



Defining Version Control



What was life like **prior to** the introduction of Version Control Systems?

- ▶ Messy—life in the vacuum was oppressive and bleak.
- ▶ To introduce and roll out a change to a product, an individual developer would, for instance
 - retrieve archived version corresponding to a version in the production environment
 - proceed to make and test the change
 - assign a new version number to the release
 - add notes detailing the change and release back to the production environment
- ▶ To revert a change
 - match the release notes to specific files
 - reverse-engineer the actual changes introduced in those files
 - revert the changes manually and test
 - deploy the rectified version with release notes back to the production environment

GitHub Training

Getting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Generations of Version Control Systems (VCS)



Third generation — Distributed Version Control Systems (DVCS)

- ▶ Simultaneous modifications with the commit-before-merge philosophy.
- ▶ Each developer keeps a copy of the remote repository alongside their working copy.
- ▶ Backups are maintained in a distributed manner, and users are required to merge differences and resolve conflicts before pushing changes.
- ▶ Example:
 - Git
 - Mercurial
 - BitKeeper

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Version Control is Your Friend

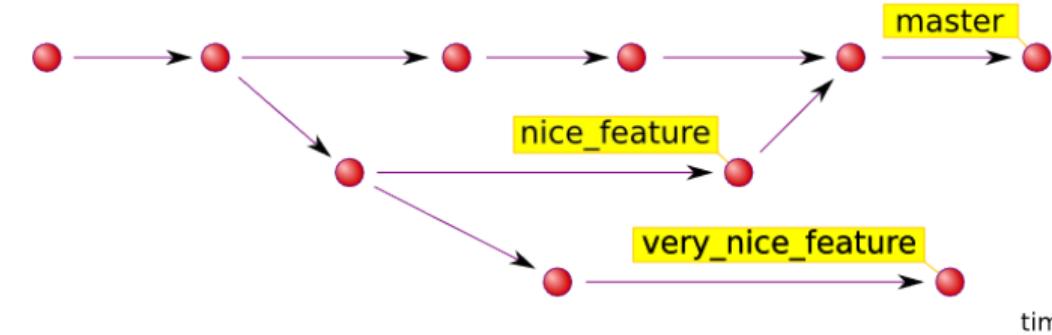


Image Taken Directly From hades.github.io

What are ...

- ▶ Repositories? Pulls?
- ▶ Commits/Pushes?
- ▶ Tags?
- ▶ Branches? (Master? Develop?)
- ▶ Conflicts?
- ▶ Merges?

