

SWCON201 Software Development Methodology and Tools

Software Maintenance

2020. 3. 1

About

- Ice-break
- Class
- Summary
- Reference
- Discussion
- Homeworks

Prof. Sungwon Lee

Office: 323

Email: drsungwon@khu.ac.kr

Web: mobilelab.khu.ac.kr

Ice-break

How multiple programmers co-work at the same time for the shared files?

- GitHub (<https://github.com/>)
- Google Docs (<https://docs.google.com/>)

Class

- Version Control
- Concurrent Versions System (CVS)
- Apache Subversion (SVN)
- Git
- GitHub

Class

Version Control

- Also known as revision control or source control
- Management of changes to documents, computer programs, large web sites, and other collections of information
- Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision"
- For example, an initial set of files is "revision 1". When the first change is made, the resulting set is "revision 2", and so on
- Each revision is associated with a time-stamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged
- Embedded in various softwares such as word processors, spreadsheets, collaborative web docs and in various content management systems (Wikipedia's page history)

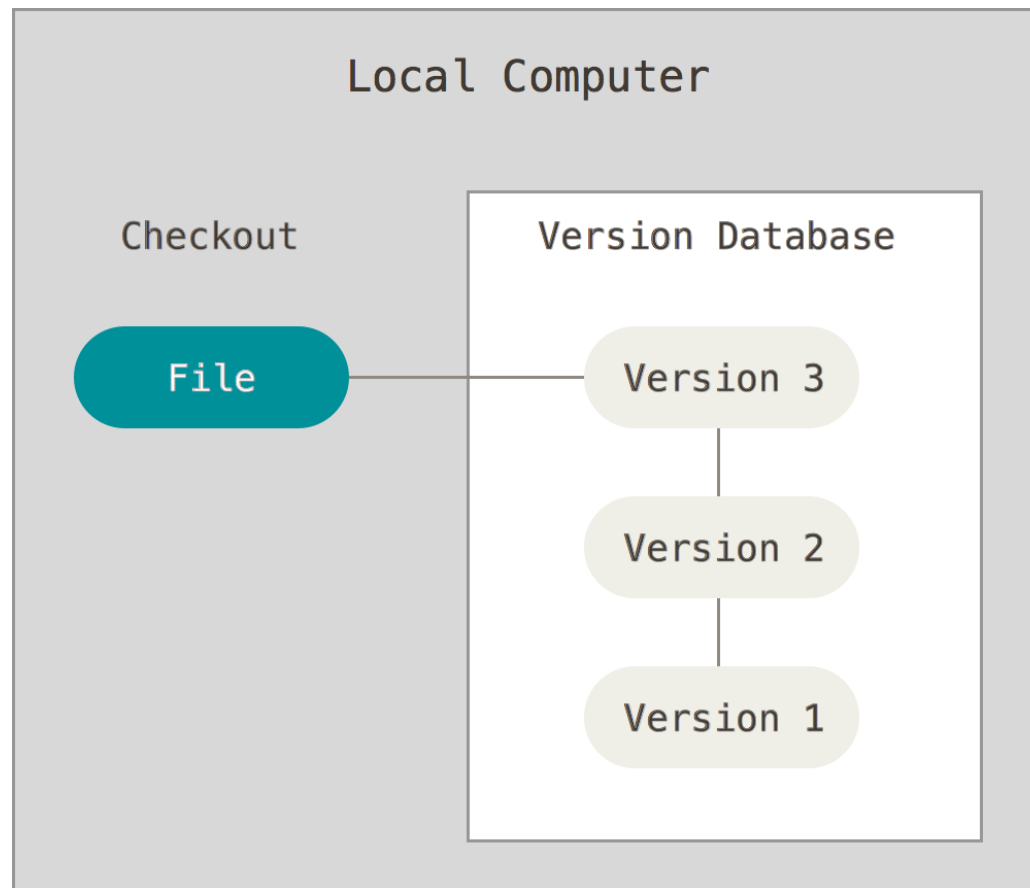
Class

Version Control like This?



