Version Control & GIT

Presented By:

$man git

NAME

git - the stupid content tracker

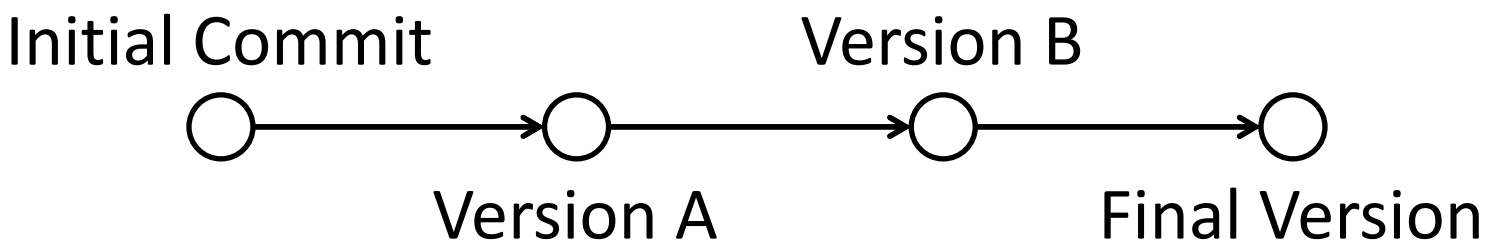
DESCRIPTION

GIT is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

Theory (Simplified)

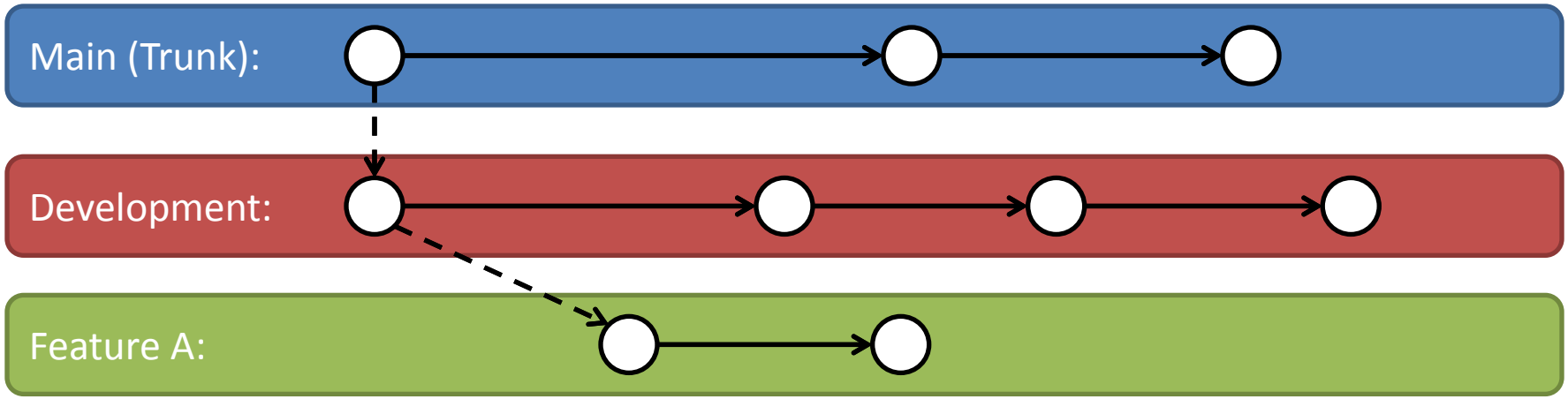
A Simplified Summary of Commit, Branches, and Merging

Commits



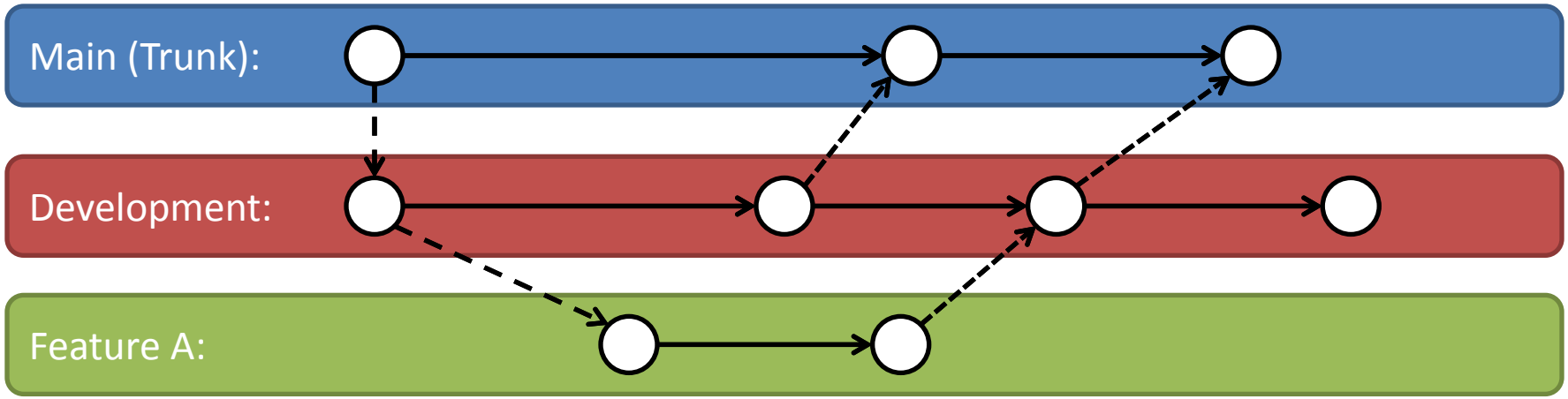
- Commits
  - Represented as nodes on a graph.
  - Symbolize an instance (version) of a project's files.
  - Can add, remove, or modify a file (or multiple files).
- (Implementation) Features:
  - Copy on Write- Only save the changes

Branches



- Branch
  - A branch is a distinct variant of the project base.
  - Branches allow isolated environments for working on new features and fixes without disrupting the project.
  - Branches are created by "**forking**" from either the main branch (Trunk) or another branch.

Merging



- Merge
  - A branch may be merged into another.
  - The branch creates a "**pull request**" requesting that its present version be merged into another. The requestor details how to make the changes.
  - An authority from the branch being merged into then is expected to review the request.

