <dotnetguy@gmail.com>	2012-11-17 17:33:41
Brad Wilson <dotnetguy@gmail.com>	2012-10-13 16:11:39
Brad Wilson <dotnetguy@gmail.com>	2012-10-13 15:42:54
bradwilson <dotnetguy@gmail.com>	2012-09-08 20:11:11
bradwilson <dotnetguy@gmail.com>	2012-09-08 17:17:55
bradwilson <dotnetguy@gmail.com>	2012-09-08 11:22:01
bradwilson <dotnetguy@gmail.com>	2012-09-03 17:24:35
bradwilson <dotnetguy@gmail.com>	2012-08-06 07:48:58
bradwilson <dotnetguy@gmail.com>	2012-08-04 14:49:32
bradwilson <dotnetguy@gmail.com>	2012-08-04 14:37:48
bradwilson <dotnetguy@gmail.com>	2012-08-04 14:22:03
bradwilson <dotnetguy@gmail.com>	2012-08-04 13:54:56
bradwilson <dotnetguy@gmail.com>	2012-08-04 13:46:06
bradwilson <dotnetguy@gmail.com>	2012-08-04 12:27:29
bradwilson <dotnetguy@gmail.com>	2012-08-04 12:21:03
bradwilson <dotnetguy@gmail.com>	2012-08-04 11:40:29
bradwilson <dotnetguy@gmail.com>	2012-07-30 20:52:54
bradwilson <dotnetguy@gmail.com>	2012-07-30 00:52:52
Brad Wilson <dotnetguy@gmail.com>	2012-11-23 14:27:01
Brad Wilson <dotnetguy@gmail.com>	2012-11-23 10:41:20
Brad Wilson <dotnetguy@gmail.com>	2012-11-20 20:03:36
Brad Wilson <dotnetguy@gmail.com>	2012-10-13 00:11:12
bradwilson <dotnetguy@gmail.com>	2012-07-29 23:34:44
bradwilson <dotnetguy@gmail.com>	2012-04-07 17:10:10



gitk: xUnit.net

File Edit View Help

- Fixing more issues with DiaSession and async test methods
- Fix the issues with DiaSession with async test methods and derived test methods**
- Add support for traits in VS2012 runner
- Updated xunit.msbuild to support SemVer package versions

SHA1 ID: ac9ebf9ab61d8a544fda99c62995c755842d9cca Row 37 / 66

Find next prev commit containing: Exact All fields

Search

Diff Old version New version Lines of context: 3 Ignore space change Line diff

```

<repository path="..\test\test.xunit.runner.utility\packages.config" />
+ <repository path="..\test\test.xunit.runner.visualstudio\packages.config" />
  <repository path="..\test\test.xunit\packages.config" />
</repositories>
\ No newline at end of file

----- src/xunit.runner.visualstudio/DiaSessionWrapper.cs -----
index 8948ece..09cf529 100644
@@ -1,4 +1,8 @@
  using System;
+using System.IO;
+using System.Reflection;
+using System.Security;
+using System.Security.Permissions;
  using Microsoft.VisualStudio.TestTools.UnitTesting;

  namespace Xunit.Runner.VisualStudio
@@ -8,7 +12,9 @@ namespace Xunit.Runner.VisualStudio
  // that the PDB file is missing or corrupt).
  public class DiaSessionWrapper : IDisposable
  {
+    AppDomain appDomain;
    string assemblyFilename;
+    DiaSessionWrapperHelper helper;
    DiaSession session;
    bool sessionHasErrors;

@@ -18,6 +24,45 @@ public DiaSessionWrapper(string assemblyFilename)

    if (Environment.GetEnvironmentVariable("XUNIT_SKIP_DIA") != null)
        sessionHasErrors = true;
  
```

Patch Tree

Comments

- packages/repositories.config
- src/xunit.runner.visualstudio/DiaSessionWrapper.cs
- src/xunit.runner.visualstudio/extension.vsixmanifest
- xunit.sln



<All Branches>

From: 4/ 7/2012

To: 12/31/2012

Filter by Subject, Messages, Paths, Authors, SHA-1, Bug-IDs

Author Email

Graph	Actions	Message	Author	Date
		Update to 2.0.0-alpha	bradwilson	8/6/2012 7:48:58 AM
		Added missing CodeAnalysisDictionary.xml	bradwilson	8/4/2012 2:49:32 PM
		Remove reflection-based Task discovery	bradwilson	8/4/2012 2:37:48 PM
		Fix for issue #9799	bradwilson	8/4/2012 2:22:03 PM
		Moved DoesNotThrow tests out of ThrowsTests.cs	bradwilson	8/4/2012 1:54:56 PM
		Fix for issue #9797	bradwilson	8/4/2012 1:46:06 PM
		Improved console runner message when running test assembly with no tests	bradwilson	8/4/2012 12:27:29 PM
		Fix for issue #9775	bradwilson	8/4/2012 12:21:03 PM
		Upgrade all runners to .NET 4.5 (except Visual Studio)	bradwilson	8/4/2012 11:40:29 AM
		Using statement cleanup	bradwilson	7/30/2012 8:52:54 PM
		The big conversion to v2.0	bradwilson	7/30/2012 12:52:52 AM
		master origin/master origin/HEAD Mistakenly left in debugging MessageBox.Show calls	Brad Wilson	11/23/2012 2:27:01 PM
		Fixing more issues with DiaSession and async test methods	Brad Wilson	11/23/2012 10:41:20 AM
		Fix the issues with DiaSession with async test methods and derived test methods	Brad Wilson	11/20/2012 8:03:36 PM
		Add support for traits in VS2012 runner	Brad Wilson	10/13/2012 12:11:12 AM
		Updated xunit.msbuild to support SemVer package versions	bradwilson	7/29/2012 11:34:44 PM
		Pre-caching environment and test framework strings to reduce the time of AssemblyResult.ToXml()	bradwilson	4/7/2012 5:10:10 PM
		Improve performance of acceptance tests	bradwilson	4/7/2012 5:32:23 PM
		Fixed incorrect output path for unit test build	bradwilson	7/29/2012 9:48:40 PM
		Incorrect path to nuget.exe for making packages	bradwilson	7/23/2012 9:31:31 AM
		Fix tests broken by TeamCity's use of JGit	bradwilson	7/23/2012 9:06:30 AM

SHA-1: ac9ebf9ab61d8a544fda99c62995c755842d9cca

* Fix the issues with DiaSession with async test methods and derived test methods

Path	Extension	Status	Lines added	Lines removed
packages/repositories.config	.config	Modified	1	0
src/xunit.runner.visualstudio/DiaSessionWrapper.cs	.cs	Modified	100	2
src/xunit.runner.visualstudio/extension.vsixmanifest	.vsixmanifest	Modified	2	2
xunit.sln	.sln	Modified	2	2

Showing 66 revision(s), from revision 399ea8c to revision 7c6e83d - 1 revision(s) selected



DiaSessionWrapper.cs:ac9ebf9 - TortoiseGitMerge

Main Style ? i

Open Save Save as Reload Undo Copy Paste Use Blocks Create patch file Previous difference Next difference Previous conflict Show Whitespaces

DiaSessionWrapper.cs:44017b8 - DiaSessionWrapper.cs:ac9ebf9

```
1 using System;
2 using System.IO;
3 using System.Reflection;
4 using System.Security;
5 using System.Security.Permissions;
6 using Microsoft.VisualStudio.TestTools.UnitTesting;
7
8 namespace Xunit.Runner.VisualStudio
9 {
10     // This class wraps DiaWrapper to get a delayed creation, as well as setting up the ability for
11     // us to stop making queries once we know that DiaSession is going to throw (which probably means
12     // that the PDB file is missing or corrupt).
13     public class DiaSessionWrapper : IDisposable
14     {
15         AppDomain appDomain;
16         string assemblyFilename;
17         DiaSessionWrapperHelper helper;
18         DiaSession session;
19         bool sessionHasErrors;
20
21         public DiaSessionWrapper(string assemblyFilename)
22         {
23             this.assemblyFilename = assemblyFilename;
24
25             if (Environment.GetEnvironmentVariable("XUNIT_SKIP_DIA") != null)
26                 sessionHasErrors = true;
27             else
28             {
29                 try
30                 {
31                     AppDomainSetup setup = new AppDomainSetup
32                     {
```

For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel Left View: UTF8 BOM CRLF / - 2 / + 100 Right View: UTF8 BOM CRLF Col



What is “HEAD”?

- Your working folder contains a copy of source code
- The point in the repository you’re working from is called HEAD
- You will typically have HEAD pointing at a branch (f.e., `master`)
- Use `git checkout` to change what HEAD is pointing to
 - Your working folder updates when you check out
 - You may not be able to check out if you have conflicting pending changes

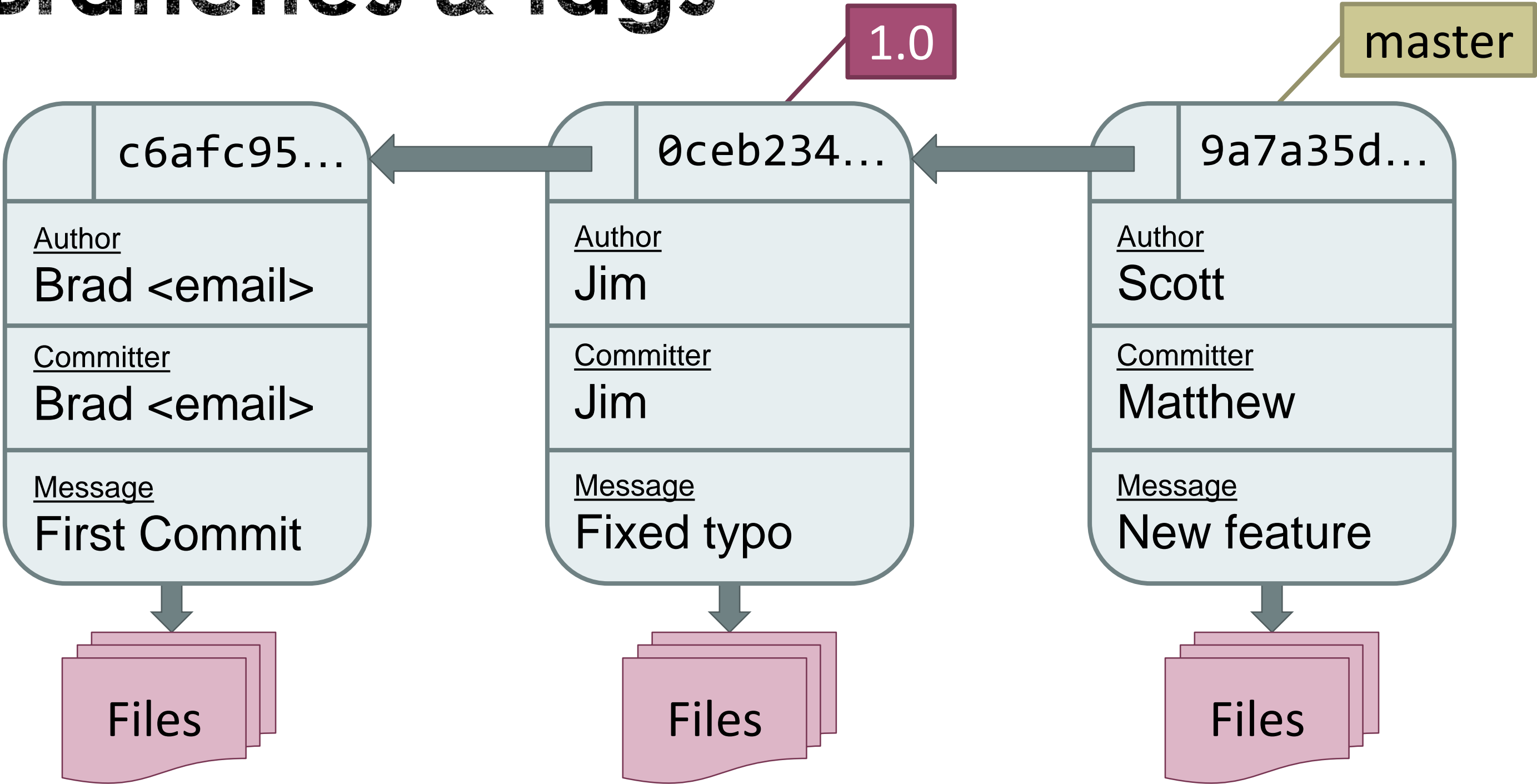


Branches & tags

- A branch in Git is a “sticky note” pointing at a commit
- `git checkout hash` points HEAD at a commit
`git checkout branch` points HEAD at a branch
- When you commit, HEAD always moves forward...
...and if HEAD is pointing at a branch, the sticky note moves, too
- Creating a new branch is just making a new sticky point, pointed (by default) to wherever HEAD is point to
- Tags are sticky notes that don't automatically move