Class

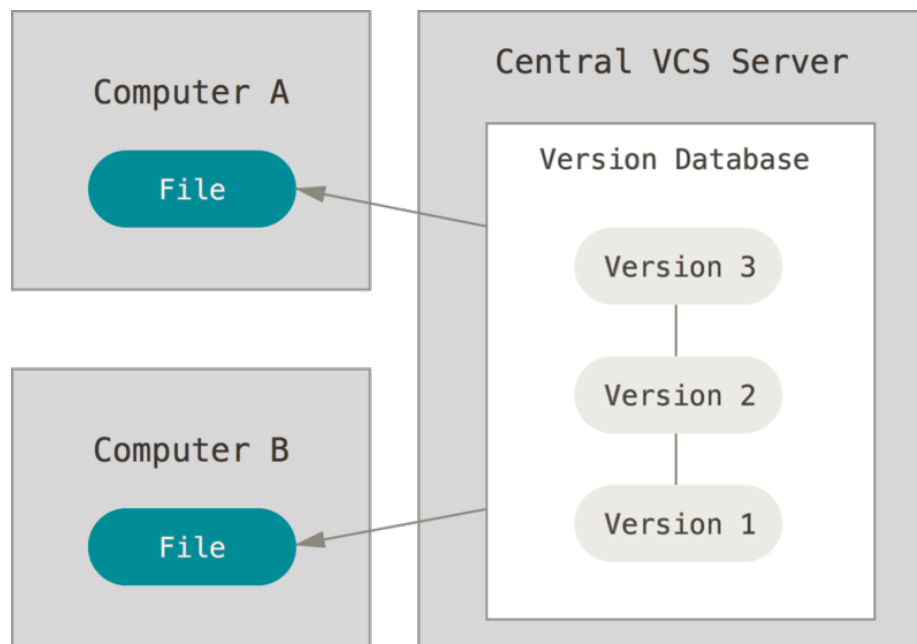
Version Control on Single Machine



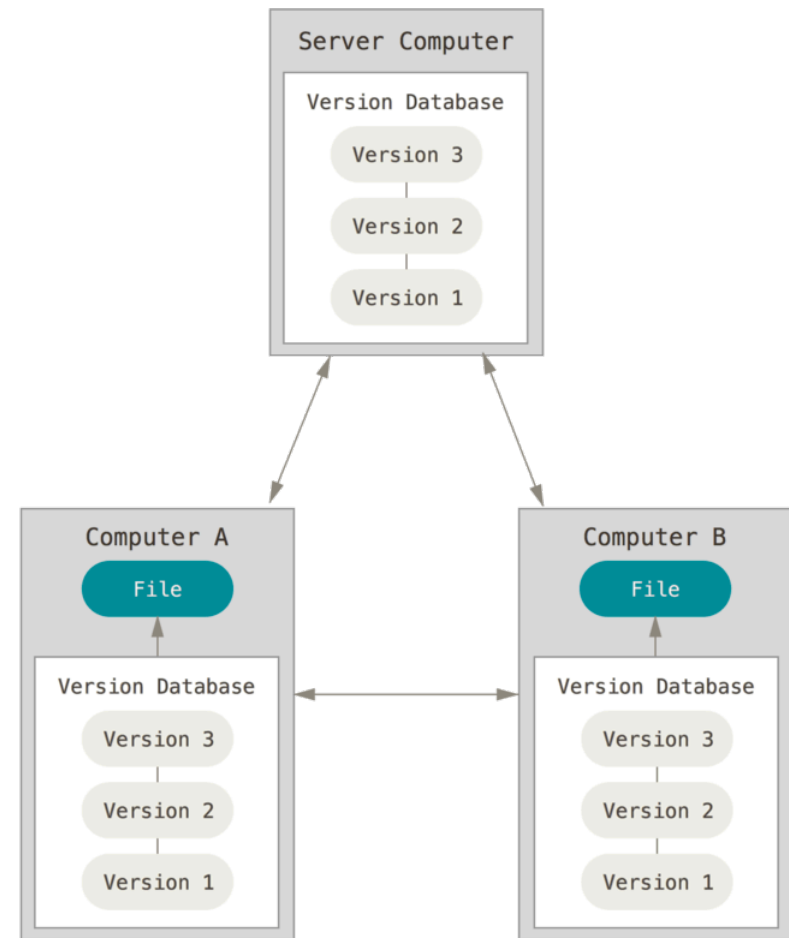
Local Version Control Systems

Class

Version Control over Multiple Machines



Centralized Version Control Systems

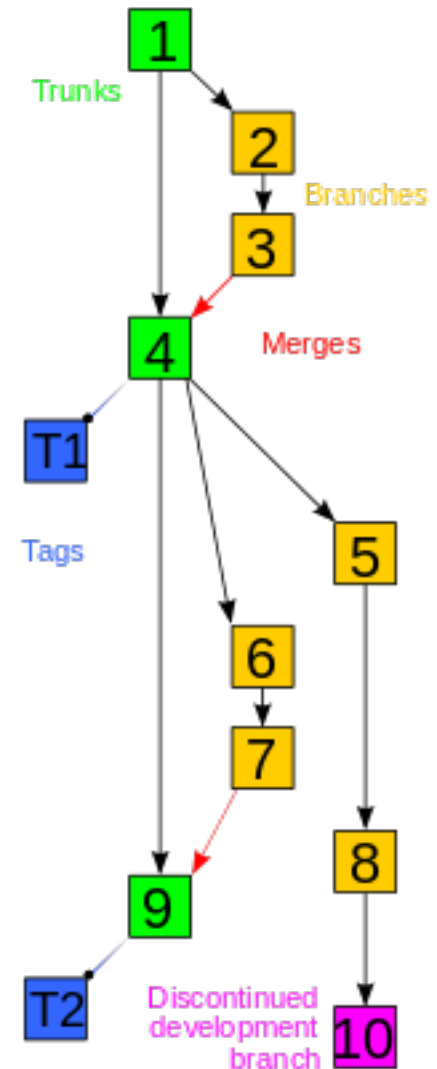


Distributed Version Control Systems

Class

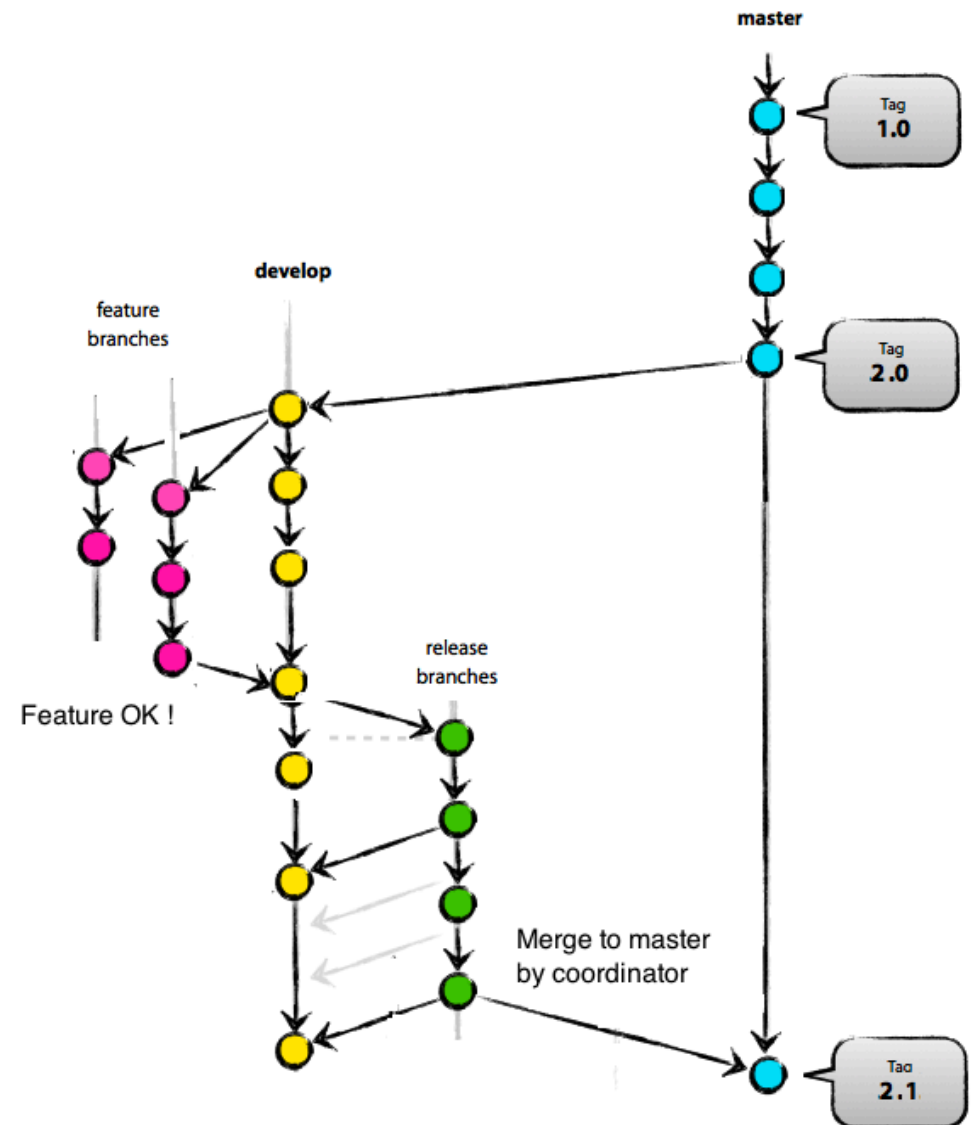
Version Control

- Revisions are generally thought of as a line of development (the trunk) with branches-off of this, forming a directed tree, visualized as one or more parallel lines of development (the "mainlines" of the branches) branching off a trunk
- In reality, the structure is more complicated, forming a directed acyclic graph, but for many purposes "tree with merges" is an adequate approximation