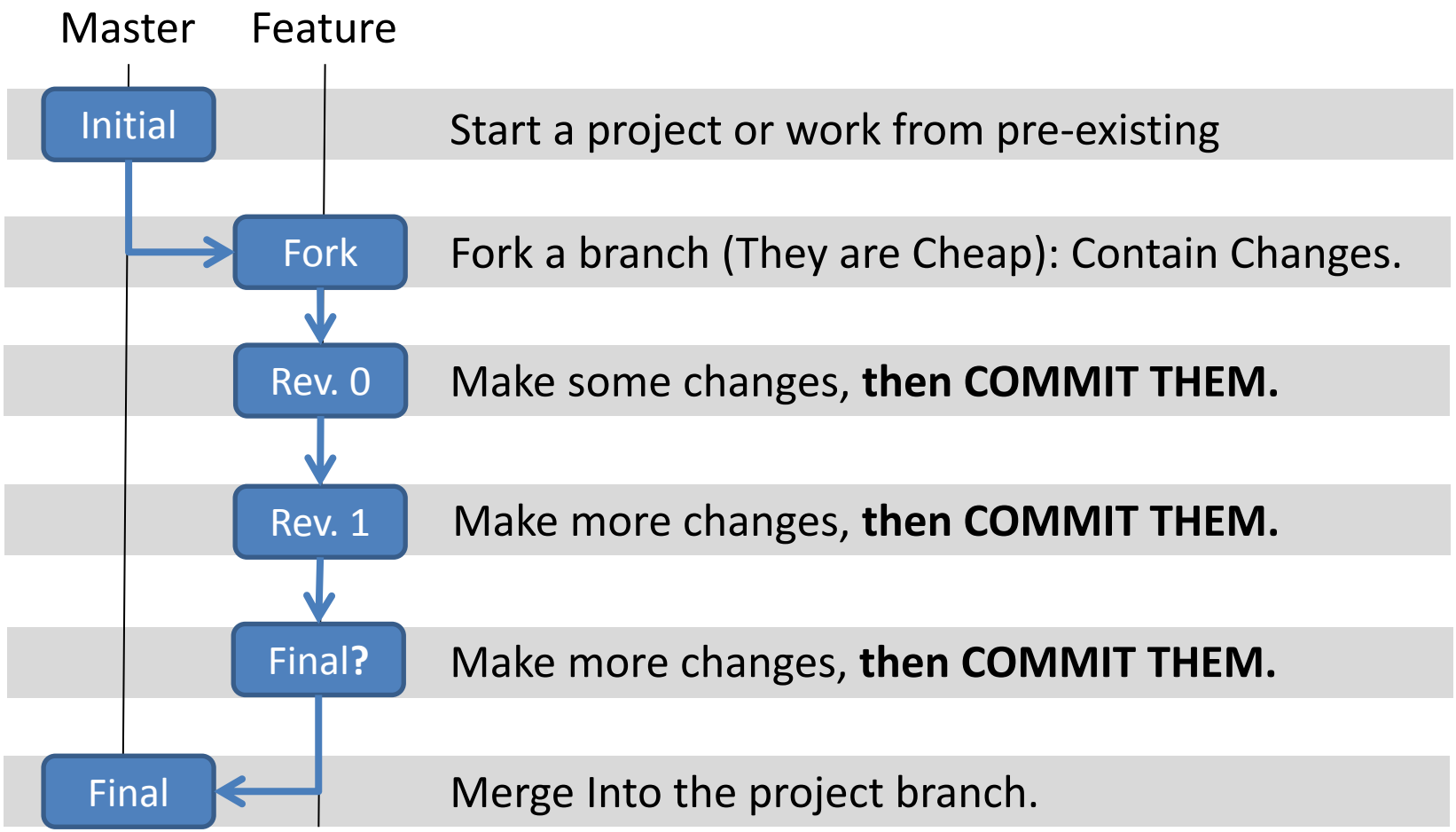
The Case for Version Control

(Or How You Have Already Made It)

CASE A: Giant Undo Button

- Implement a solution, doubt, and delete.
  - Is there a backup of an older version?
    - Is the older version the most recent?
    - What work, if any, must be redone (lost productivity)?
- This is why **backups are important**
  - This is a fundamental tenant of version control.
  - **Lesson 1: VERSION CONTROL AS BACKUP**