Branches & tags

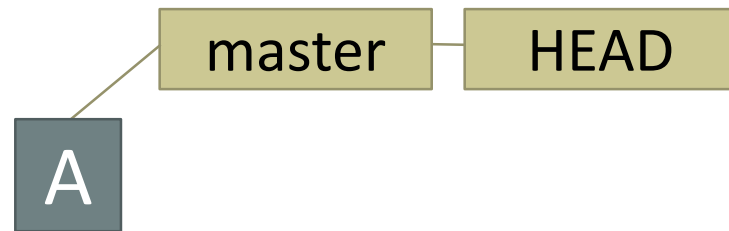




Branches



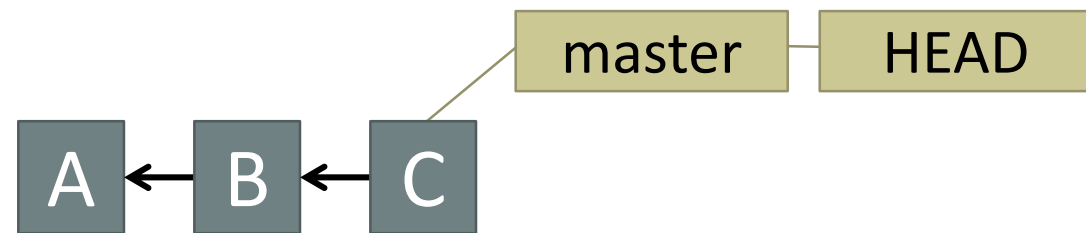
Branching



```
git commit -m "My first commit"
```



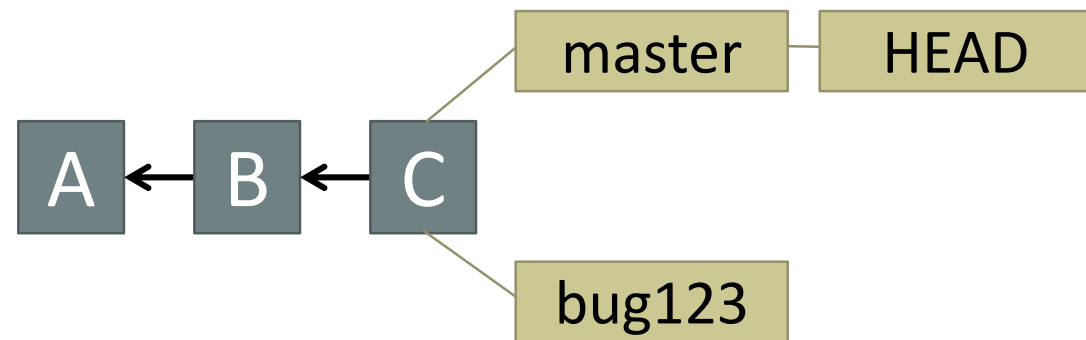
Branching



```
git commit (x2)
```



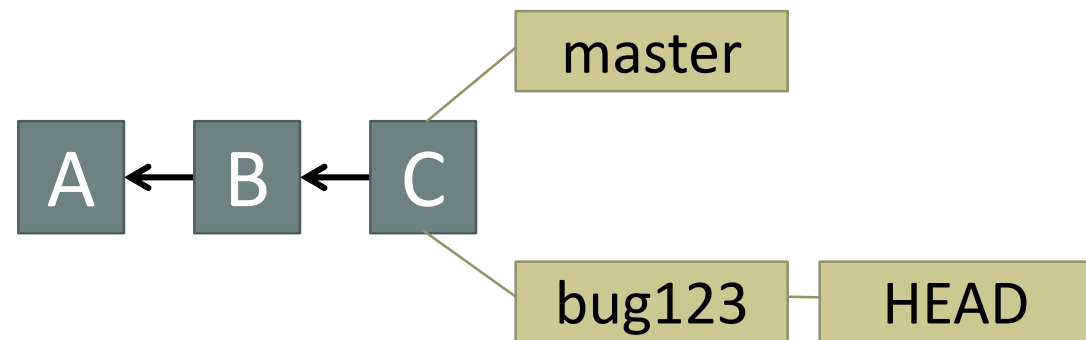
Branching



```
git branch bug123
```



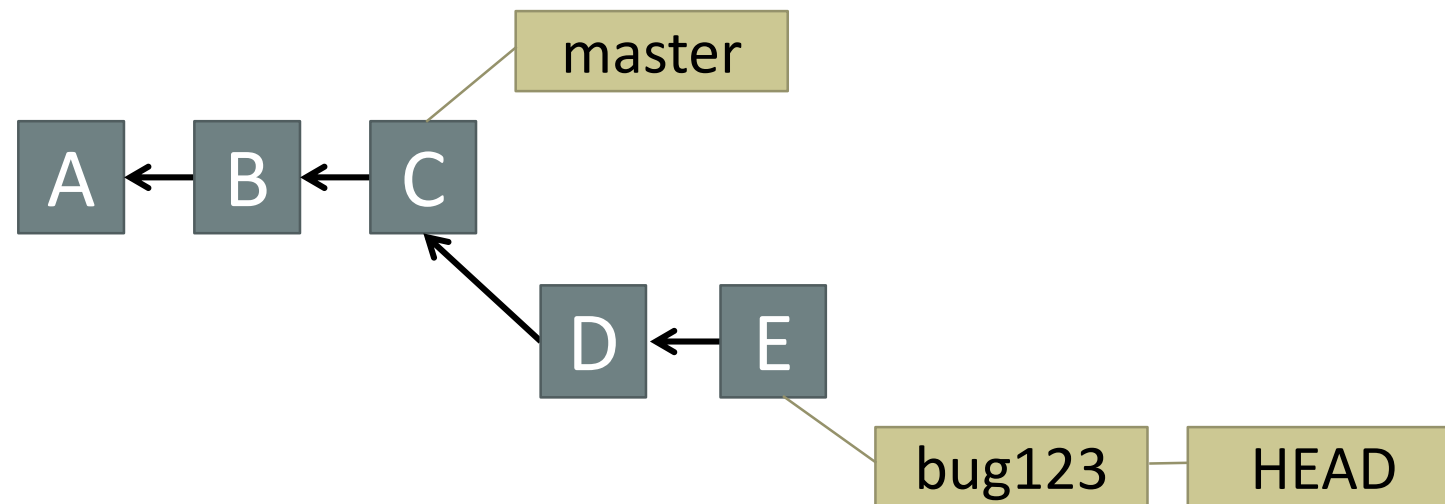
Branching



```
git checkout bug123
```



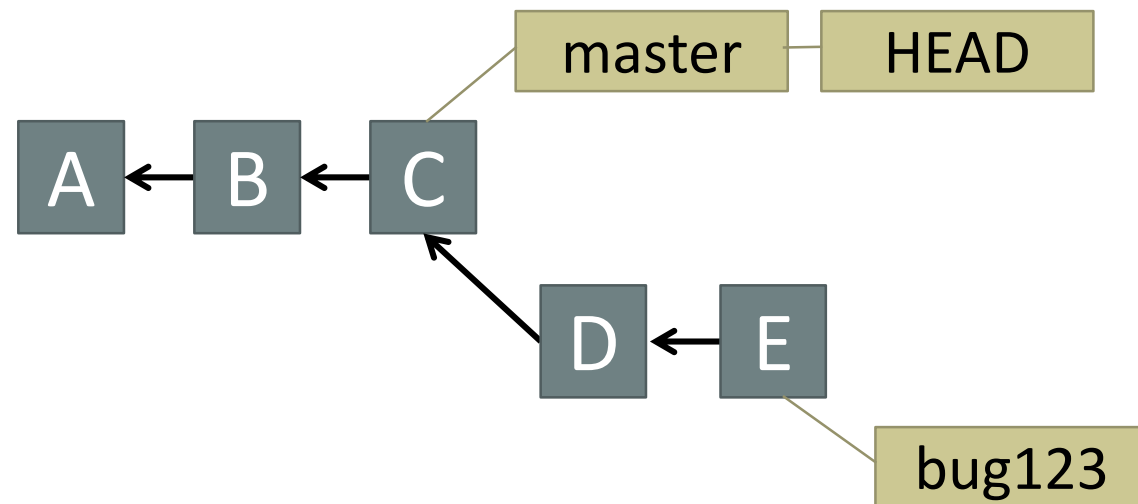
Branching



`git commit (x2)`



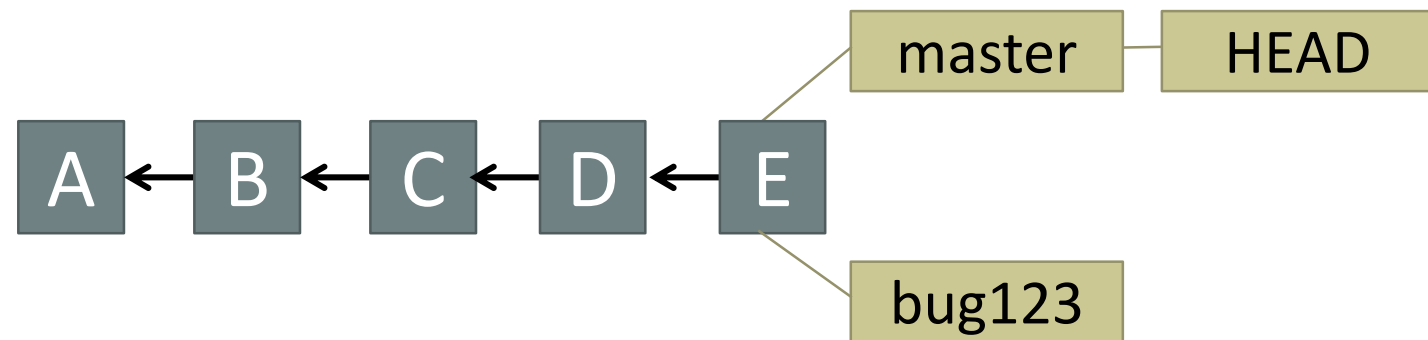
Branching



```
git checkout master
```



Branching

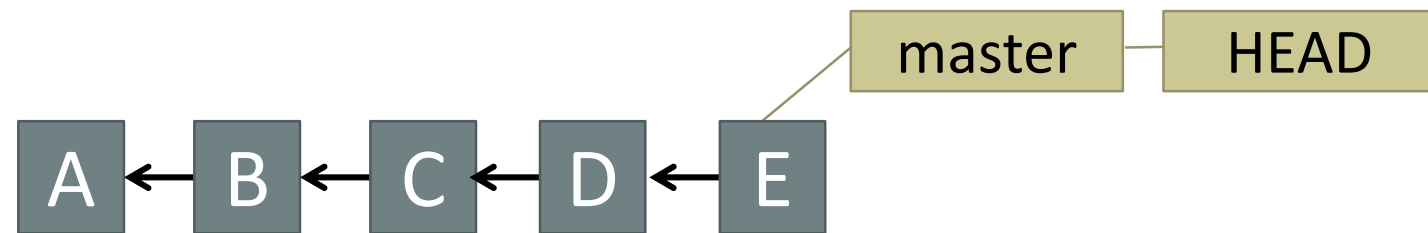


Fast-forward
Merge

```
git merge bug123
```



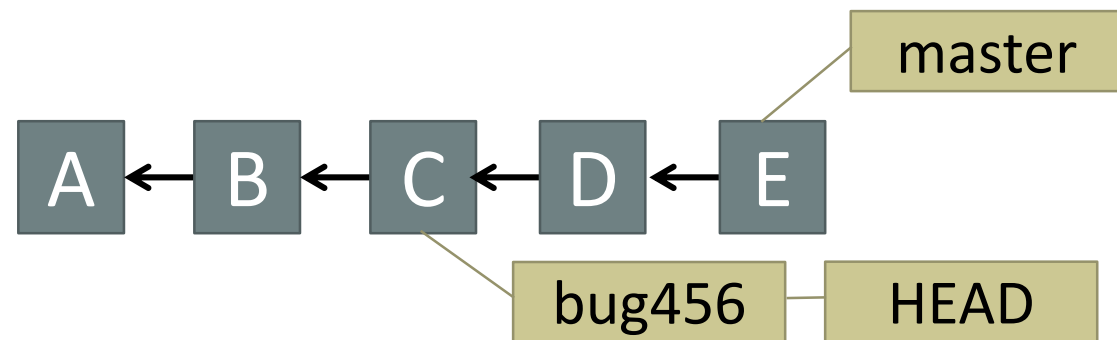
Branching



```
git branch -d bug123
```