GitHub Training

Getting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

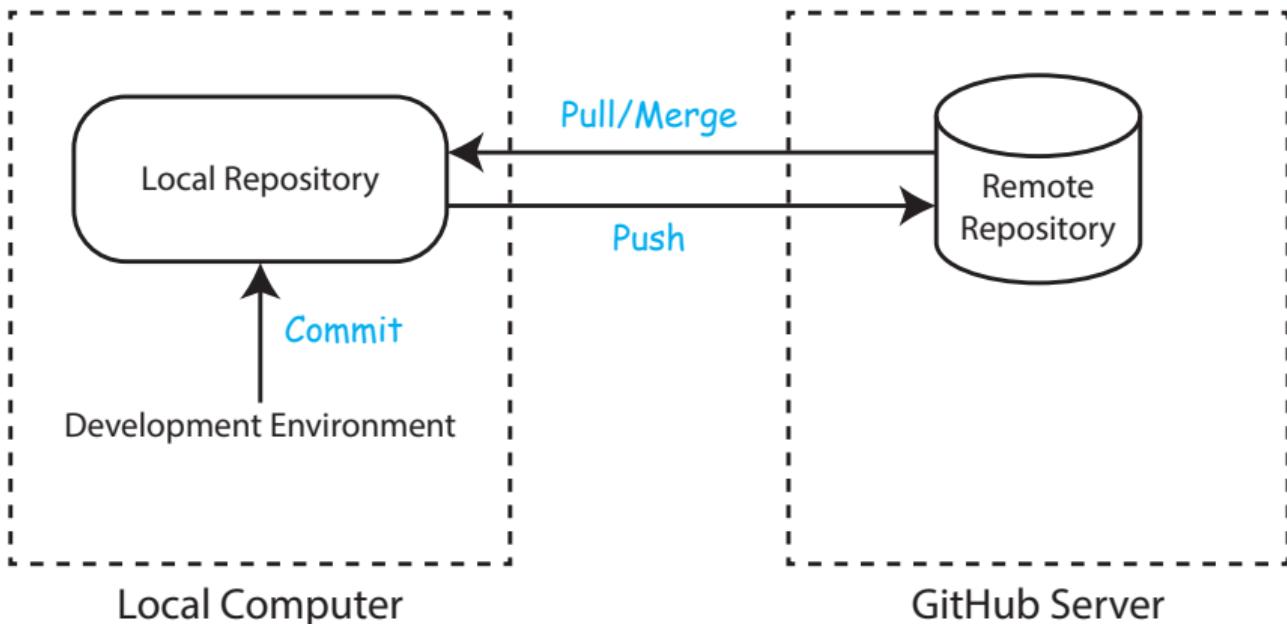
Issues

Pull Requests

Practical Training



GitHub Architecture



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Git Commands



What is Git? — Distributed version control system

- ▶ `git init`
Initialization of a repository — creates .git folder and contents
- ▶ `git clone /path/to/repository`
Create a local copy of an existing repository
- ▶ `git remote add origin /path/to/repository`
Add a new remote repository
- ▶ `git add <filename>`
Add a file to be committed
- ▶ `git commit -m "This is my awesome commit message"`
Commit in your local repository with a message
- ▶ `git push -u origin master`
Updates remote repository with local repository changes
- ▶ `git pull`
Fetch and merge changes from remote repository

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Git Commands, tl;dr



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training

Check out these cheat sheets:

<https://education.github.com/git-cheat-sheet-education.pdf>

<https://www.git-tower.com/blog/git-cheat-sheet>



And Now For The Discipline...



A Developer's **Checklist** for pushing to a Remote Repository.

1. Make Changes Locally to Code Base
2. Build Changes Error-Free in Working Directory
3. Commit Changes Locally with Detailed Commit Messages
4. pull from the Remote Repository
5. Merge and Reconcile Conflicts
6. Build Changes Error-Free in Working Directory
7. push to the Remote Repository

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training

Easy when you get the hang of it!



Git Workflows



Workflows refer to the approach a team takes to introduce changes to a code base.

- ▶ A workflow is characterized by a distinct approach in the usage of branches (or lack thereof) to introduce changes to a repository. For example,
 - Gitflow Workflow
 - Centralized Workflow
 - Feature Branch Workflow
 - Forking Workflow

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Git Workflows



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

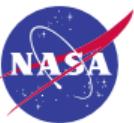
Practical Training

Workflows refer to the approach a team takes to introduce changes to a code base.

- ▶ A workflow is characterized by a distinct approach in the usage of branches (or lack thereof) to introduce changes to a repository. For example,
 - Gitflow Workflow
 - » Two branches are used: **master** and **develop**.
 - » The master branch is used to track release history.
 - » The develop branch is used to track the total history of feature integration.
 - Centralized Workflow
 - Feature Branch Workflow
 - Forking Workflow



Git Workflows



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training

Workflows refer to the approach a team takes to introduce changes to a code base.

- ▶ A workflow is characterized by a distinct approach in the usage of branches (or lack thereof) to introduce changes to a repository. For example,
 - Gitflow Workflow
 - Centralized Workflow
 - » The **master** branch is the default development branch too.
 - » All changes are committed to the master branch.
 - » This is a suitable workflow for small teams or those transitioning from SVN (trunk = master).
 - Feature Branch Workflow
 - Forking Workflow



Git Workflows



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training

Workflows refer to the approach a team takes to introduce changes to a code base.

- ▶ A workflow is characterized by a distinct approach in the usage of branches (or lack thereof) to introduce changes to a repository. For example,
 - Gitflow Workflow
 - Centralized Workflow
 - Feature Branch Workflow
 - » Feature development is carried out in a dedicated branch.
 - » That branch is then merged to the **master** once the intended changes are project-approved.
 - Forking Workflow



Git Workflows



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training

Workflows refer to the approach a team takes to introduce changes to a code base.

- ▶ A workflow is characterized by a distinct approach in the usage of branches (or lack thereof) to introduce changes to a repository. For example,

- Gitflow Workflow
- Centralized Workflow
- Feature Branch Workflow
- Forking Workflow

- » The developer seeking to make a change creates a copy of the desired repository in their individual GitHub account.
- » The changes are made in the copy of the source repository.
- » It is then merged to the source repository through a “pull request”.