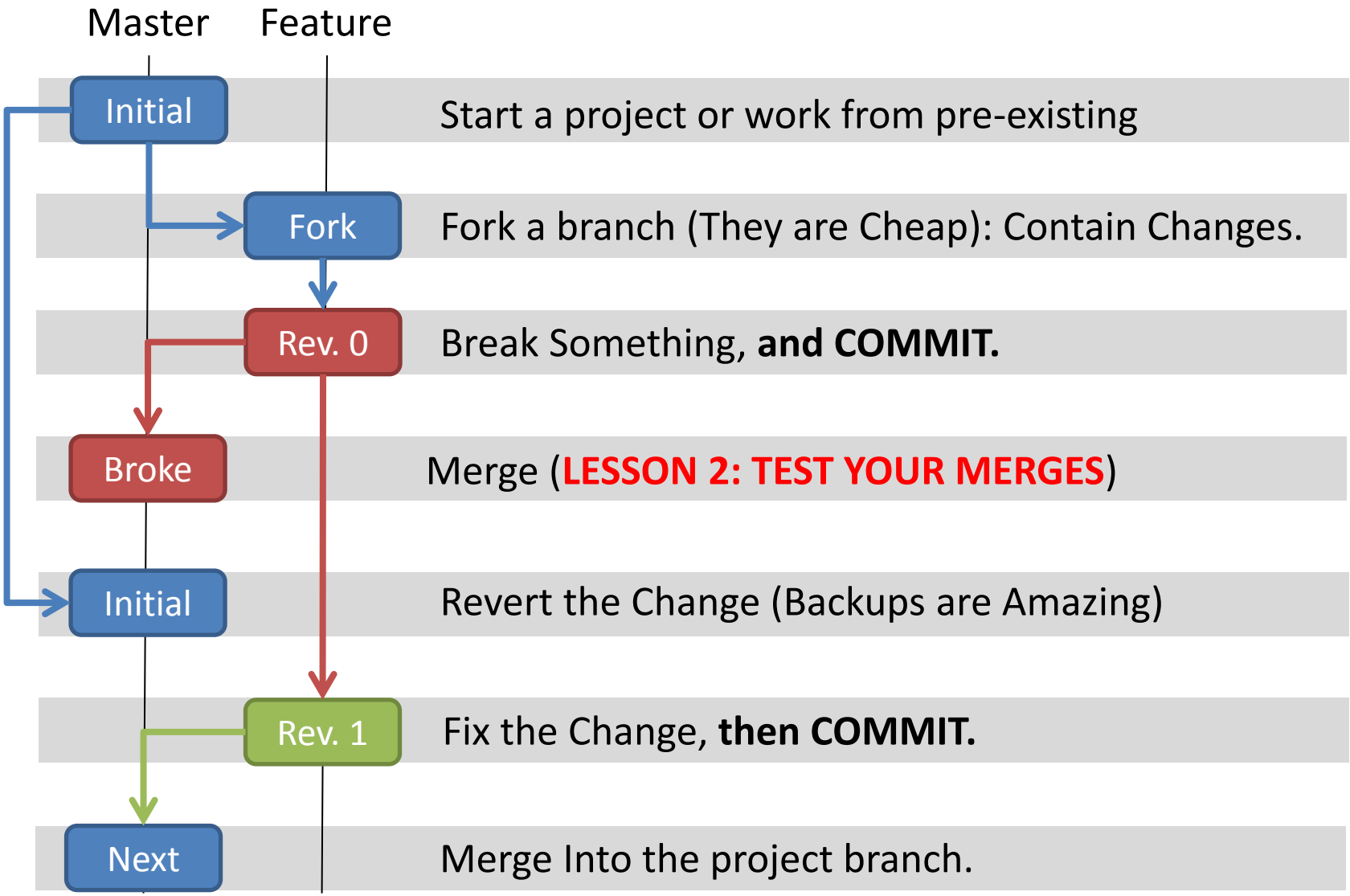
Solution A: Save Incremental Changes



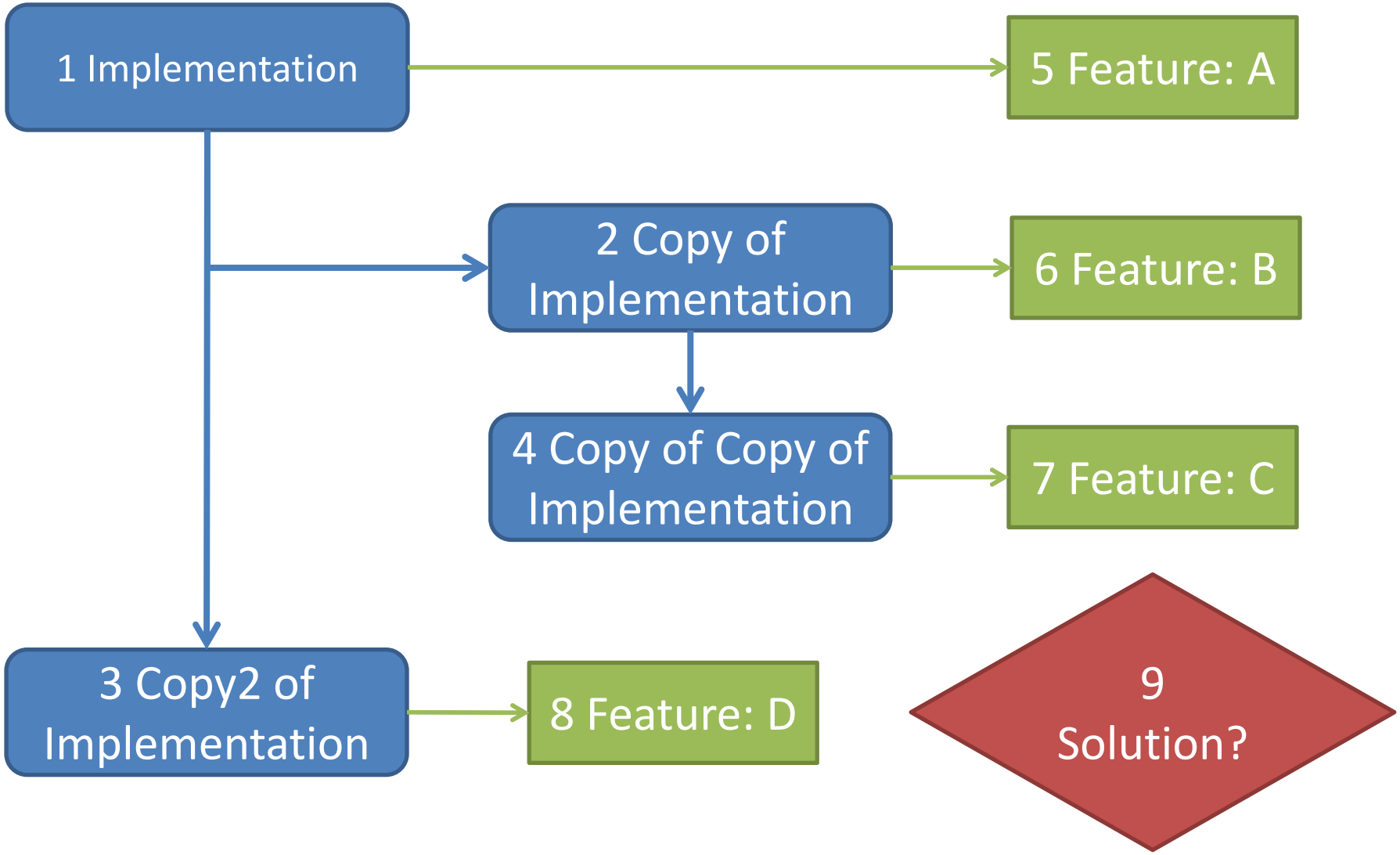
CASE B: The New Breaks The Old

- Fixing a smaller problem breaks a significant feature.
  - What changed?
    - First step in fixing the problem.
    - Humans **SUCK** at answering this question.
  - **Can I revert the changes?**
    - This is why version control is important.