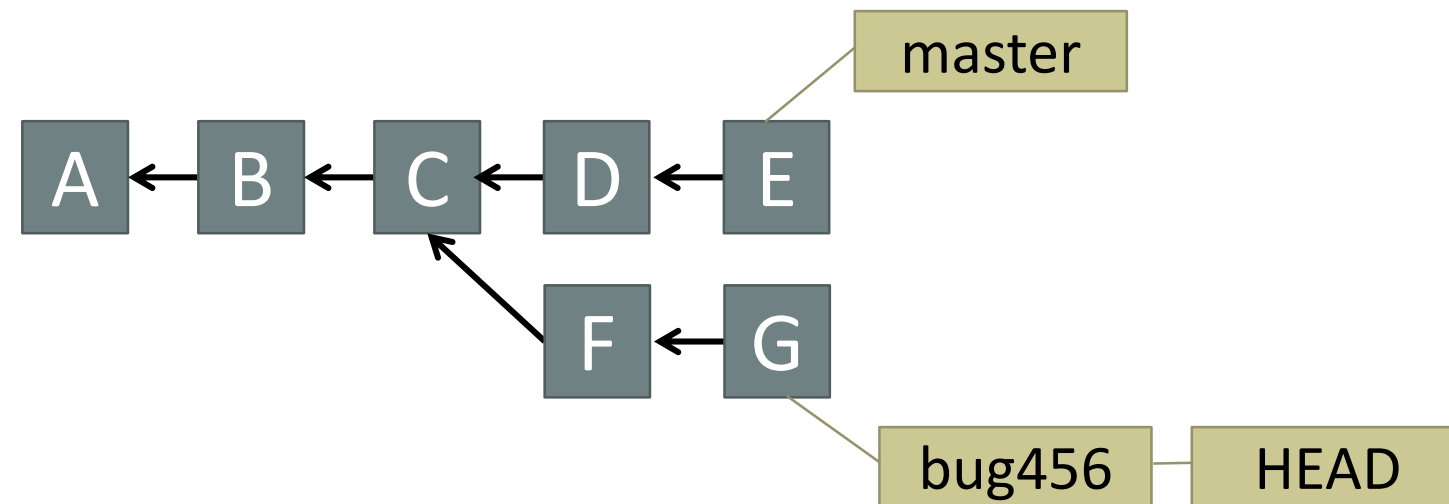
Branching



```
git checkout bug456
```



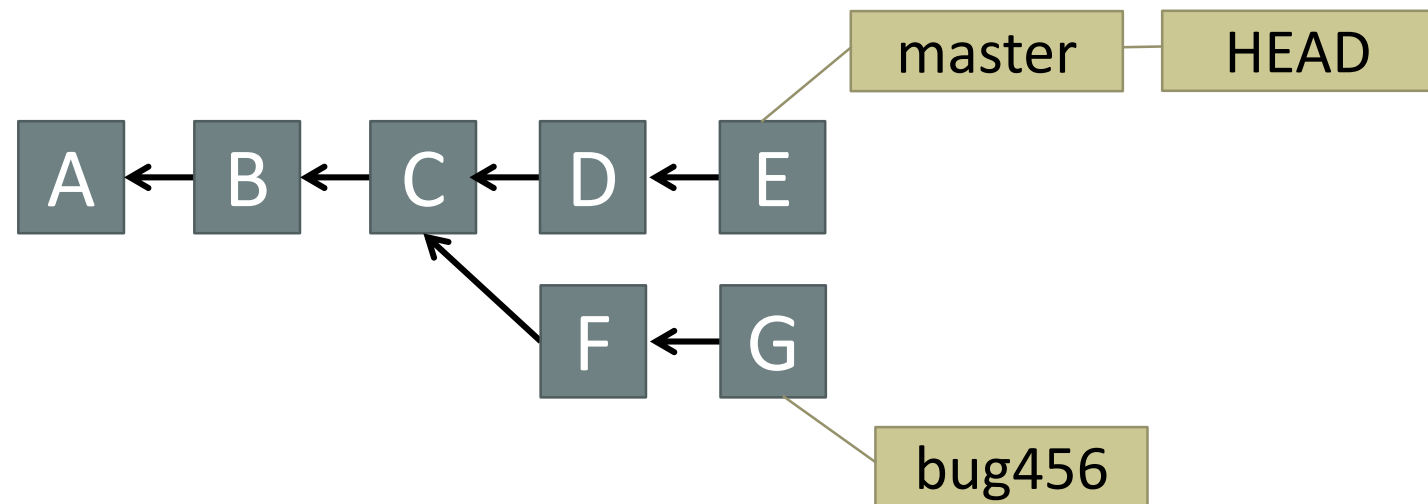
Branching



`git commit (x2)`



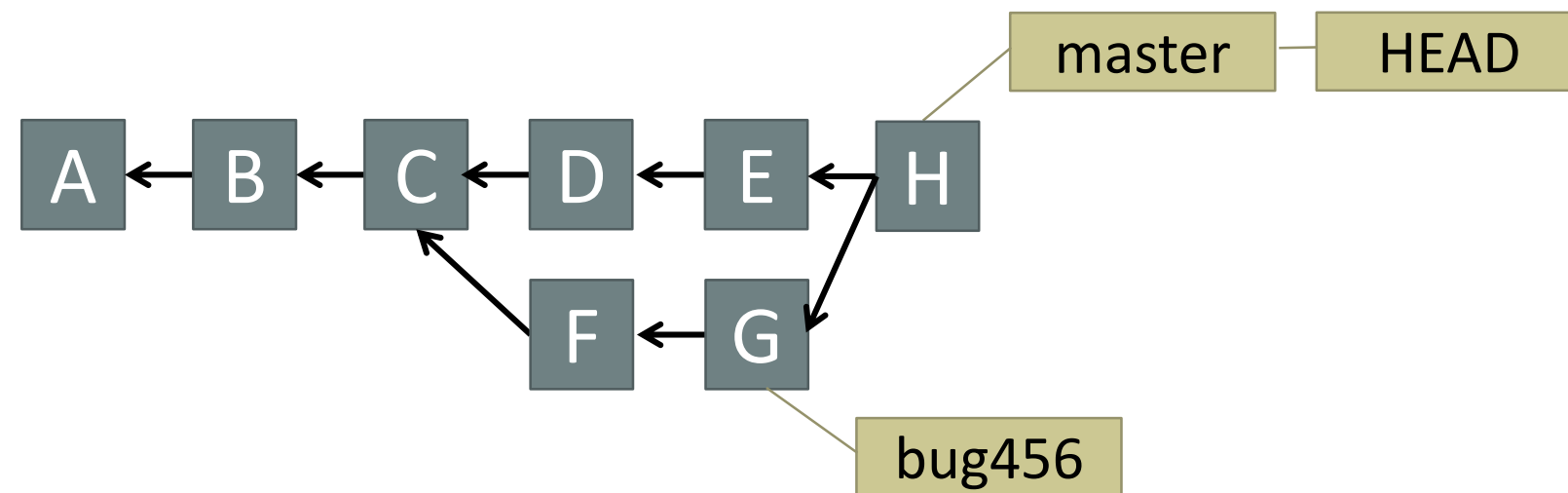
Branching



```
git checkout master
```



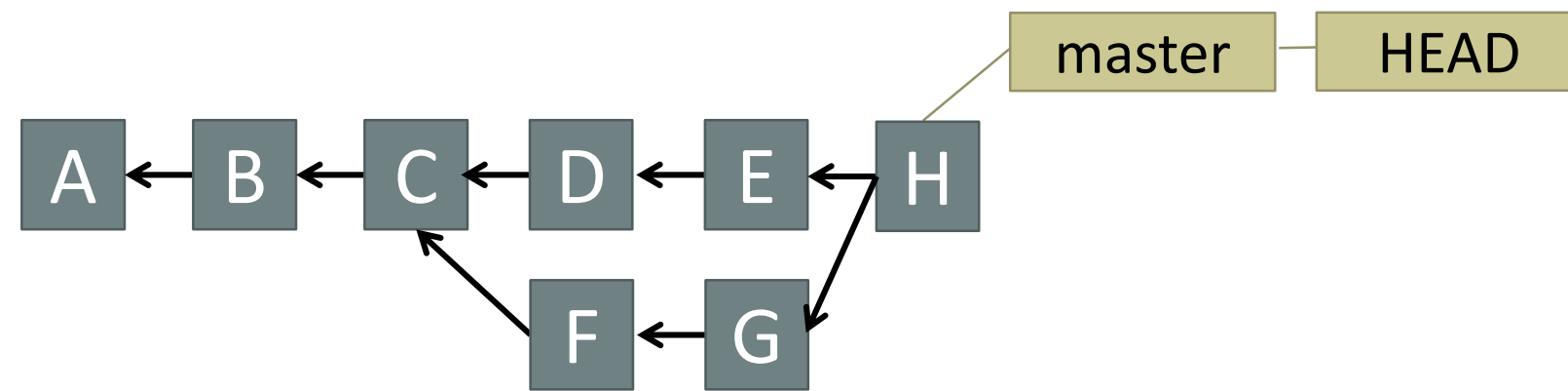
Branching



```
git merge bug456
```



Branching

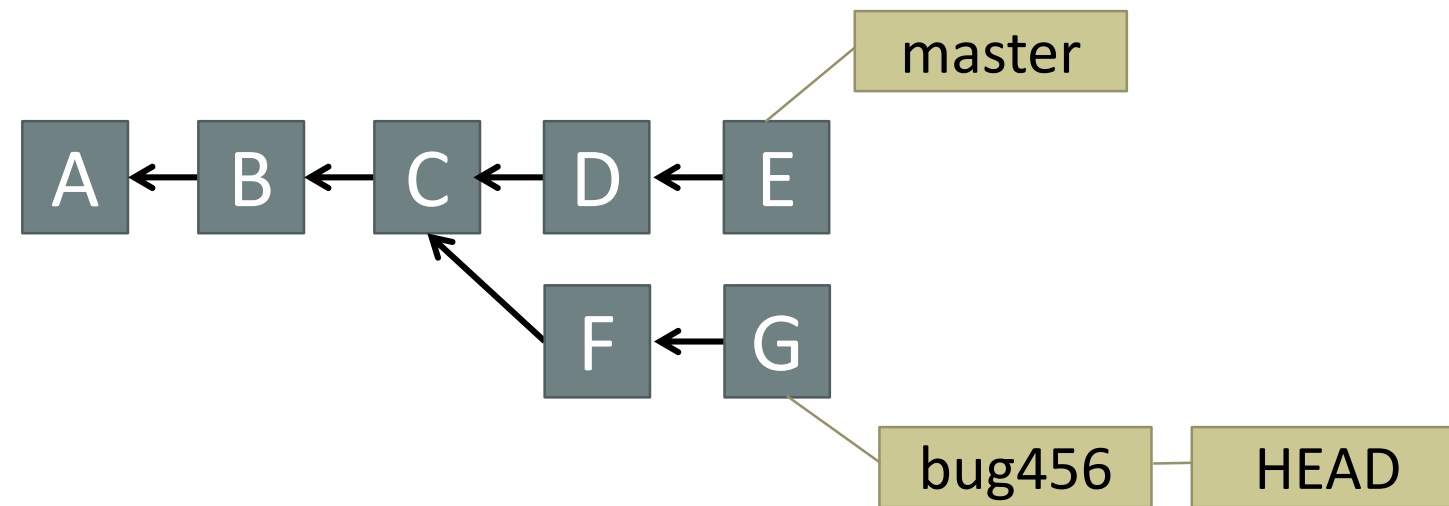


Merge
Workflow

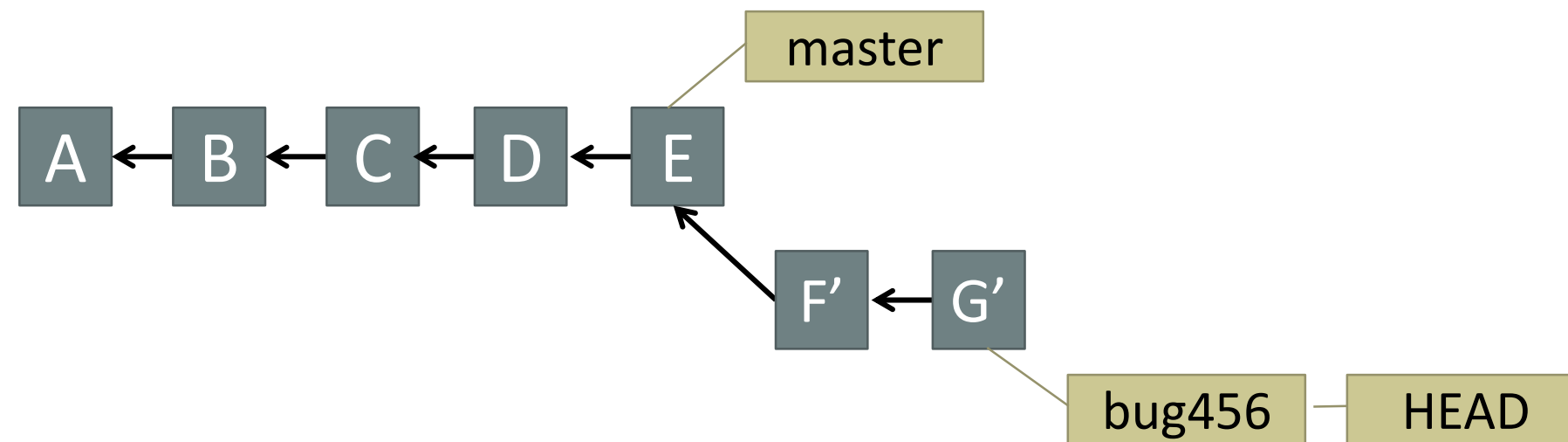
```
git branch -d bug456
```



Branching



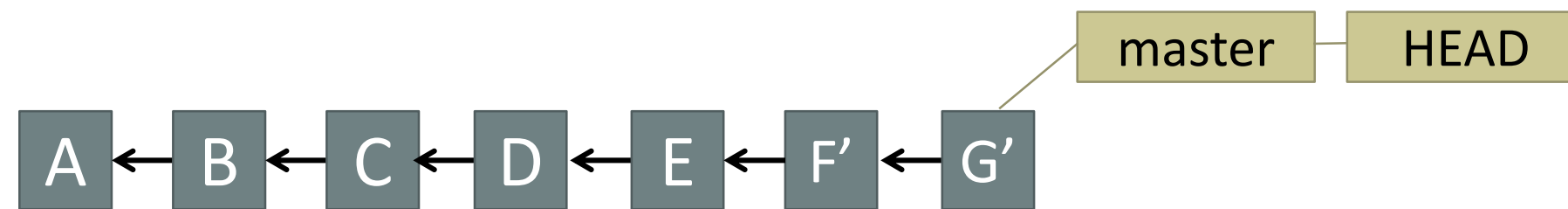
Branching



```
git rebase master
```



Branching



Rebase
Workflow

```
git checkout master / git merge bug456 / git branch -d bug456
```



Merge vs. rebase

- Merge preserves history, at the expense of extra commits
- Rebase rewrites history, but makes sharing branches painful
- Many tools will push you towards a merge workflow
- You can use whatever suits you best

You may have heard rebase is scary...