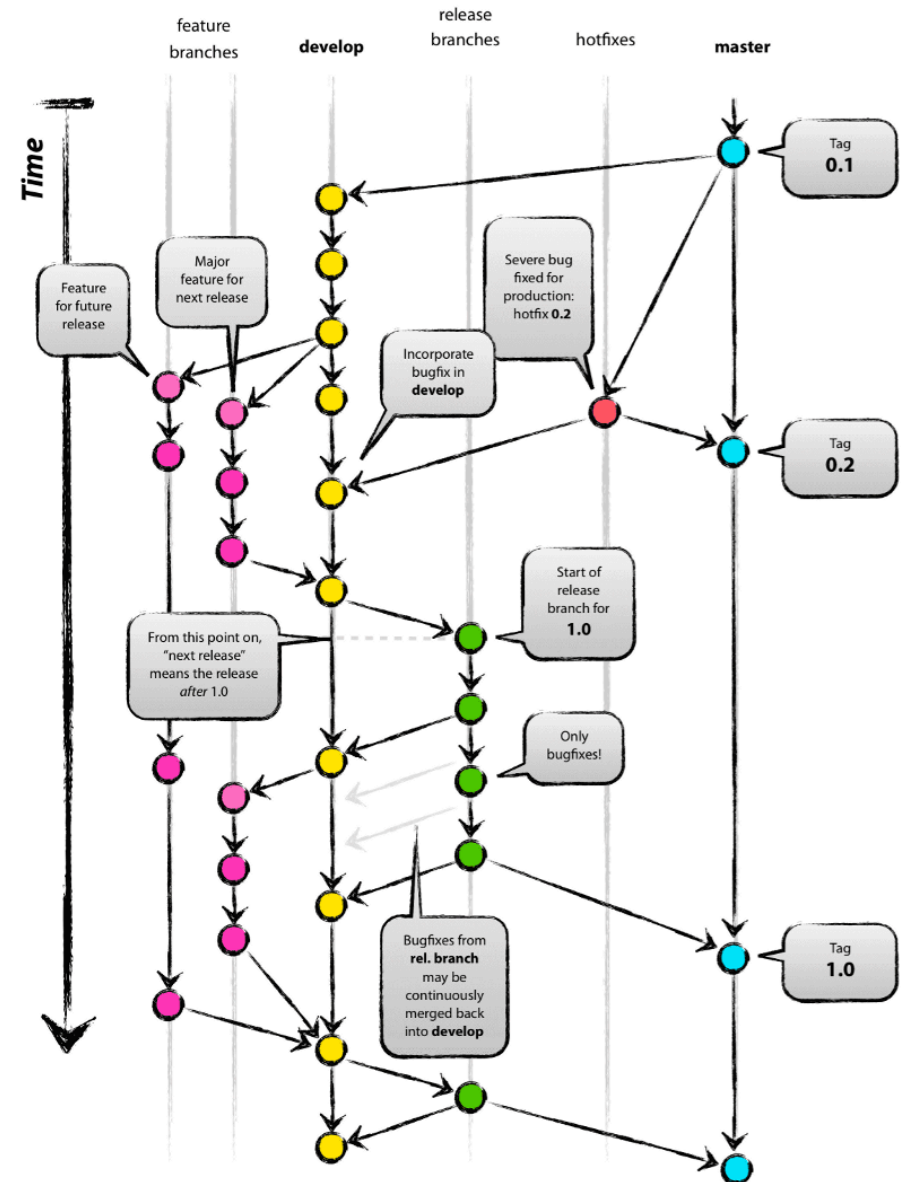
Class

Version Control Example 1



Class

Version Control Example 2




Class

Version Control

- Common Vocabularies
 - Go to: https://en.wikipedia.org/wiki/Version_control
 - Visit “Common Vocabulary”

Class

Concurrent Versions System (CVS)



Not Logged in

Login
New User

This Page

Language
Clean Reload
Printer Version

Search

in Projects Search

Hosted Projects


Register New Project
Full List
Contributors Wanted
Statistics

Site Help

User Docs: Cookbook
User Docs: In Depth Guide
Get Support
Contact Savannah
User Docs: FAQ

GNU Project

Help GNU
All GNU Packages
Dev Resources
License List



Concurrent Versions System - Summary

[Group](#) [Main](#) [Homepage](#) [Download](#) [Mailing lists](#) [Source code](#) [Bugs](#) [Tasks](#) [Patches](#) [News](#)

This project is not part of the GNU Project.

CVS is a version control system, an important component of Source Configuration Management (SCM). Using it, you can record the history of sources files, and documents. It fills a similar role to the free software RCS, PRCS, and Aegis packages.

Please see the <http://www.nongnu.org/cvs/> project home page for more information.

Registration Date: Wed 06 Jul 2005 08:30:35 PM UTC
License: [GNU General Public License v2 or later](#)
Development Status: 5 - Production/Stable

Membership Info

Project Admins:

- Larry Jones
- Mark D. Baushke
- Derek Robert Price

10 active members
[\[View Members\]](#)





Group identification

Id: #7788
System Name: cvs
Name: Concurrent Versions System