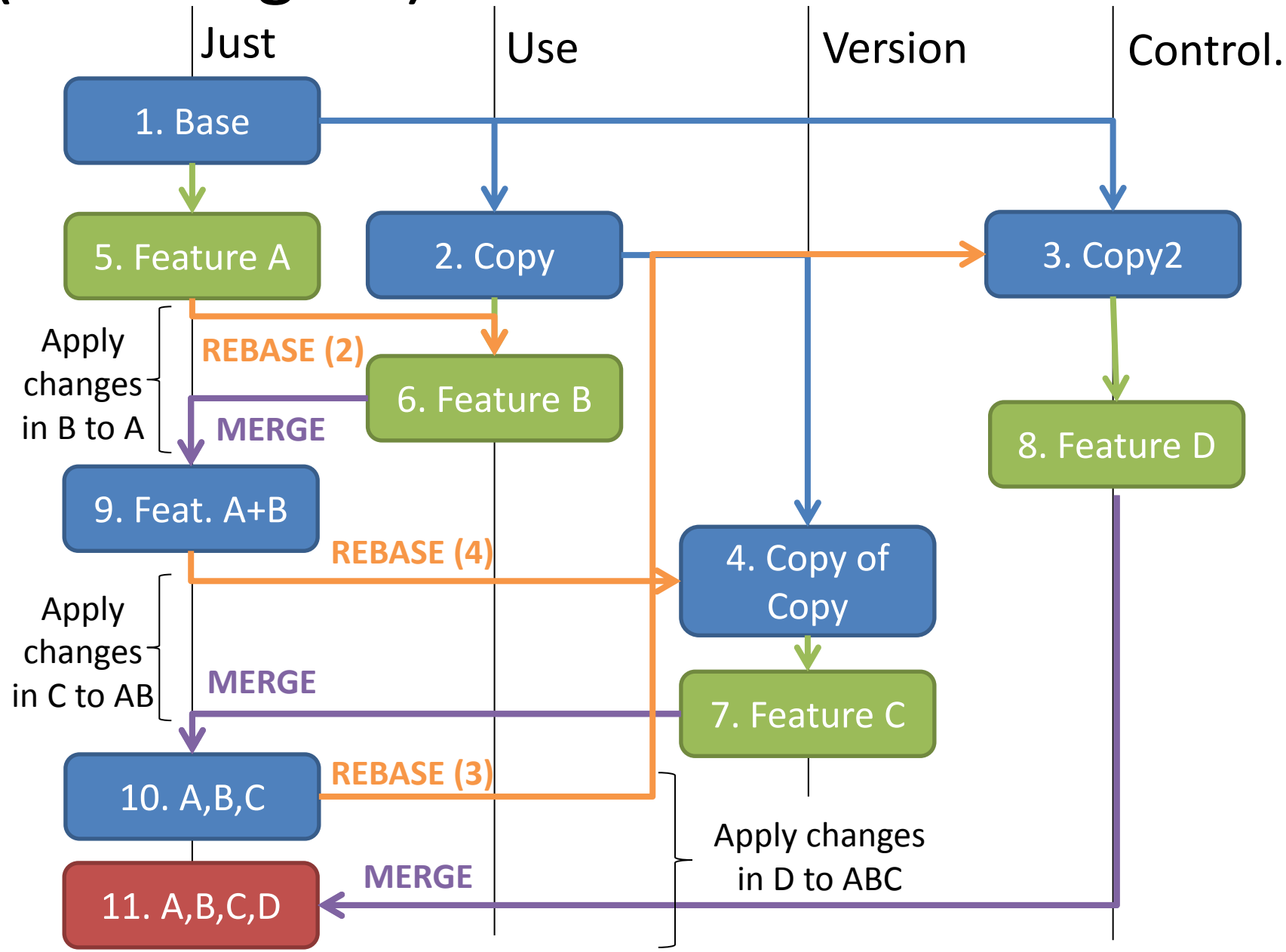
Scenario B: Something Broke



CASE C: Multiple Copies



(Tautological) Solution C: Version Control

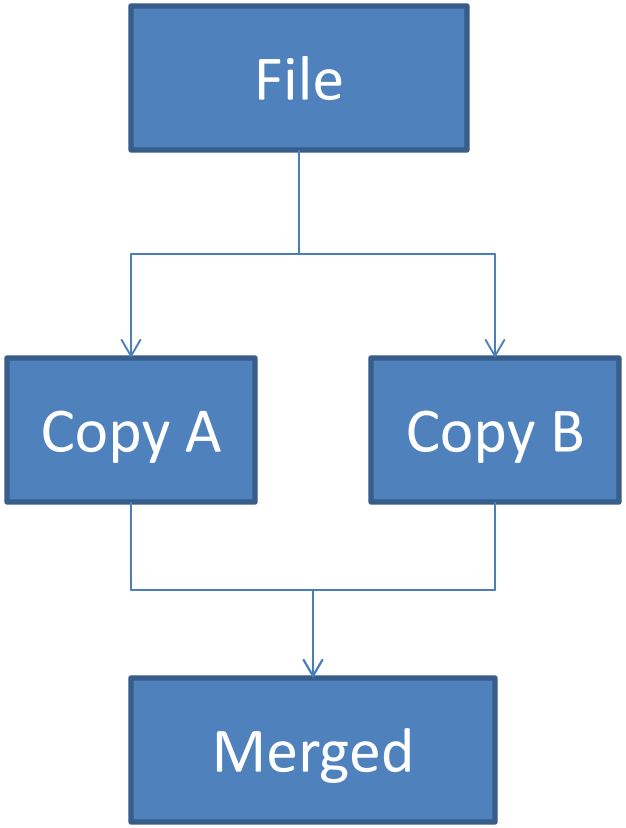


Engineering: Applications

Why should **I** care?

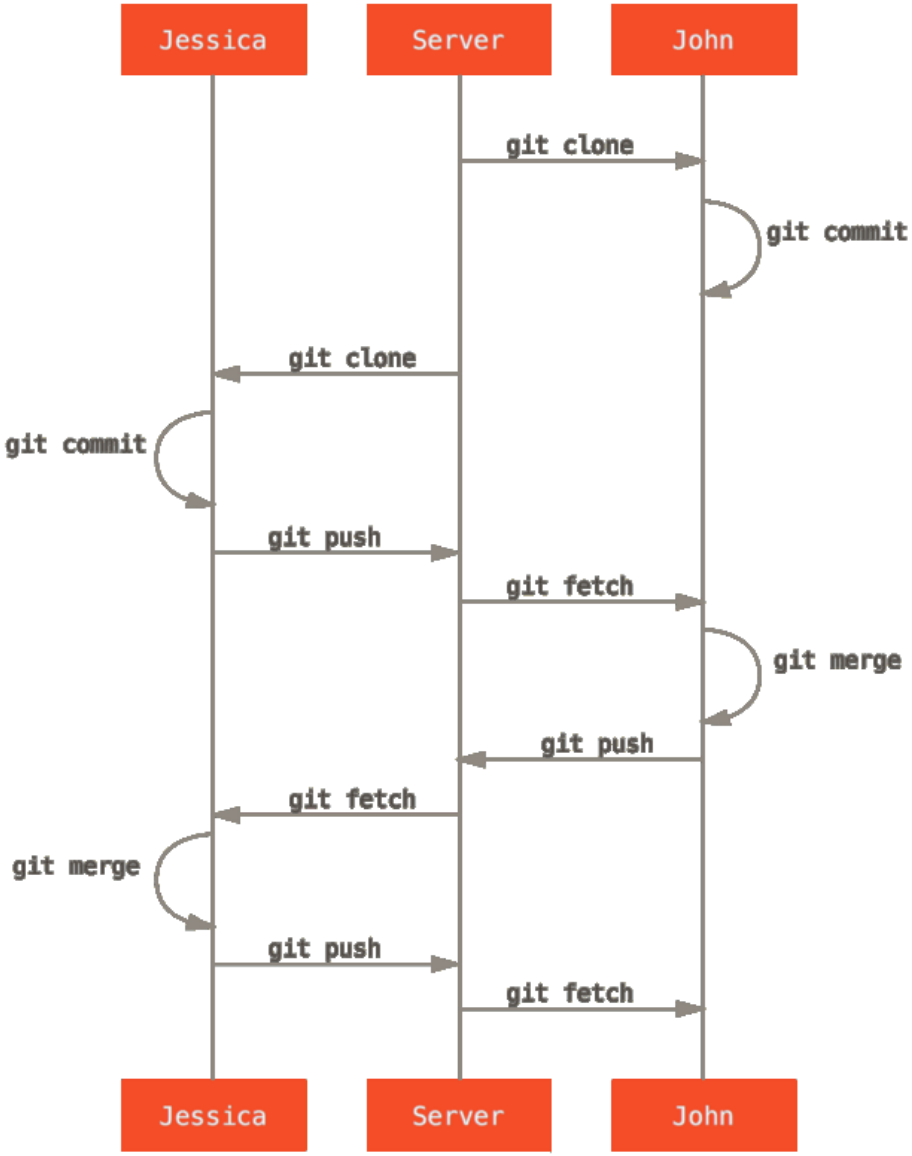
CASE 1: Multiple Team Members

- Multiple individuals need to work on the same file.
  - Option A: Only one may work on the file at a time (Checkout Model).
  - Option B: Everyone gets a copy, but you must reconcile the differences.



Solution 1: User Branches

- GIT follows Option B.
  - Some overhead reconci differences is created.
  - GIT's model requires changes be made with respect the current vers
    - Remember REBASE.



CASE 2: Everything is Broke

- Changes are made irresponsibly.
  - Commits track whom made the change.
    - "**BLAME**" Cites whom made the previous change(s)
  - "**DIFF**" allows side-by-side comparison of versions logging additions, changes, and removals.

Solution 2: Diff

GitHub (3rd Party) Implementation

88 Makefile

View

```
@@ -7,11 +7,55 @@ BINARY_NAME = circle_box
 
 # COMPILER OPTIONS
 COMPILE_OPTIONS = -pthread
-COMPILE_OPTIMIZATION = -O0
 
-# [DEBUGGING] Settings
-LINK_DEBUG = -g3
 
 #https://gcc.gnu.org/onlinedocs/gcc-4.2.4/gcc/Debugging-Options.html#Debugging-Options
+#*********************************************************************
+# Support for Additional compile-time parameters
+#*********************************************************************
+# override directive utilized for text assertions printed to end-user
+#https://www.gnu.org/software/make/manual/make.html#Override-Directive
+
+# [BUILD TYPE]
+# Defaults to debug build configuration if not specified
+# Supported options: debug, internal, release
+ifeq ($(build),release)
+       override build = RELEASE
+       LD_BUILD =
+       CPP_DEBUG =
+else
+       ifeq ($(build),internal)
+               override build = INTERNAL
+               LD_BUILD =
+               CPP_DEBUG = -g3
+       else
+               # DEFAULT CASE
+               override build = DEBUG
+               LD_BUILD = -Xlinker -M=$(BINARY_NAME).map -Xlinker --cref
+               CPP_DEBUG = -pedantic -Wextra -Wconversion -g3
+       endif
+endif
+
+# [OPTIMIZATIONS]
+# Defaults to NO optimizations
+# Supported options: O0, O1, O2, O3
+ifeq ($(optimize),O3)
```