...but isn't that exactly what your centralized VCS does today?



Remotes

- Named repositories for sharing with
 - Configure as many as you like using `git remote`
 - By convention, `origin` is the name of the remote you cloned from
- Use `git fetch remote` to get new changes w/out applying them
- Use `git pull remote` to get **and** apply new changes
- Use `git push remote` to send new changes



Fetch & pull

- Both fetch and pull receive *all* changes for *all* remote branches
- Pull only applies changes to the *current* branch
 - Applies changes with merge workflow by default
 - Add `--rebase` to use rebase workflow instead
- Use `git branch -a` to see all branches, local and remote
- Use `git merge remotes/remote/branch` to merge in changes from the remote branch to your local branch
- Use `git checkout branch` with a remote branch name to create a local branch that tracks the remote (like `master` does by default)

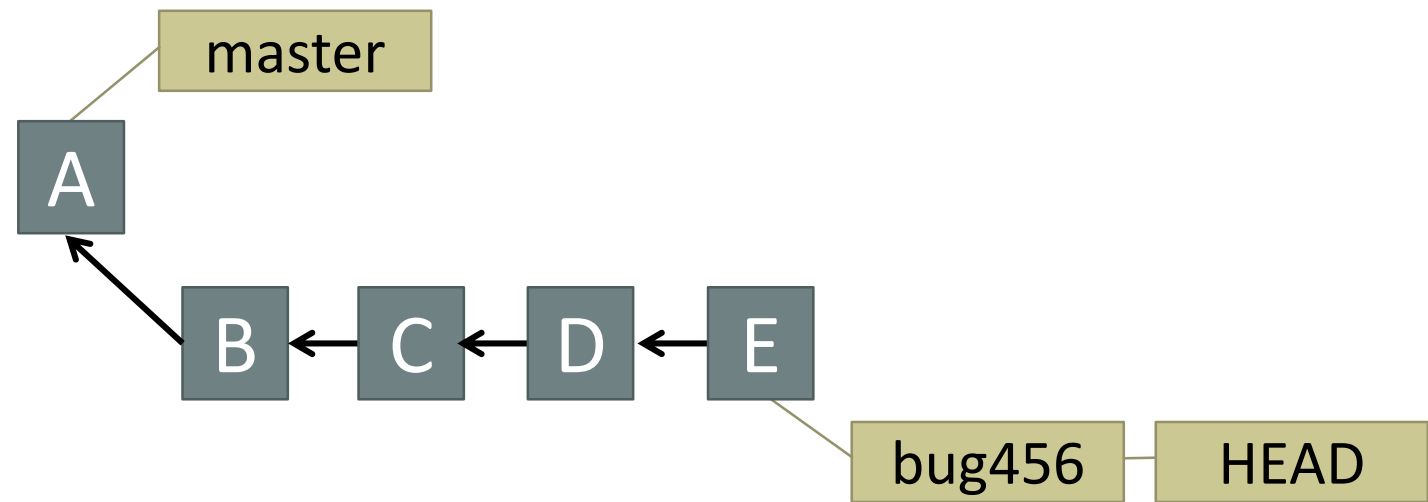


Push

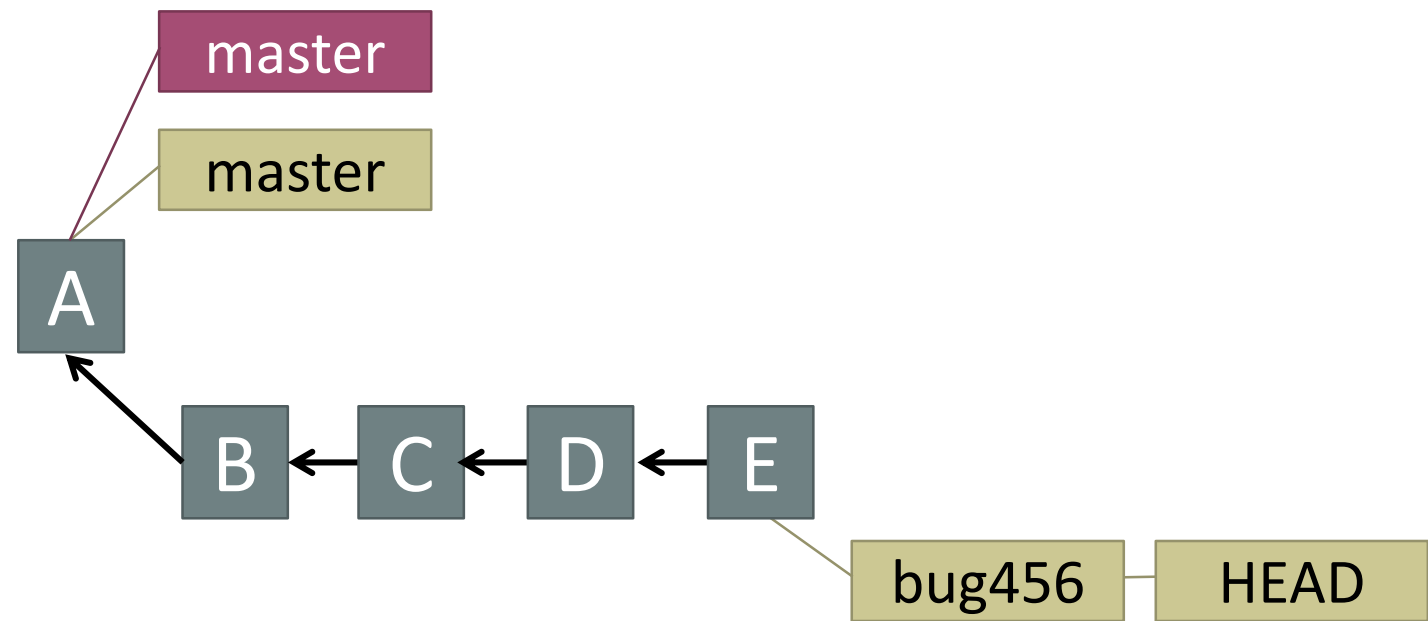
- Push sends changes from your local branch to the tracked remote branch.
- If you're out of date, you can't push before you pull
- It is typical to use the same name for local and remote branches which track each other (though not required)
- Use `git push --set-upstream remote` when first pushing a branch, to automatically set the tracking relationship



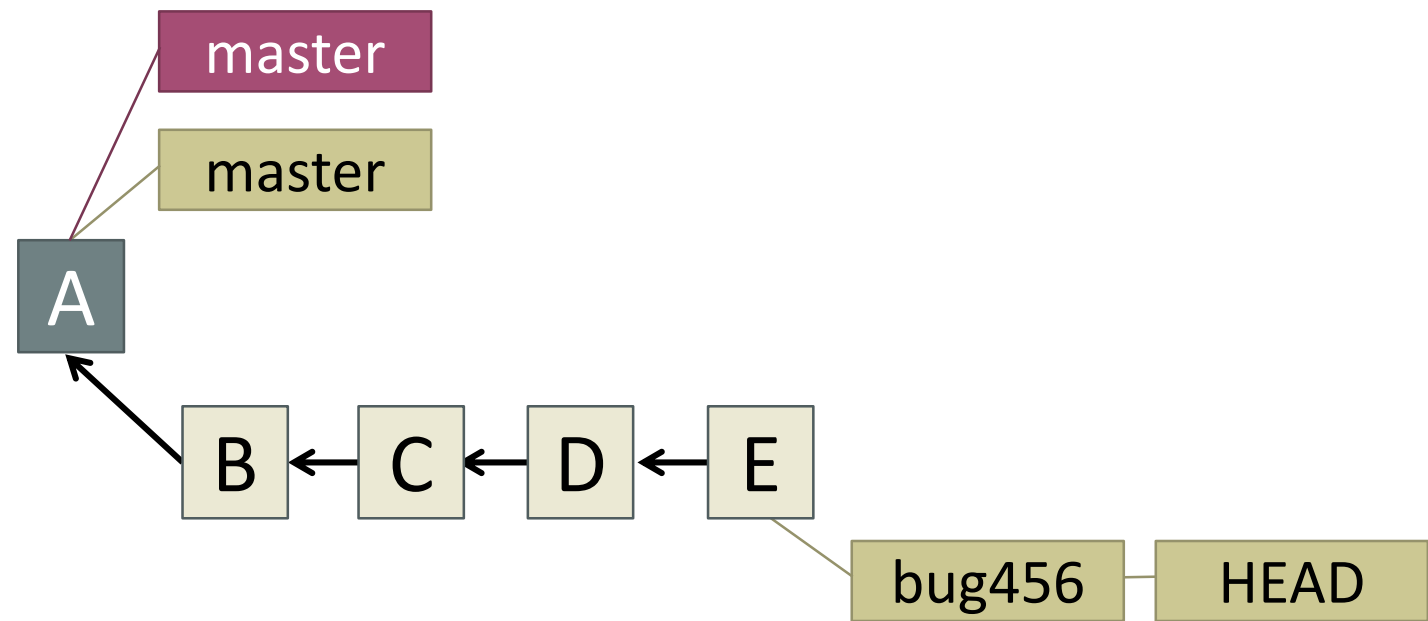
Branches (with remotes)



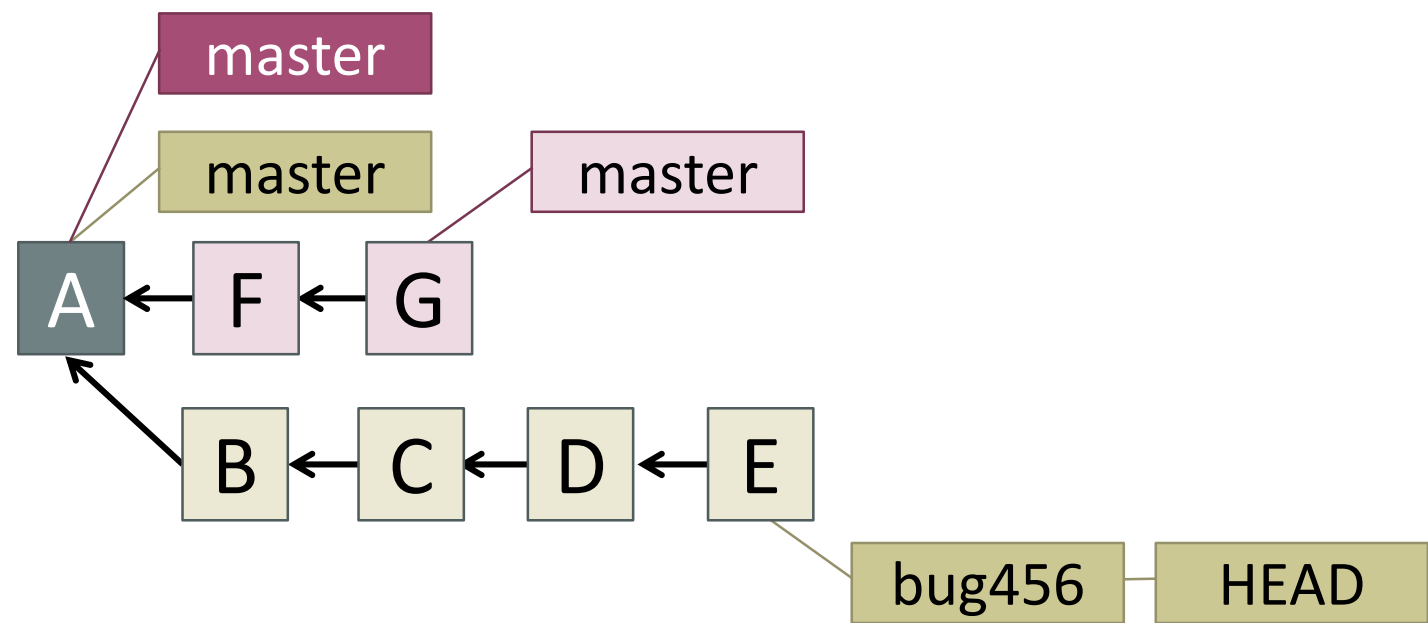
Branches (with remotes)