Group Type: non-GNU software and documentation

Search in this Group

in Cookbook Search

Quick Overview

-  [Project Homepage](#)
-  [Download Area](#)
-  [Project Memberlist \(10 members\)](#)
-  [Project Member GPG Keyring](#)

Latest News

Stable CVS Version 1.11.23 Released!

posted by dprice, Thu 08 May 2008 11:56:21 AM UTC - 0 replies

Stable CVS 1.11.23 has been released. Stable releases contain only bug fixes from previous versions of CVS. This version includes an efficiency fix that reduces checkouts of very old revisions from an $O(n^2)$ operation to an $O(n)$ one, as well as ...

Class

Concurrent Versions System (CVS)

- Developer(s): The CVS Team
- Initial release: November 19, 1990; 27 years ago
- **Stable release: 1.11.23 / May 8, 2008; 9 years ago**
- Written in: C
- Operating system: Unix-like, Windows
- Type: Revision control
- License: GNU General Public License
- Website: savannah.nongnu.org/projects/cvs



Class

Concurrent Versions (Versioning) System (CVS)

- Free software client-server revision control system in the field of software development
- A version control system keeps track of all work and all changes in a set of files, and allows several developers (potentially widely separated in space and time) to collaborate
- CVS uses a client - server architecture: a server stores the current version(s) of a project and its history, and clients connect to the server in order to "check out" a complete copy of the project, work on this copy and then later "check in" their changes
- No new releases since 2008