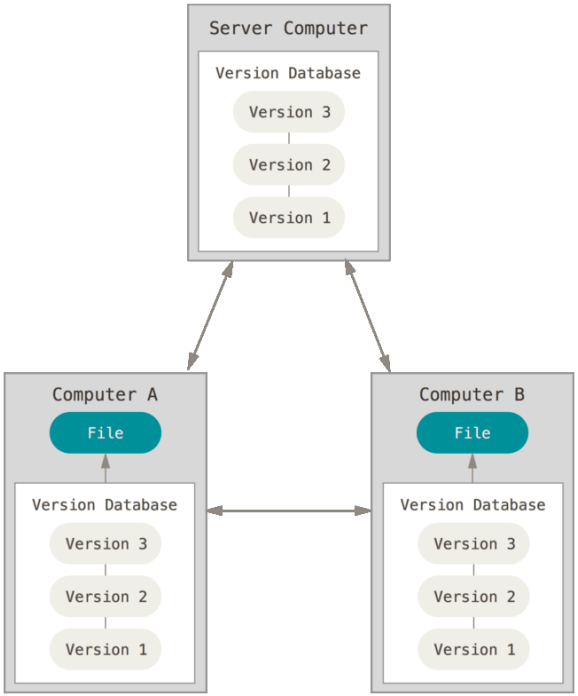
-{Removed}

+{Added}

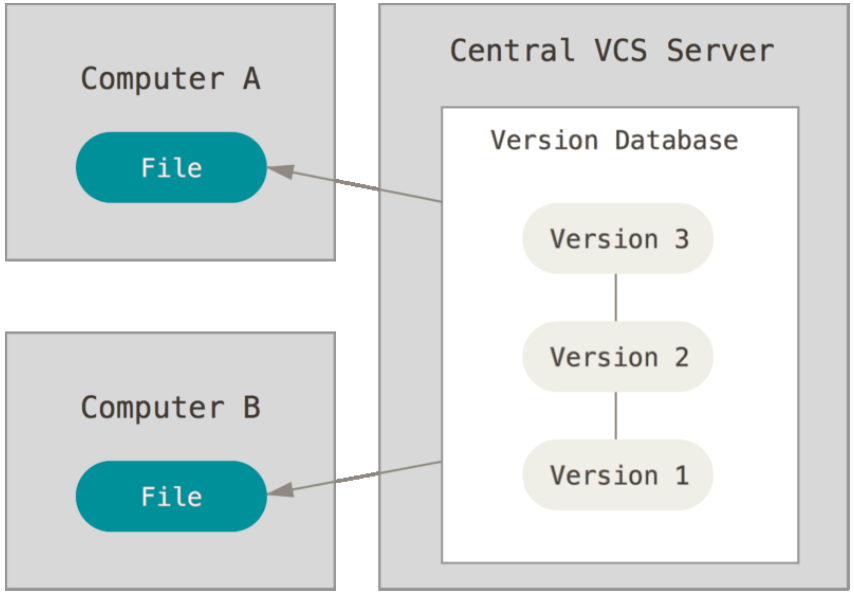
Note: Plain text files work best.
Binary files are not supported.

No (Internet) Connection?

- Distributed Version Control
  - The user "**clone**" a repository.
  - The copy is a complete version
  - GIT uses this model.



- Centralized Version Control
  - The user has copies of files.
  - The repository only on the server
  - Subversion (SVN) uses this model.