Git Workflows



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training

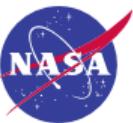
Workflows refer to the approach a team takes to introduce changes to a code base.

- ▶ A workflow is characterized by a distinct approach in the usage of branches (or lack thereof) to introduce changes to a repository. For example,
 - Gitflow Workflow
 - Centralized Workflow
 - Feature Branch Workflow
 - Forking Workflow

tl;dr Teams must make process decisions before development starts!



GitHub



What is GitHub? — Git repository hosting service

Why use GitHub? What are some of the features, tools, and advantages?

- ▶ Collaborative software development
- ▶ Issue tracking
- ▶ Web-based GUI
- ▶ Project management

Limitations

- ▶ No files >150MB
- ▶ Will receive a warning for any files >50MB

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



GUIs



- ▶ TortoiseGit (<https://tortoisegit.org/>)
 - Pushing of large amounts of data is known to be slow.
- ▶ Git Extensions (<https://sourceforge.net/projects/gitextensions/>)
- ▶ GitHub Desktop (<https://desktop.github.com/>)
 - Currently does not play well with ssh.
- ▶ GitKraken
 - Requires login paired with GitKraken account; Paid to interface with GitHub Enterprise.
- ▶ Fork
 - Visually-oriented GUI with significant-praise.

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Profiles and Personalization



Profile

- ▶ Add your email address
- ▶ Add and name your public key

Personalization

- ▶ To start with, you will be issued an “Identicon”
- ▶ These are “simple 5×5 “pixel” sprites that are generated using a hash of the user’s ID. The algorithm walks through the hash and turns pixels on or off depending on even or odd values. These generated patterns, combined with hash-determined color values, ensures a huge number of unique Identicons.”
- ▶ You can easily select your own avatar and upload a picture of your choosing.

For this training we will have a competition for best avatar!

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Pull Requests



To start a Pull Request, click “New Pull Request”

- ▶ You will choose a base branch (left) and the branch to merge from (right — Note the arrow)
- ▶ The interface provides a quicklook at the changes.
- ▶ When ready, click “Create Pull Request”
 - For new branches, GitHub will also give you an option to jump to this step from the Repository dashboard.

Your Pull Request must follow rules for merging.

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Pull Requests



But what can I do with Pull Requests?

- ▶ Individuals can be assigned as Reviewers or Assignees
 - Reviewers can be anyone with READ access.
 - Reviewers can be re-added once suggested changes are made to re-review.
 - Assignees are placed as developers expected to perform the merge.
- ▶ Pull Requests can be given labels, placed under a project, or be associated with a milestone.
- ▶ Specific users can additionally be mentioned using @USERNAME. Assigned and mentioned users are subscribed to discussion.
- ▶ The Pull Request becomes a record of the conversation and decisions that led to a change being incorporated.
 - The Pull request contains the commit history and a complete changelog.
- ▶ When the change is ready to merge, click “Merge Pull Request”

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Pull Requests



Conversation 0 Commits 1 Checks 0 Files changed 1 +3 -1

kleavor commented 4 minutes ago

Updating README with real sections

Update README d56afaa

kleavor self-assigned this 4 minutes ago

Add more commits by pushing to the `updatereadme` branch on `kleavor/ThePlace`.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Close pull request Comment

Reviewers
No reviews

Assignees
kleavor

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications

Unsubscribe
You're receiving notifications because you were assigned.

1 participant

Lock conversation

GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training



Pull Requests



A screenshot of a GitHub pull request interface. At the top, there are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). Below the tabs, it says "Changes from all commits" and "Jump to...". There are buttons for "Diff settings" and "Review changes". The main area shows a diff of a README.md file. The diff shows a merge commit with the message "# My Test README" and a descriptive section below it. The commit has 4 additions and 1 deletion.

In Closing...

- ▶ When you merge a pull request, you are given the usual commit log
 - Enter a short (≤ 50 char) description. By default this is “Merge ...”
 - Provide a more descriptive commit log for the merge. If this closes a ticket, inserting language like “Closes #432” will allow GitHub to automatically close the ticket.
- ▶ After merging, GitHub will also give you the option to delete the feature branch (if created from a branch, not a fork). It can be restored!



GitHub Training

Gitting Started

What is Git?

GUIs

Utilizing GitHub

VCS Philosophy

Projects

Issues

Pull Requests

Practical Training