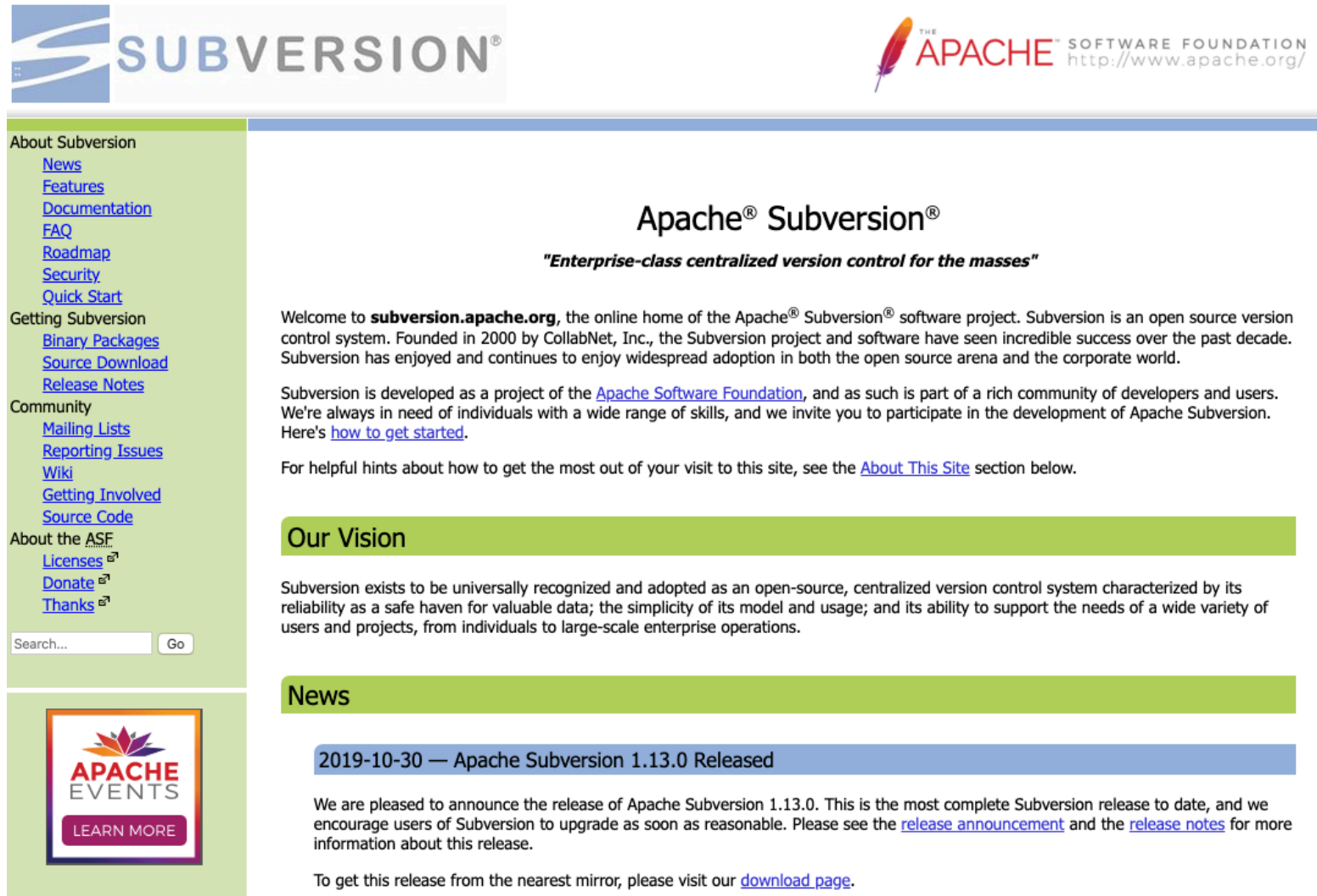
Class

Concurrent Versions (Versioning) System (CVS)