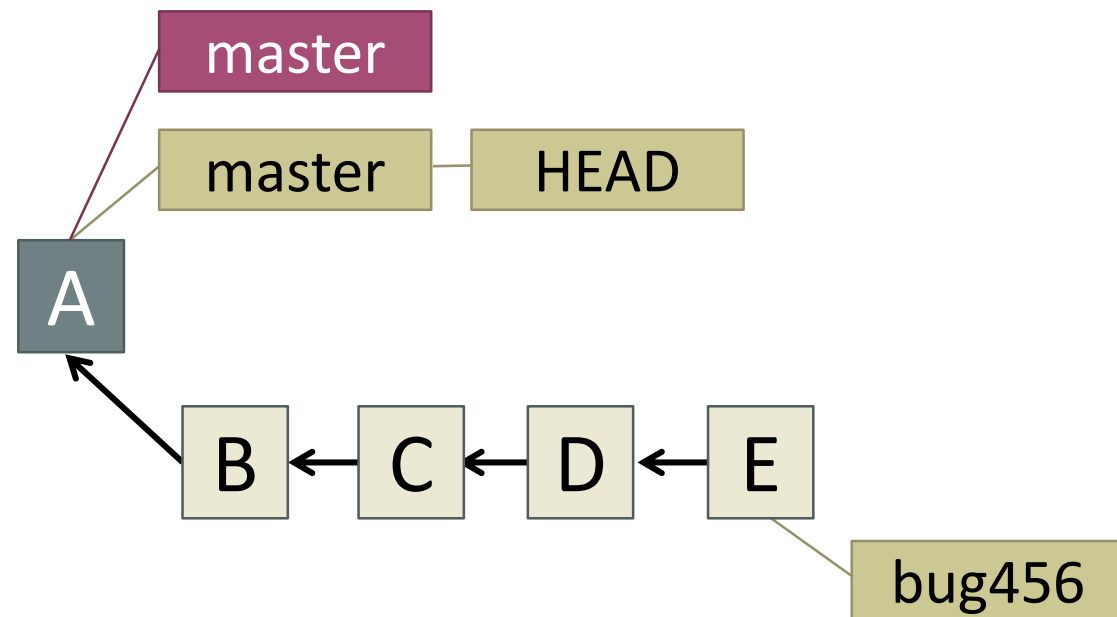
Branches (with remotes)



Branches (with remotes)



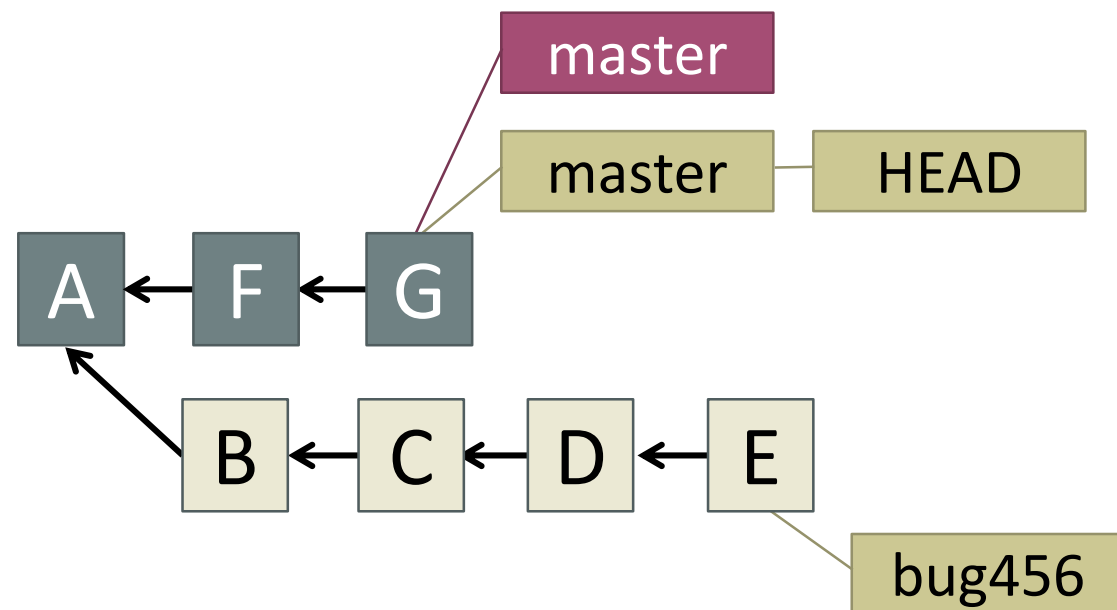
Merging and pushing remote



```
git checkout master
```



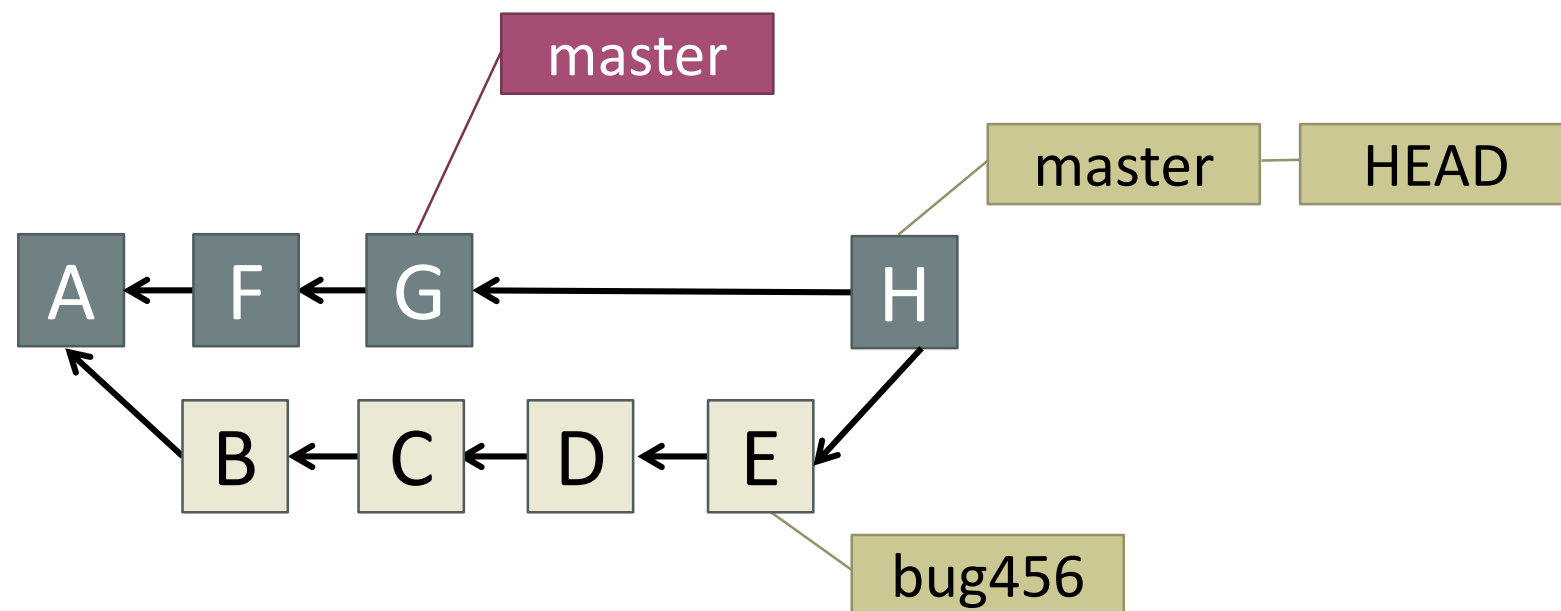
Merging and pushing remote



```
git pull origin
```



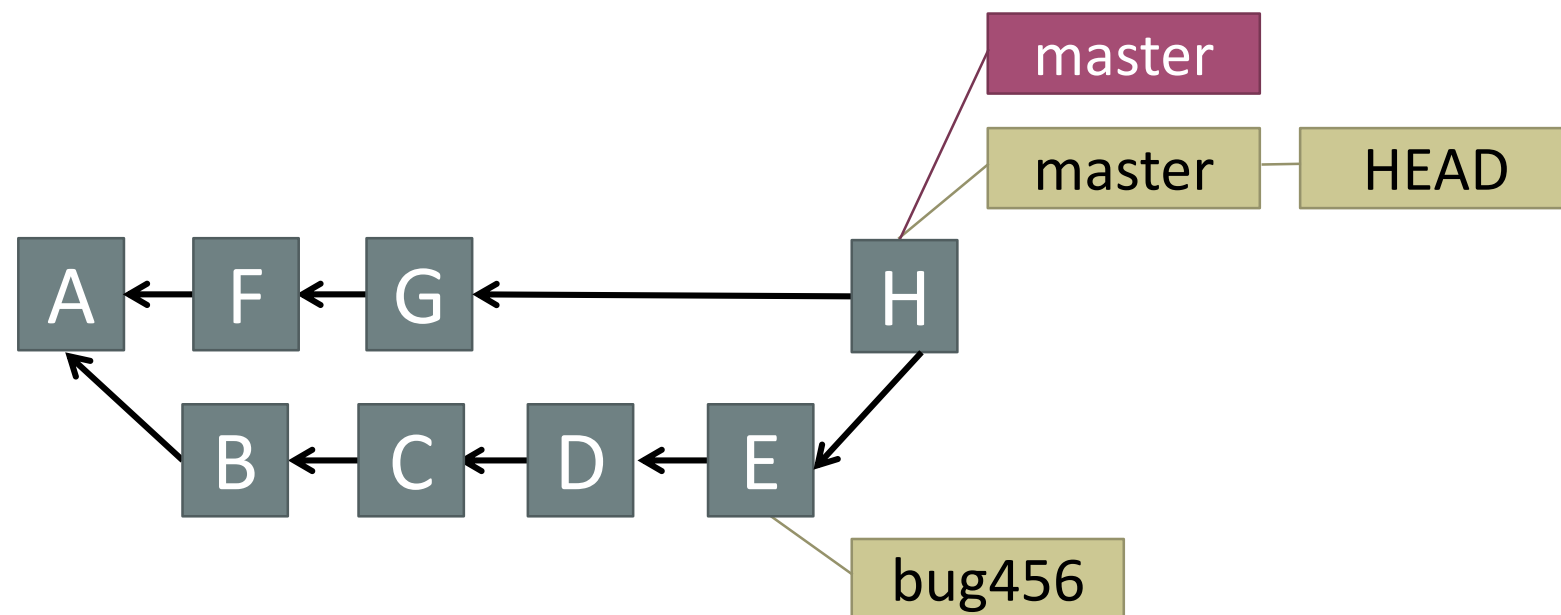
Merging and pushing remote



```
git merge bug 456
```