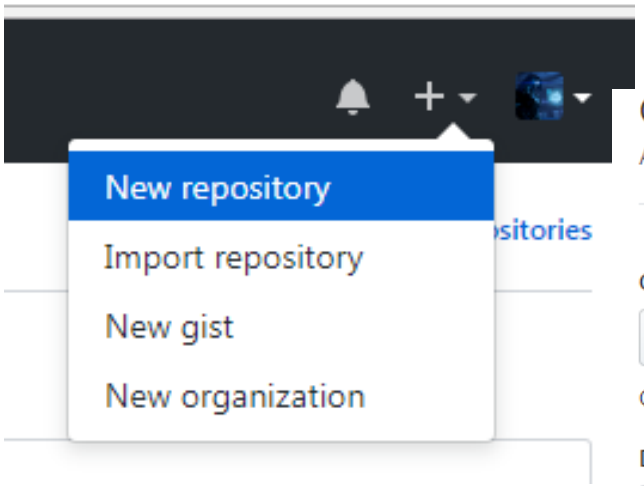


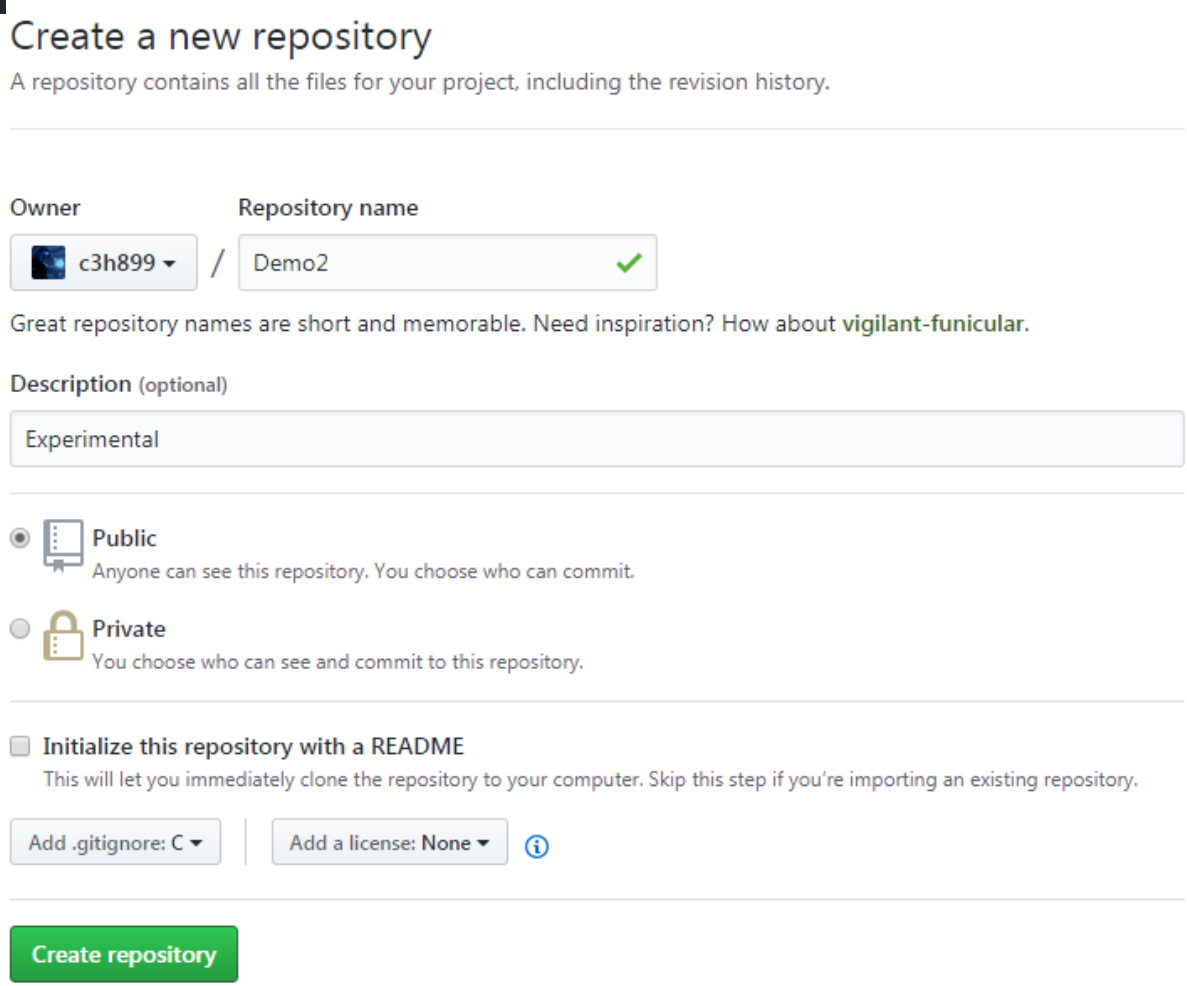
Getting Started

- GIT
  - https://git-scm.com/downloads
- Choose a license
  - https://choosealicense.com/
- Configure .gitignore
  - https://www.gitignore.io/
- Consider a Web Front-End
  - https://education.github.com/pack (Recommended)
  - https://about.gitlab.com/pricing/

Simple GitHub Usage

Create a New Repository

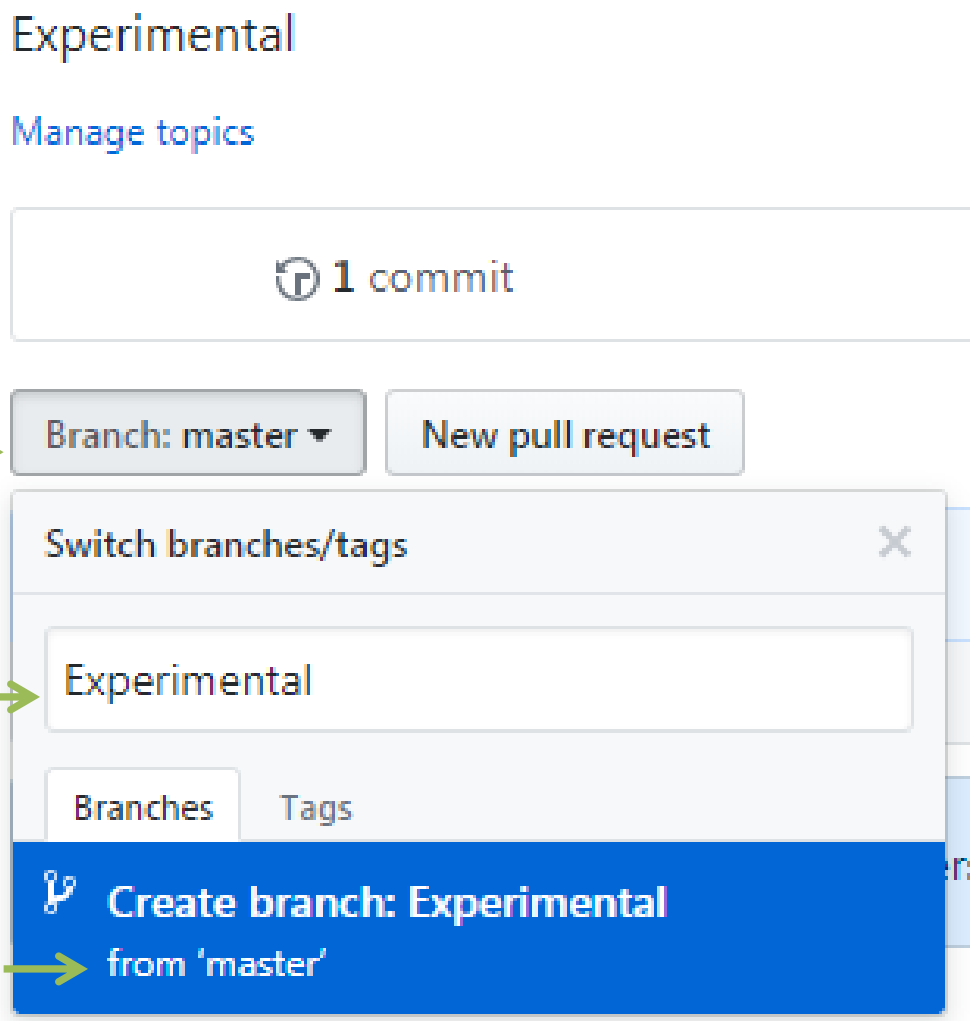




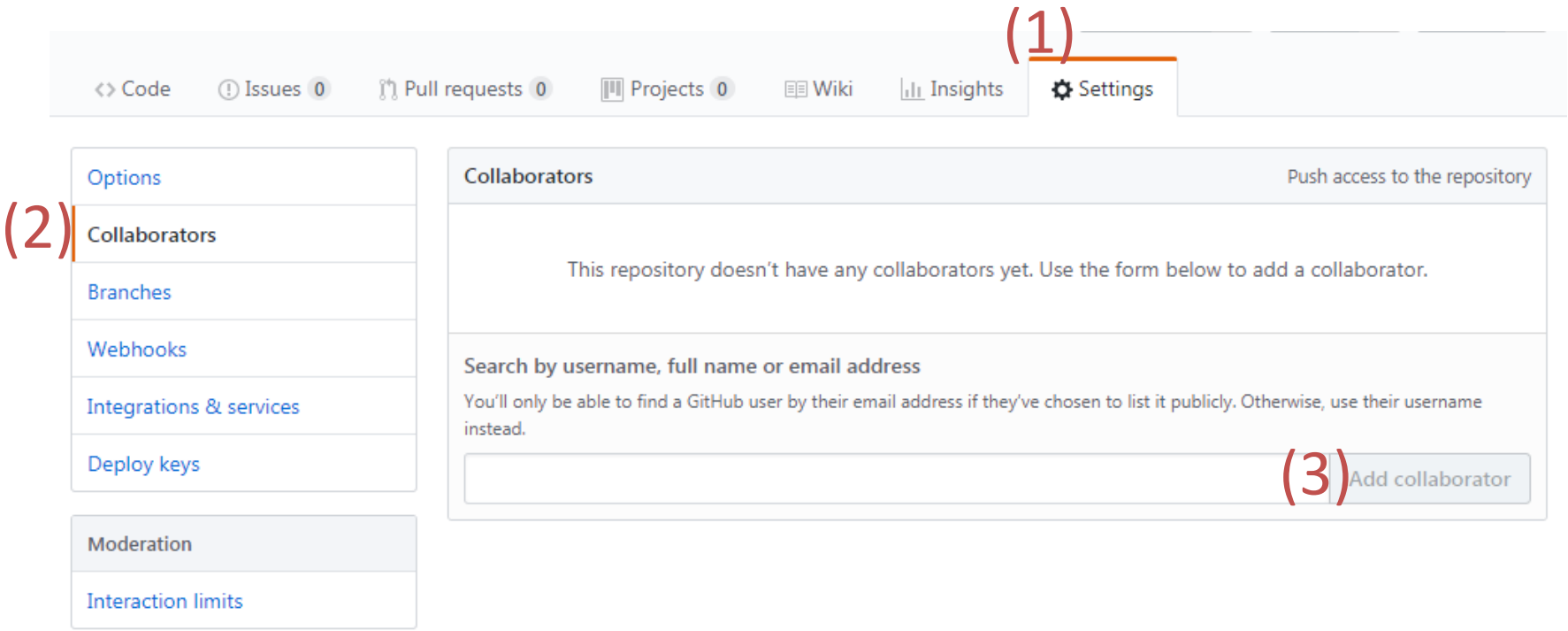
As a student, **Private repositories** are free. **USE** THEM