- “I created CVS to be able to cooperate with my students, Erik Baalbergen and Maarten Waage, on the ACK (Amsterdam Compiler Kit) C compiler. The three of us had vastly different schedules (one student was a steady 9–5 worker, the other was irregular, and I could work on the project only in the evenings). Their project ran from July 1984 to August 1985. CVS was initially called cmt, for the obvious reason that it allowed us to commit versions independently.”
- From: <http://dickgrune.com/Programs/CVS.orig/#History>

Class

Apache Subversion (SVN)



The screenshot shows the Apache Subversion website. At the top left is the Subversion logo, and at the top right is the Apache Software Foundation logo with the URL <http://www.apache.org/>. The left sidebar contains a navigation menu with links for 'About Subversion' (News, Features, Documentation, FAQ, Roadmap, Security, Quick Start), 'Getting Subversion' (Binary Packages, Source Download, Release Notes), 'Community' (Mailing Lists, Reporting Issues, Wiki, Getting Involved, Source Code), and 'About the ASF' (Licenses, Donate, Thanks). Below the menu is a search bar and a 'Go' button. At the bottom left is a banner for 'APACHE EVENTS' with a 'LEARN MORE' button. The main content area features the title 'Apache® Subversion®' with the tagline 'Enterprise-class centralized version control for the masses'. It includes a welcome message, a description of the project, and a link to 'how to get started'. Below this is a section titled 'Our Vision' and another titled 'News'. The 'News' section has a blue header for '2019-10-30 — Apache Subversion 1.13.0 Released' and contains text about the release, including links to the 'release announcement' and 'release notes'. It also provides a link to the 'download page' for the nearest mirror.

Apache® Subversion®
"Enterprise-class centralized version control for the masses"

Welcome to **subversion.apache.org**, the online home of the Apache® Subversion® software project. Subversion is an open source version control system. Founded in 2000 by CollabNet, Inc., the Subversion project and software have seen incredible success over the past decade. Subversion has enjoyed and continues to enjoy widespread adoption in both the open source arena and the corporate world.

Subversion is developed as a project of the [Apache Software Foundation](#), and as such is part of a rich community of developers and users. We're always in need of individuals with a wide range of skills, and we invite you to participate in the development of Apache Subversion. Here's [how to get started](#).

For helpful hints about how to get the most out of your visit to this site, see the [About This Site](#) section below.

Our Vision

Subversion exists to be universally recognized and adopted as an open-source, centralized version control system characterized by its reliability as a safe haven for valuable data; the simplicity of its model and usage; and its ability to support the needs of a wide variety of users and projects, from individuals to large-scale enterprise operations.

News

2019-10-30 — Apache Subversion 1.13.0 Released

We are pleased to announce the release of Apache Subversion 1.13.0. This is the most complete Subversion release to date, and we encourage users of Subversion to upgrade as soon as reasonable. Please see the [release announcement](#) and the [release notes](#) for more information about this release.

To get this release from the nearest mirror, please visit our [download page](#).

Class

Apache Subversion (SVN)

- Original author(s): CollabNet
- Developer(s): Apache Software Foundation
- Initial release: 20 October 2000; 17 years ago
- **Stable release: 1.11.0 (Nov 30, 2018; 11 months ago)**
- Development status: Active
- Written in: C
- Operating system: Cross-platform
- Type: Revision control
- License: Apache License version 2.0
- Website: subversion.apache.org

Class

Apache Subversion

- Software versioning and revision control system distributed as open source under the Apache License
- Software developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation
- Its goal is to be a mostly compatible successor to the widely used Concurrent Versions System (CVS)
- The open source community has used Subversion widely: for example in projects such as Apache Software Foundation, Free Pascal, FreeBSD, GCC and SourceForge
- Subversion was created by CollabNet Inc. in 2000, and is now a top-level Apache project being built and used by a global community of contributors

Git

안전함 | https://git-scm.com

앱

Web

Tech


Research

Future

Fun

Book

수업자료




git


--distributed-is-the-new-centralized

Search entire site...

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.


Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient [staging areas](#), and [multiple workflows](#).

 Learn Git in your browser for free with [Try Git](#).




About

The advantages of Git compared to other source control systems.