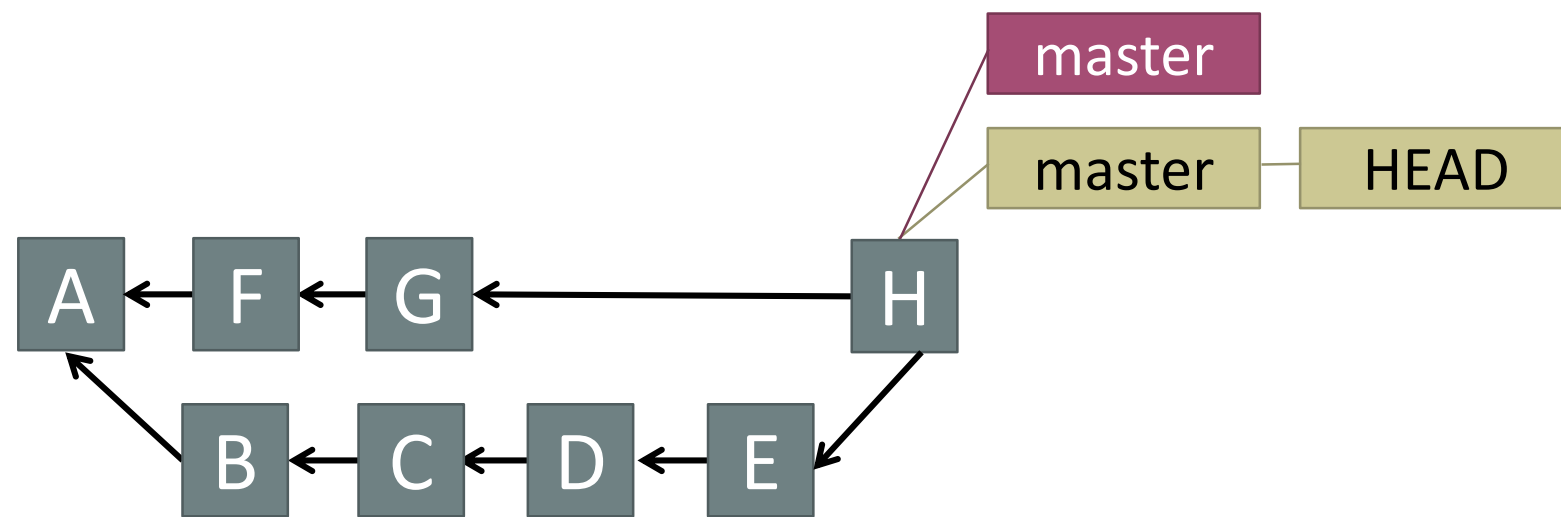
Merging and pushing remote



`git push origin`



Merging and pushing remote



```
git branch -d bug456
```

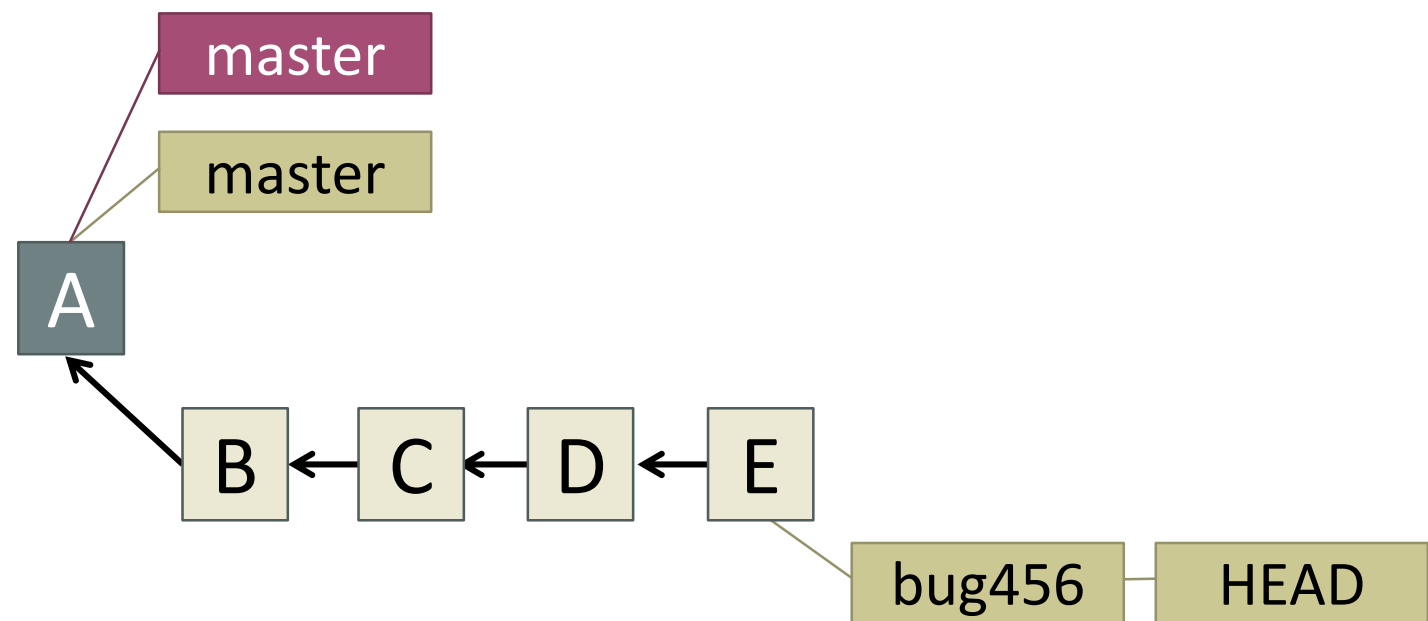


Remote branches

- Remember, a remote branch is just a branch
- The same merge vs. rebase strategies apply when merge between a remote and local branch as when merging between local branches
- Tracking relationships are local: your repository's relationships are local to you only, not shared



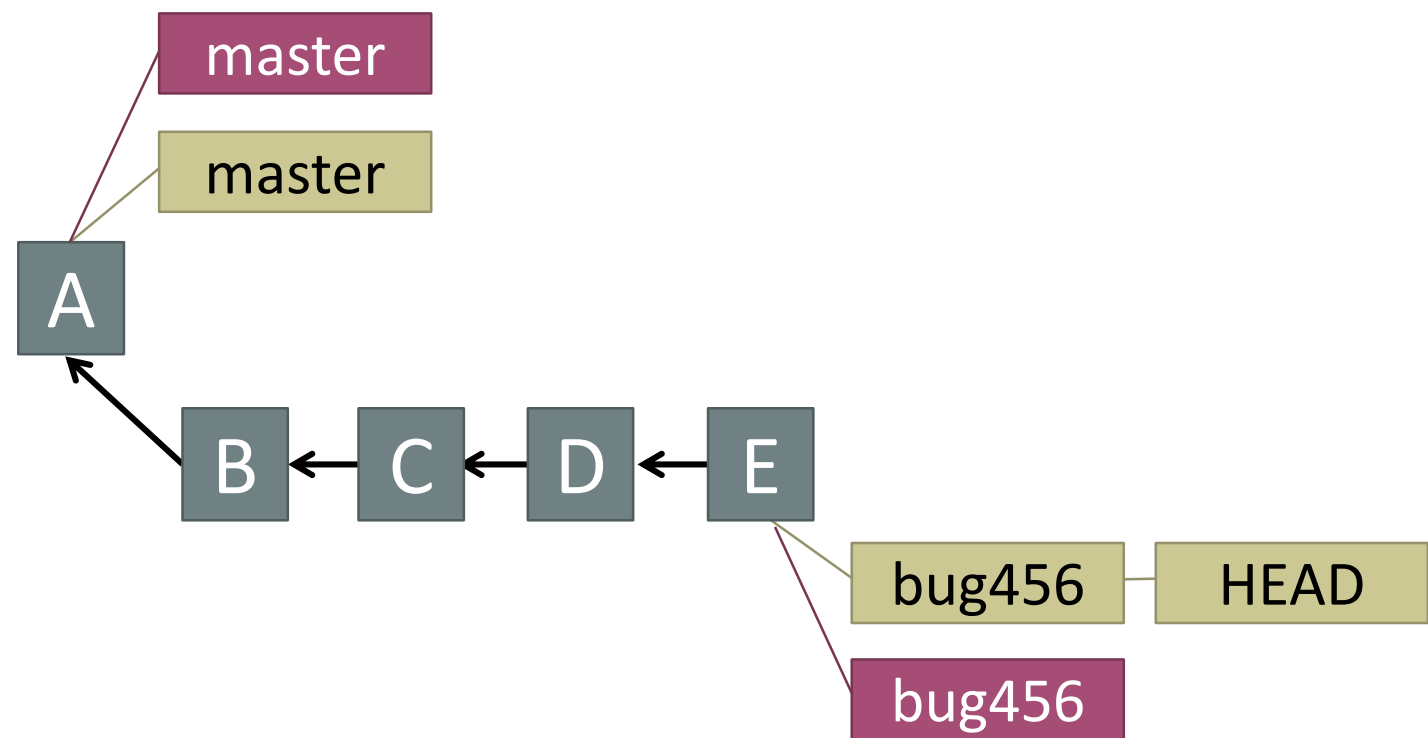
Using server branches



```
git checkout bug456
```



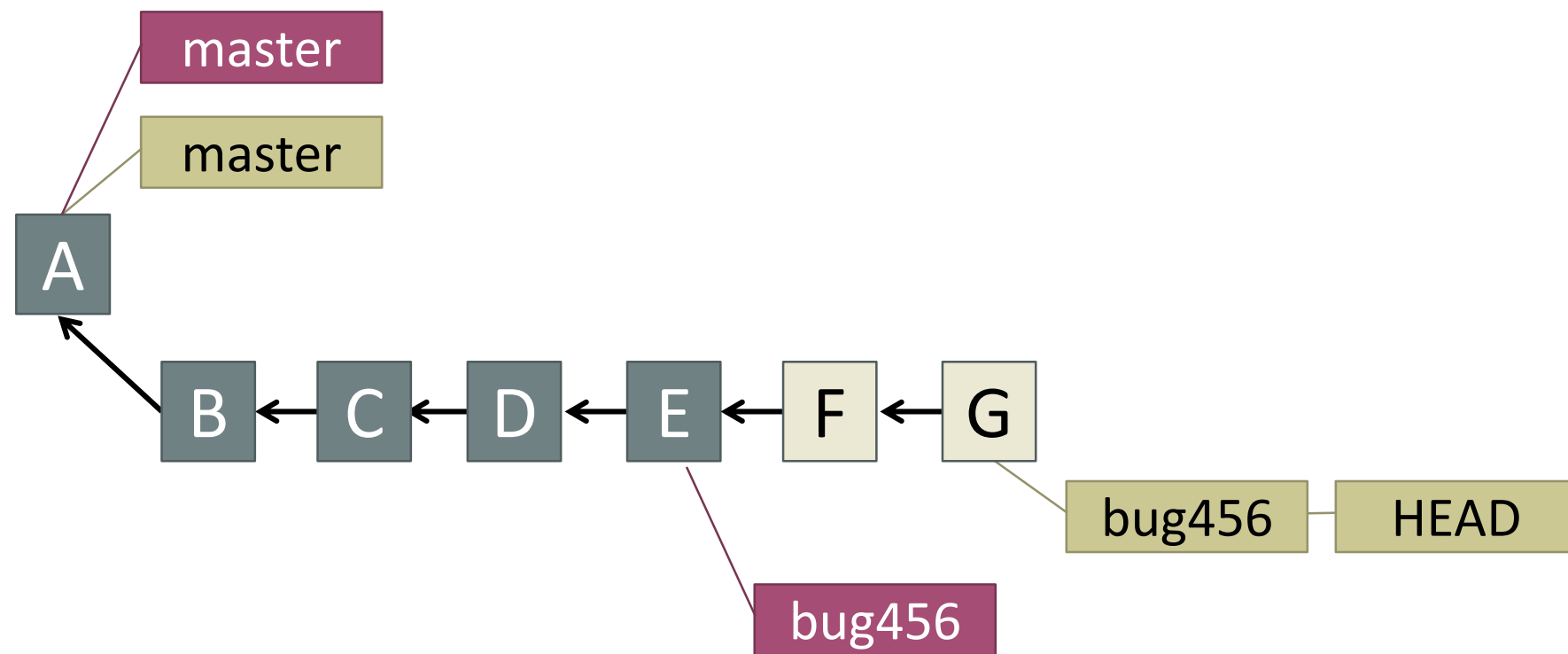
Using server branches



```
git push --set-upstream origin
```