Create an Experimental Branch

1. Click Here
2. Give it a Name
3. Create the Branch

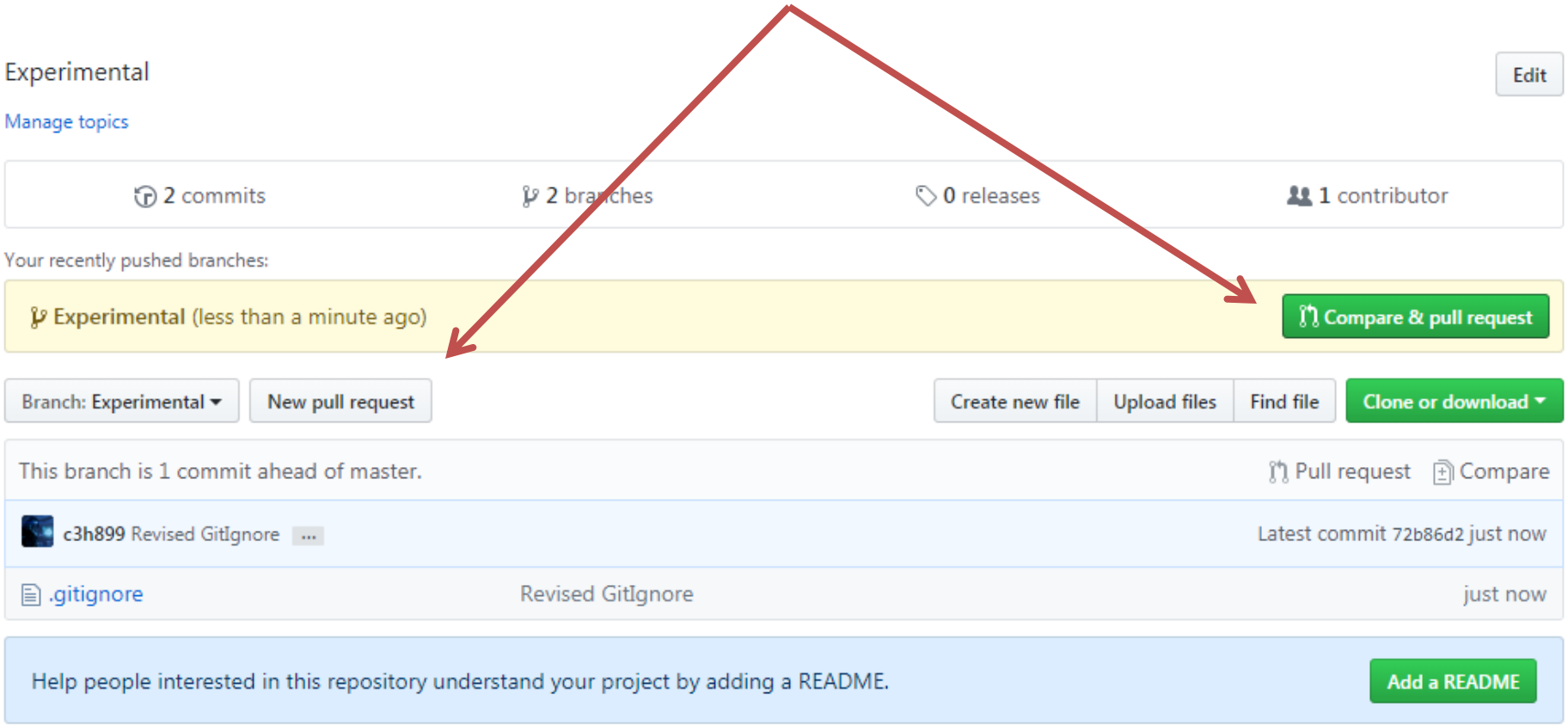


Add Collaborators (Optional)