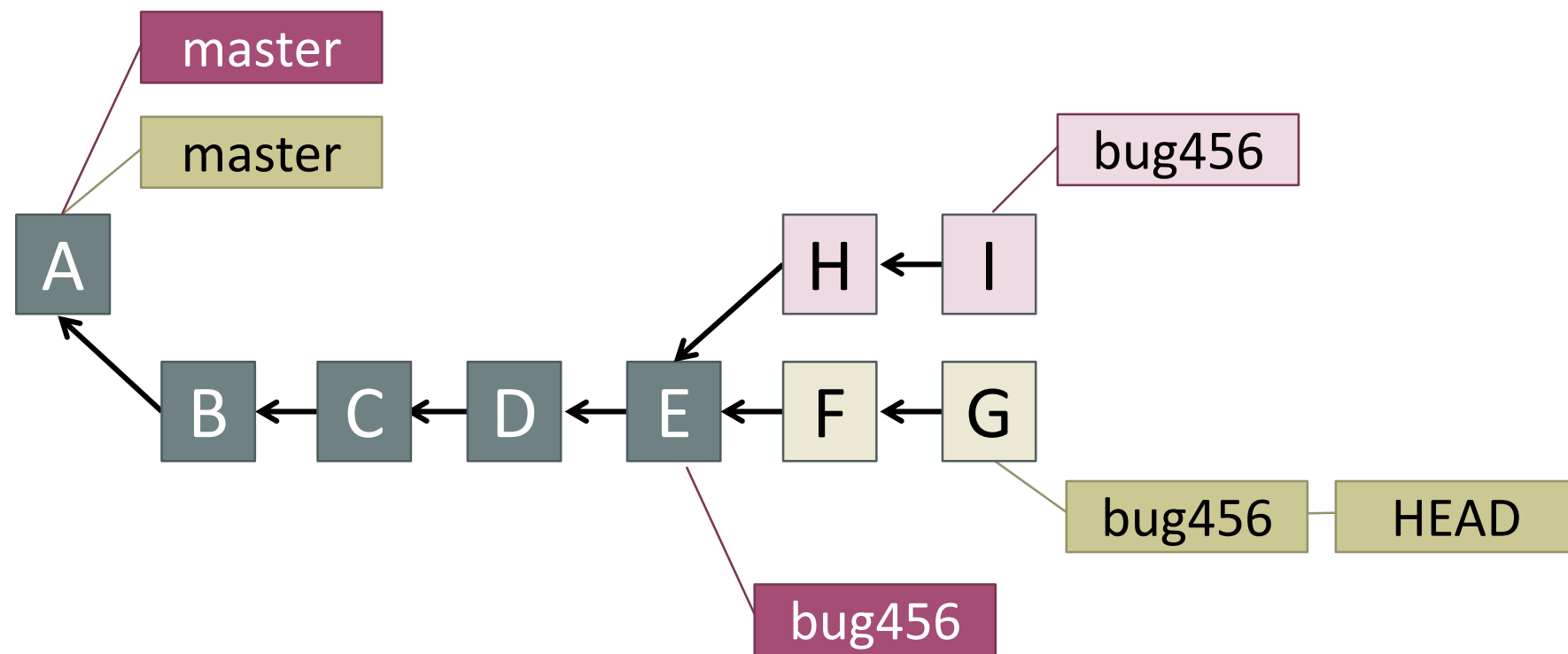
Using server branches



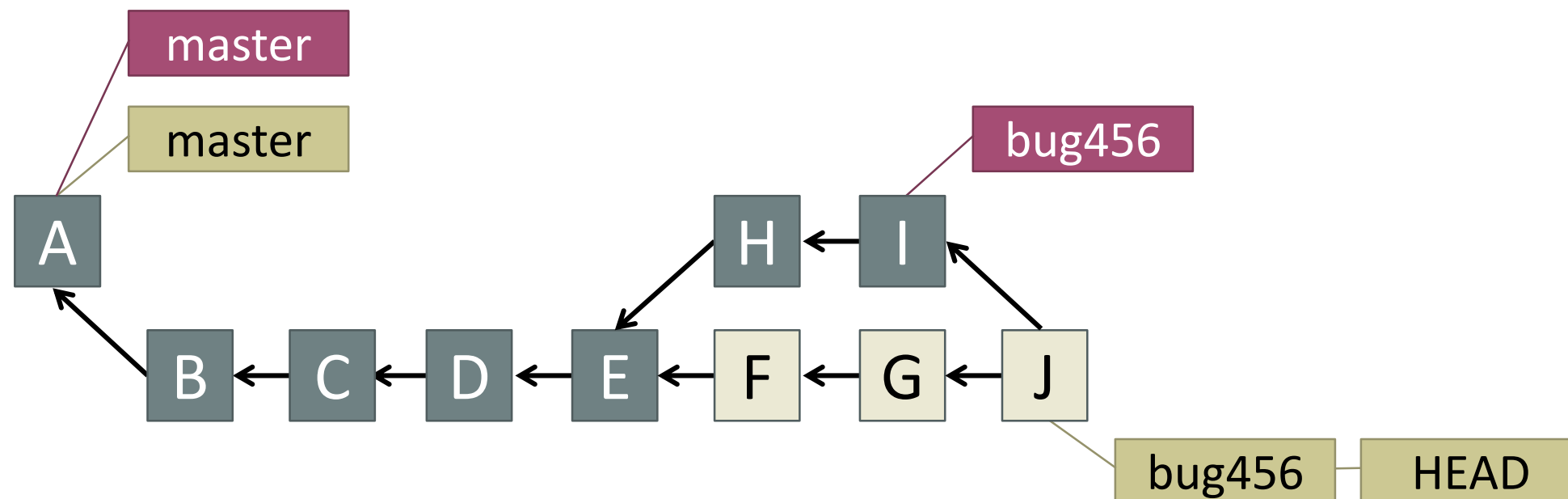
git commit (x2)



Using server branches



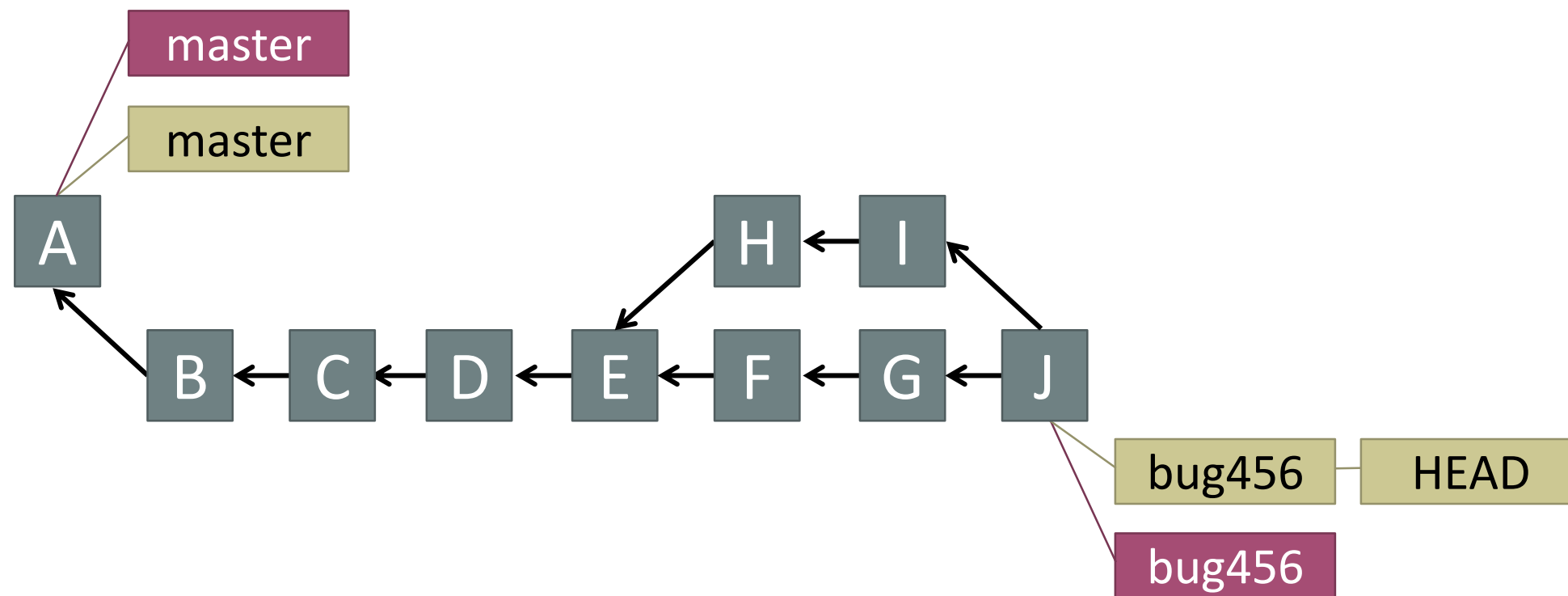
Using server branches



```
git pull origin
```



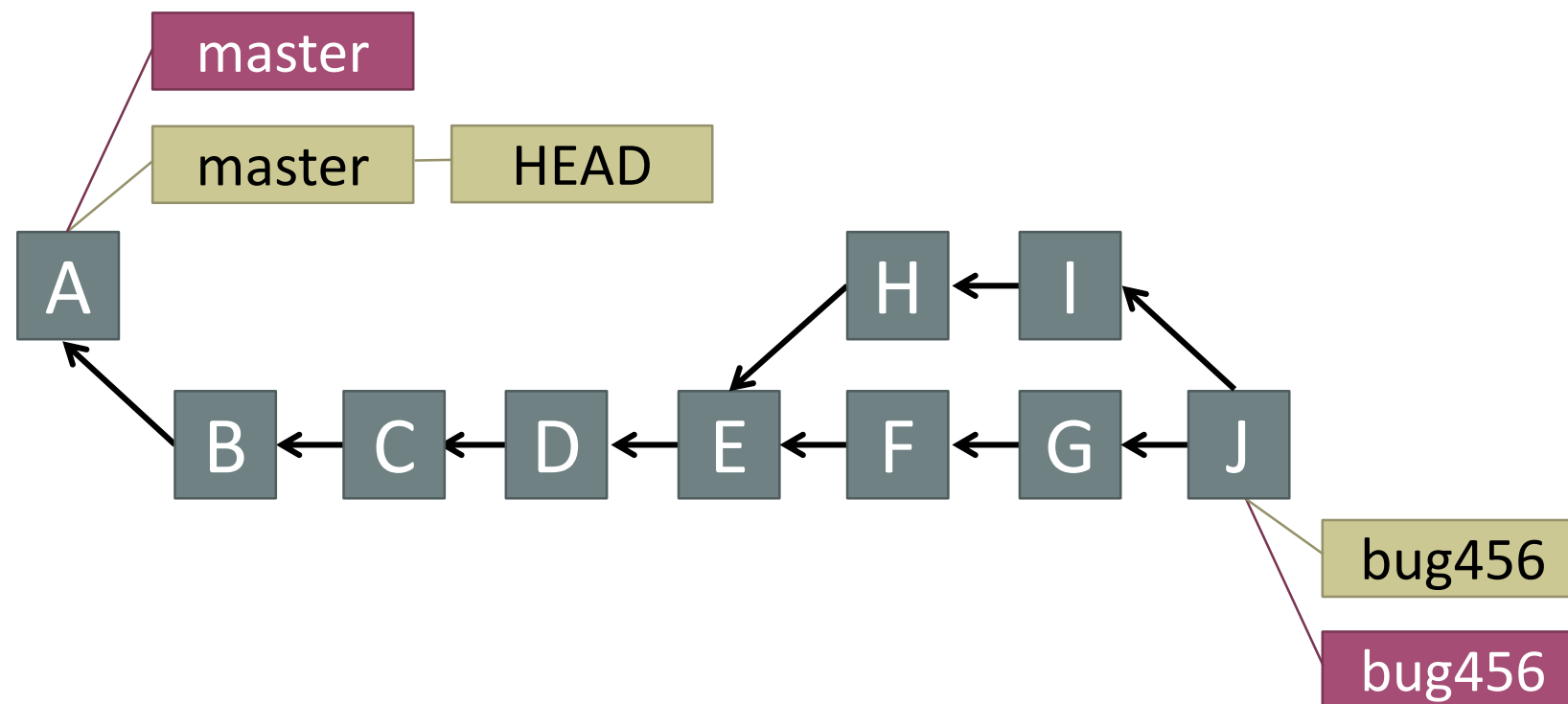
Using server branches



```
git push origin
```



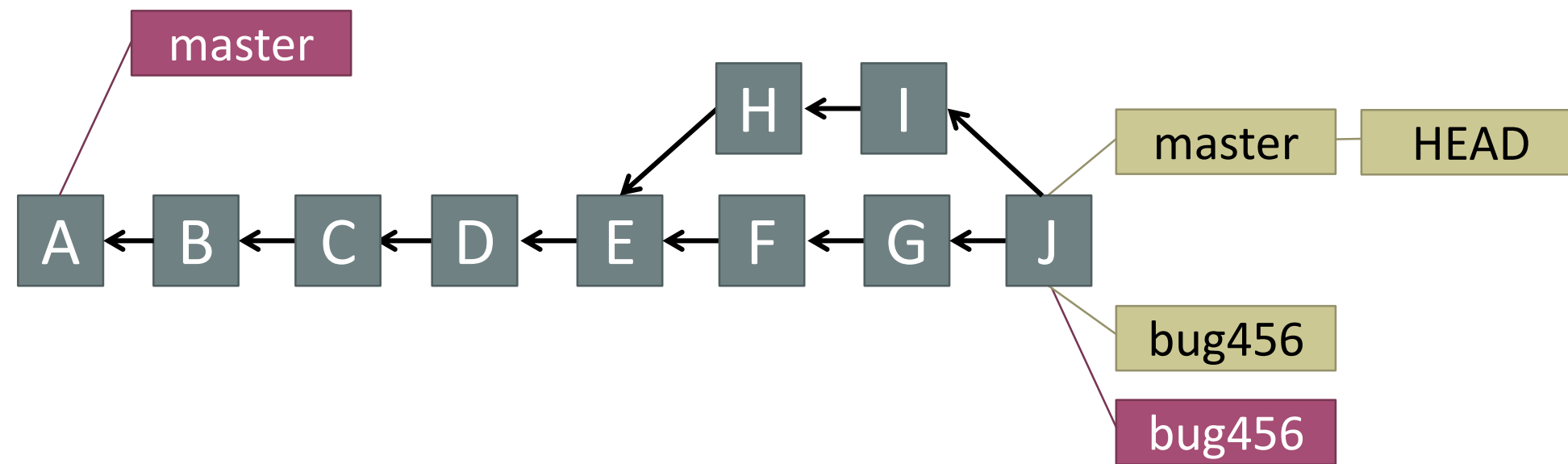
Using server branches



```
git checkout master
```