Merging Branches (Pull Request) 1/3

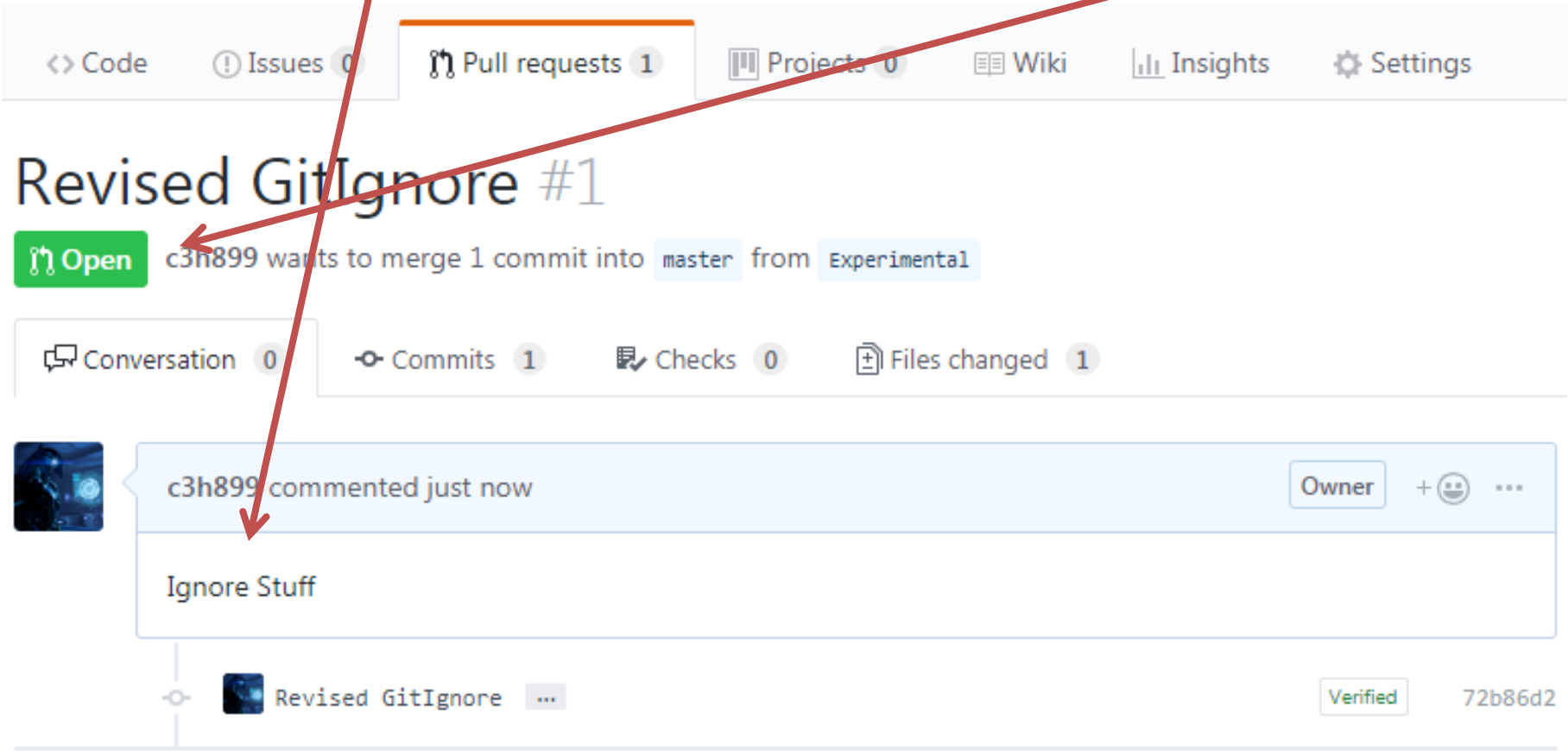
1. Click Here



Merging Branches (Pull Request) 2/3

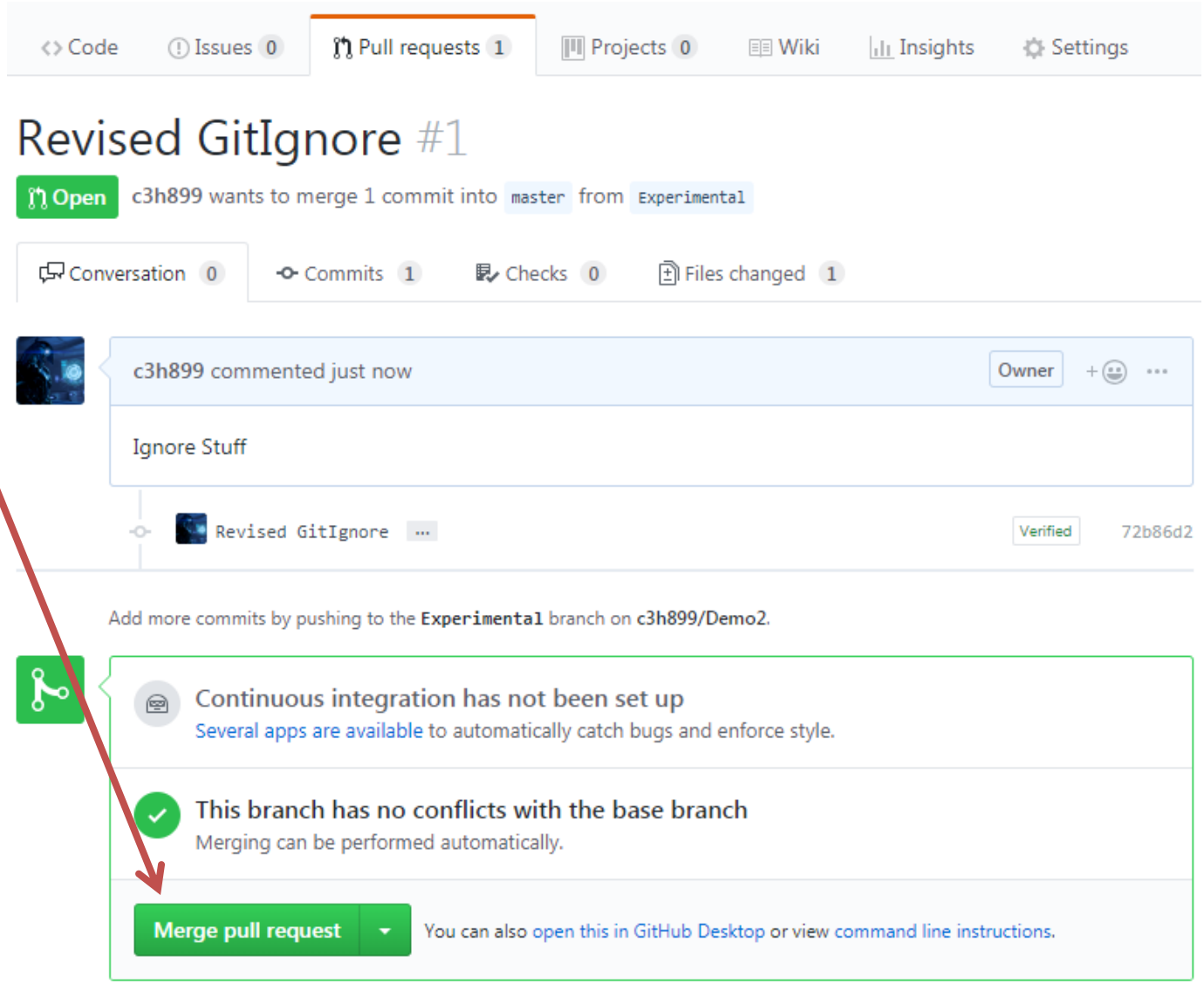
1. Annotate the Request

2. Open the Request



Merging Branches (Pull Request) 3/3

- Merge



Neat Graphs

(1)

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

- Pulse
- Contributors
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network (2)
- Forks

Owners	Sep
c3h899	16

Try GitHub’s Desktop Client

- https://desktop.github.com/

File Edit View Repository Branch Help

Current repository
desktop

Current branch
esc-pr
#3972

Fetch origin
Last fetched 2 minutes ago

Changes | History

Appease linter
iAmWillShepherd committed a day ago

Add event handler to dropdown compon...
iAmWillShepherd and Markus Olsson co...

Move escape behavior to correct compo...
iAmWillShepherd and Markus Olsson co...

Remove event handler from the branches..
iAmWillShepherd and Markus Olsson co...

Merge branch 'master' into esc-pr
iAmWillShepherd committed a day ago

Merge pull request #4044 from desktop/...
Neha Batra committed a day ago

Merge pull request #4070 from desktop/...
Brendan Forster committed 2 days ago

bump to beta3
Brendan Forster committed 2 days ago

Merge pull request #4057 from desktop/...
Brendan Forster committed 2 days ago

Merge pull request #4067 from desktop/...
Brendan Forster committed 2 days ago

Release to 1.1.0-beta2
Neha Batra committed 2 days ago

Add event handler to dropdown component

iAmWillShepherd and Markus Olsson committed c79e71c 1 changed file

```
Co-Authored-By: Markus Olsson <niik@users.noreply.github.com>
```

app\src\ui\toolbar\dropdown.tsx

```
@@ -145,6 +145,10 @@ export class ToolbarDropdown extends React.Component<
145 145     this.state = { clientRect: null }
146 146   }
147 147
    148 +  private get isOpen() {
    149 +    return this.props.dropdownState === 'open'
    150 +  }
    151 +
148 152   private dropdownIcon(state: DropdownState): OcticonSymbol {
149 153     // @TODO: Remake triangle octicon in a 12px version,
150 154     // right now it's scaled badly on normal dpi monitors.
@@ -249,6 +253,13 @@ export class ToolbarDropdown extends React.Component<
249 253     }
250 254   }
251 255
    256 +  private onFoldoutKeyDown = (event: React.KeyboardEvent<HTMLElement>) => {
    257 +    if (!event.defaultPrevented && this.isOpen && event.key === 'Escape') {
    258 +      event.preventDefault()
```

Additional References

- GIT SCM (Official Site)
  - https://git-scm.com/
- git - the simple guide
  - https://rogerdudler.github.io/git-guide/
- GitHub GIT CHEAT SHEET
  - https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

Citations

- Figures from the CASE section: ProGit 2nd Edition (2014) by Scott Chacon and Ben Straub released under a CC NC SA license.
- GitHub Diff from GitHub.com