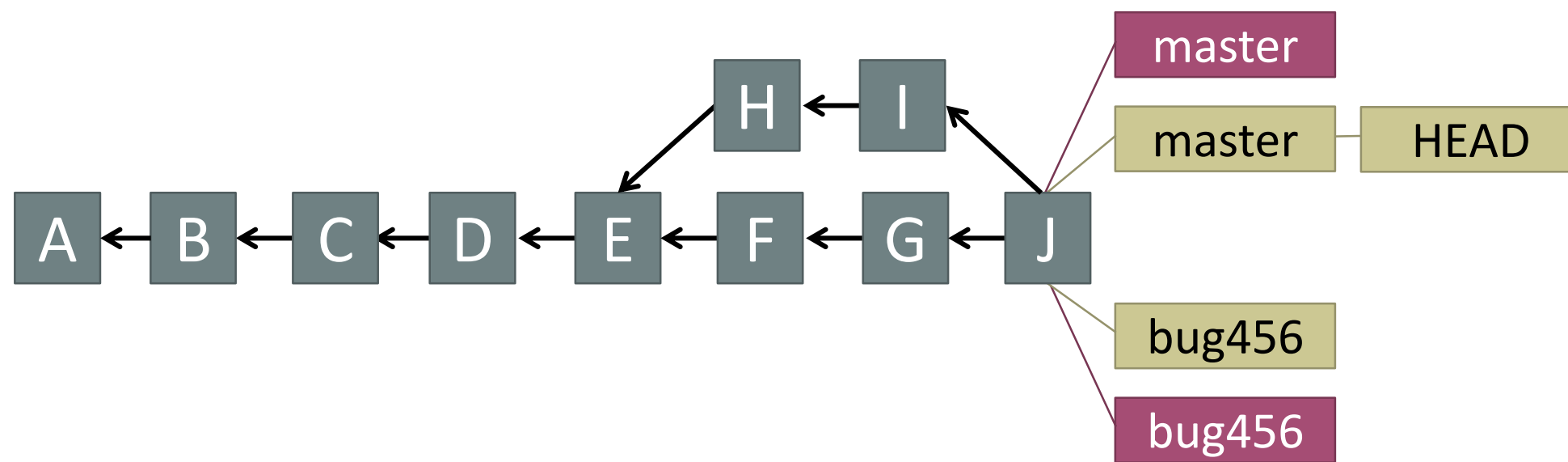
Using server branches



```
git merge bug456
```



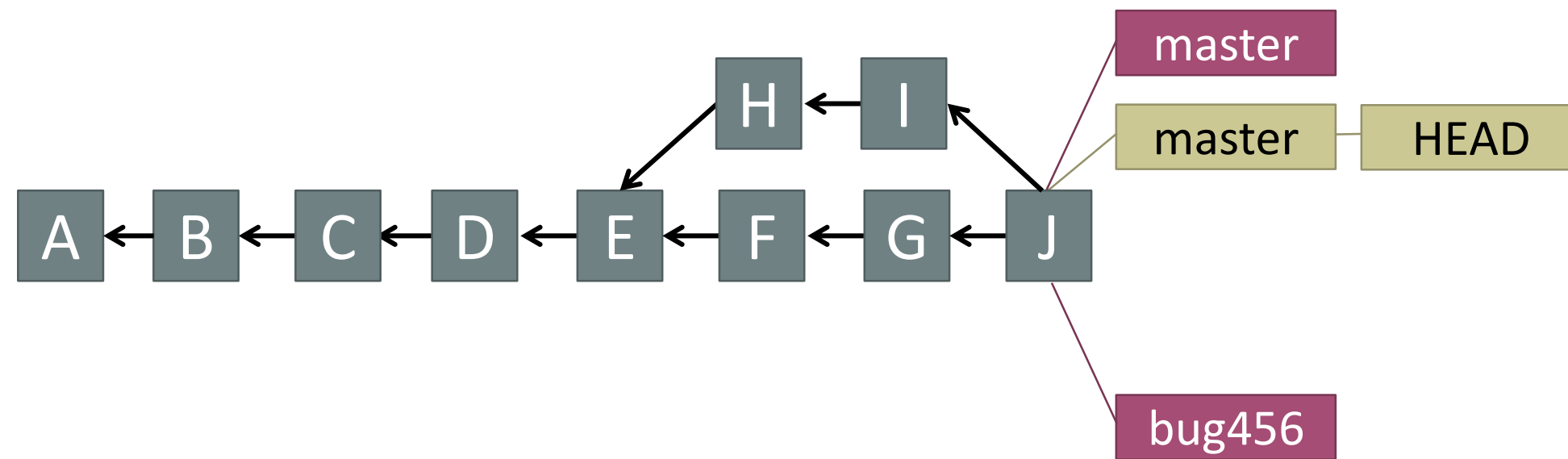
Using server branches



`git push origin`



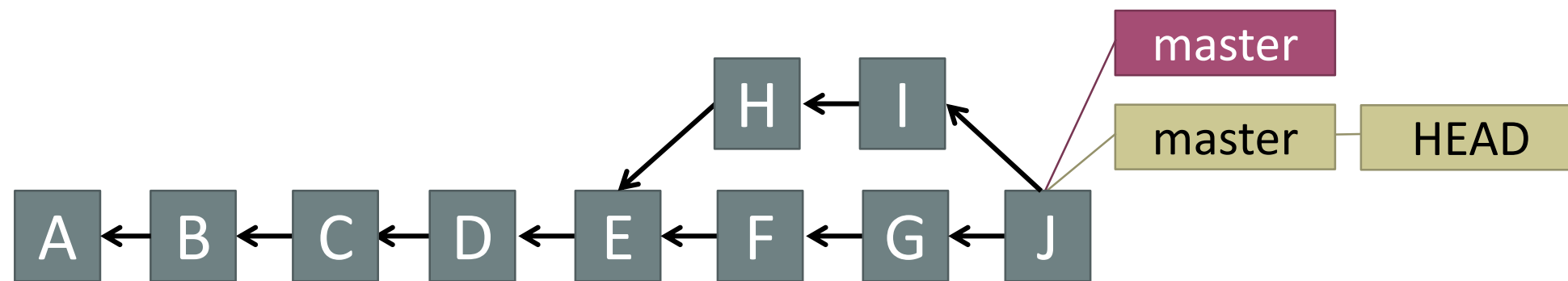
Using server branches



```
git branch -d bug456
```



Using server branches



```
git push origin :bug456
```





Tips & Tricks



Local branch strategies

- Short-lived local branches
 - The merges from master are simpler that way
 - Work on several things at once in several branches
 - Create, commit, merge, push, delete



Server branch strategies

- Short-lived server branches
 - Collaboration with co-workers
 - Sharing code between multiple PCs
 - Backing up in-progress code
- Long-lived server branches
 - Use to work in parallel on multiple versions
 - Same strategy as master: work in a local branch and integrate often



Thoughts on branching & tags

- The longer people are out of the main branch, the more painful the merges become – merge frequently
- Feature branches help prevent collisions, but delay integration
- Version branches have more churn, with frequent integration benefits
- Use tags when shipping; create branches only if you need to create an out-of-band patch
- Branches & tags are super cheap; don't be afraid to use them freely



Continuous integration

- Branches on the server represent work in progress
- Set up your CI server to automatically build all branches
- Push-button or automatic deployments from designated branches



CI

Run ...

refs/...master	#1.0.283	🌸 Tests passed: 671, ignored: 1; Test	No artifacts	Changes (2)	4m:07s left	Stop
Found...erTask	#1.0.282	✅ Tests passed: 5746, ignored: 32	No artifacts	(1)	13 minutes ago (4m:28s)	
Throw...Errors	#1.0.281	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	9 minutes ago (12m:21s)	
Creat...g-#439	#1.0.280	✅ Tests passed: 5661, ignored: 32	No artifacts	(1)	18 minutes ago (4m:38s)	
refs/...master	#1.0.277	✅ Tests passed: 5661, ignored: 32	No artifacts	(2)	19 minutes ago (18m:22s)	
Group...g-#431	#1.0.272	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	49 minutes ago (4m:53s)	
Start...rvices	#1.0.271	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	one hour ago (5m:20s)	
Accou...s-#420	#1.0.270	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	one hour ago (4m:57s)	
peer1dc	#1.0.267	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	16 hours ago (7m:05s)	
remov...onfigs	#1.0.262	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	21 hours ago (4m:48s)	
octop...pdates	#1.0.256	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	22 hours ago (14m:06s)	
423-N...ptions	#1.0.248	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	one day ago (4m:23s)	
Prici...Update	#1.0.242	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	one day ago (4m:36s)	
Deplo...rvices	#1.0.226	✅ Tests passed: 5630, ignored: 32	No artifacts	(1)	3 days ago (4m:29s)	
Couch...Server	#1.0.221	✅ Tests passed: 5724, ignored: 32	No artifacts	(1)	3 days ago (6m:04s)	
encryption	#1.0.203	✅ Tests passed: 5664, ignored: 32	No artifacts	(1)	4 days ago (5m:25s)	
octop...ackage	#1.0.196	✅ Tests passed: 5656, ignored: 32	No artifacts	(1)	4 days ago (4m:33s)	
373-H...nError	#1.0.172	❌ Tests failed: 1, passed: 5661, ignored: 32; process exited with code 1	No artifacts	(1)	4 days ago (4m:25s)	
Deplo...ations	#1.0.127	✅ Tests passed: 5653, ignored: 32	No artifacts	(2)	5 days ago (13m:12s)	

Release

Pending (159)

Run ...

refs/...master	#1.0.754	✅ Tests passed: 5664, ignored: 32	Artifacts	Changes (3)	3 hours ago (22m:49s)	
----------------	----------	-----------------------------------	-----------	-------------	-----------------------	--



Code reviews

- What do you mean you aren't pairing? 😊
- Use server branches to share and contribute
- Some hosts offer code review facilities via pull requests



Being evaluated

Agat requested [Agat/BuildManagerCompiledView0Render:master](#) to be merged into [aspnetwebstack:master](#) on Sun Jan 27, 2013 at 6:58 AM

Assigned to: Unassigned

Commits: [08bd0e534186](#), [4bbcb80d5c7f](#), [5dbfcdd0ba01](#), [bd158d4653d3](#), [deaea4ff0d5a](#), [f404a7416093](#)

Comments (5)

Changes (3)

edit [src/System.Web.Mvc/BuildManagerCompiledView.cs \(View full diff\)](#)

edit [src/System.Web.WebPages/WebPageBase.cs \(View full diff\)](#)

edit [src/System.Web.WebPages/WebPageExecutingBase.cs \(View full diff\)](#)

src/System.Web.Mvc/BuildManagerCompiledView.cs

[Back to changes](#) | [View full diff](#)

```
55
56     public string ViewPath { get; protected set; }
57
58     public void Render(ViewContext viewContext, TextWriter writer)
59     public virtual void Render(ViewContext viewContext, TextWriter writer)
```

ADD NOTE

CANCEL

```
59     {
60         if (viewContext == null)
61         {
```



```
...  ... @@ -6,8 +6,9 @@
6    6    using System.Net;
7    7    using System.Threading;
8    8    using System.Threading.Tasks;
9    -using Newtonsoft.Json.Linq;
10   9    using Microsoft.AspNet.SignalR.Client.Http;
11   10   +using Newtonsoft.Json;
12   11   +using Newtonsoft.Json.Linq;
13   12
14   13   namespace Microsoft.AspNet.SignalR.Client
15   14   {
16   ...  ... @@ -37,5 +38,8 @@ public interface IConnection
37   38       void OnReconnecting();
38   39       void OnReconnected();
39   40       void PrepareRequest(IRequest request);
40   41   +
41   42   +     JsonSerializer JsonSerializer { get; set; }
42   43   +     string SerializeObject(object value);
```

5

**davidfowl** repo collab

5 days ago

This should not be on the interface.

**bjarteskogoy**

5 days ago

I did this because the HubProxy is referencing the interface and needs to serialize to Json on line 153 in HubProxy.cs.
The JsonSerializer must be set when testing the HubProxy using a mocked up IConnection. Maybe it can be solved in another way.



Resources

<http://www.git-scm.com/>

Git's home on the internet

<http://www.git-scm.com/book>

Online book "Pro Git"

<http://gitref.org/>

Online reference for all git commands (don't forget about "git help"!)

<http://try.github.com/>

Try git in your browser, with step-by-step instructions

<http://gitready.com/>

Tons of tips and tricks for Git users of all experience levels



Tools

<http://code.google.com/p/msysgit>

Git command line for Windows (includes git-bash)

<http://code.google.com/p/tortoisegit>

TortoiseGit (with TortoiseMerge)

<http://github.com/dahlbyk/posh-git>

Posh-Git (for PowerShell users)

<http://www.jetbrains.com/teamcity>

TeamCity Continuous Integration



Thank You!

@bradwilson

<http://bradwilson.typepad.com/>