Documentation

Command reference pages, Pro Git book content, videos and other material.



Community


Get involved! Bug reporting, mailing list, chat, development and more.





[Pro Git](#) by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).


Free Good Book

Latest source Release
2.17.0
Release Notes (2018-04-02)
Downloads for Linux

 [Linux GUIs](#)

 [Mac Build](#)

 [Tarballs](#)

 [Source Code](#)

https://git-scm.com/doc

Class

Git

- Original author(s): Linus Torvalds
- Developer(s): Junio Hamano and others
- Initial release: 7 April 2005; 13 years ago
- **Stable release: 2.21.0 / 24 Feb 2018; 9 Months ago**
- Repository: github.com/git/git
- Development status: Active
- Written in: C, Shell, Perl, Tcl, Python
- Operating system: POSIX: Linux, Windows, macOS
- Platform: IA-32, x86-64
- Available in: English
- Type: Version control
- License” GNU GPL v2 and GNU LGPL v2.1
- Website: git-scm.com

Class

Git

- Git (/git/) is a version control system for tracking changes in computer files and coordinating work on those files among multiple people
- It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files
- As a distributed revision control system it is aimed at speed, and data integrity
 - As with most other distributed version control systems, and unlike most client - server systems, every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server

Class

Git

- Git was created by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development. Its current maintainer since 2005 is Junio Hamano
- Git is free and open source software distributed under the terms of the GNU General Public License version 2

Class

Git – Distributed Version Control

- Distributed version control (also known as distributed revision control) is a form of version control where the complete codebase – including its full history – is mirrored on every developer's computer
- Allows branching and merging to be managed automatically, increases speeds of most operations, improves the ability to work offline, and does not rely on a single location for backups
- Distributed revision control systems (DVCS) takes a peer-to-peer approach to version control, as opposed to the client – server approach of centralized systems
- Distributed revision control synchronizes repositories by exchanging patches from peer to peer – There is no single central version of the codebase; instead, each user has a working copy and the full change history

Class

Git – Distributed Version Control

- Central and branch repositories
 - Every project has a central repository that is considered as the official repository, which is managed by the project maintainers
 - Developers clone this repository to create identical local copies of the code base
 - Source code changes in the central repository are periodically synchronized with the local repository
 - The developer creates a new branch in his local repository and modifies source code on that branch
 - Once the development is done, the change needs to be integrated into the central repository

Class

Git – Distributed Version Control

- Pull requests
 - Contributions to a source code repository that uses a distributed version control system are commonly made by means of a pull request
 - The contributor requests that the project maintainer "pull" the source code change, hence the name "pull request"
 - The maintainer has to merge the pull request if the contribution should become part of the source base
 - The developer creates a pull request to notify maintainers of a new change; a comment thread is associated with each pull request
 - This allows for focused discussion of code changes

Class

Git – Distributed Version Control

- Pull requests (continued...)
 - Submitted pull requests are visible to anyone with repository access
 - A pull request can be accepted or rejected by maintainers
 - Once the pull request is reviewed and approved, it is merged into the repository
 - Depending on the established workflow, the code may need to be tested before being included into official release
 - Therefore, some projects contain a special branch for merging untested pull requests
 - Other projects run an automated test suite on every pull request, and the reviewer checks that any new code has appropriate test coverage

The world's leading software development platform

GitHub, Inc. [US] | <https://github.com>

Features Business Explore Marketplace Pricing

Search GitHub Sign in or Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Class

GitHub

- Type of site: Git-repository hosting service
- Available in: English
- Founded: February 8, 2008; 11 years ago
- Headquarters: San Francisco, United States
- Employees: 1079
- Website: github.com
- Users: 40 million (Dec 2019)
- Launched: 10 April 2008; 11 years ago
- Current status: Active
- Written in: Ruby

Class

GitHub

MICROSOFT BREAKING TECH

Microsoft completes GitHub acquisition

By Tom Warren | @tomwarren | Oct 26, 2018, 9:28am EDT



Microsoft revealed earlier this year that it's [acquiring GitHub for \\$7.5 billion](#).

Today, the software maker is [confirming that this big acquisition is complete](#).

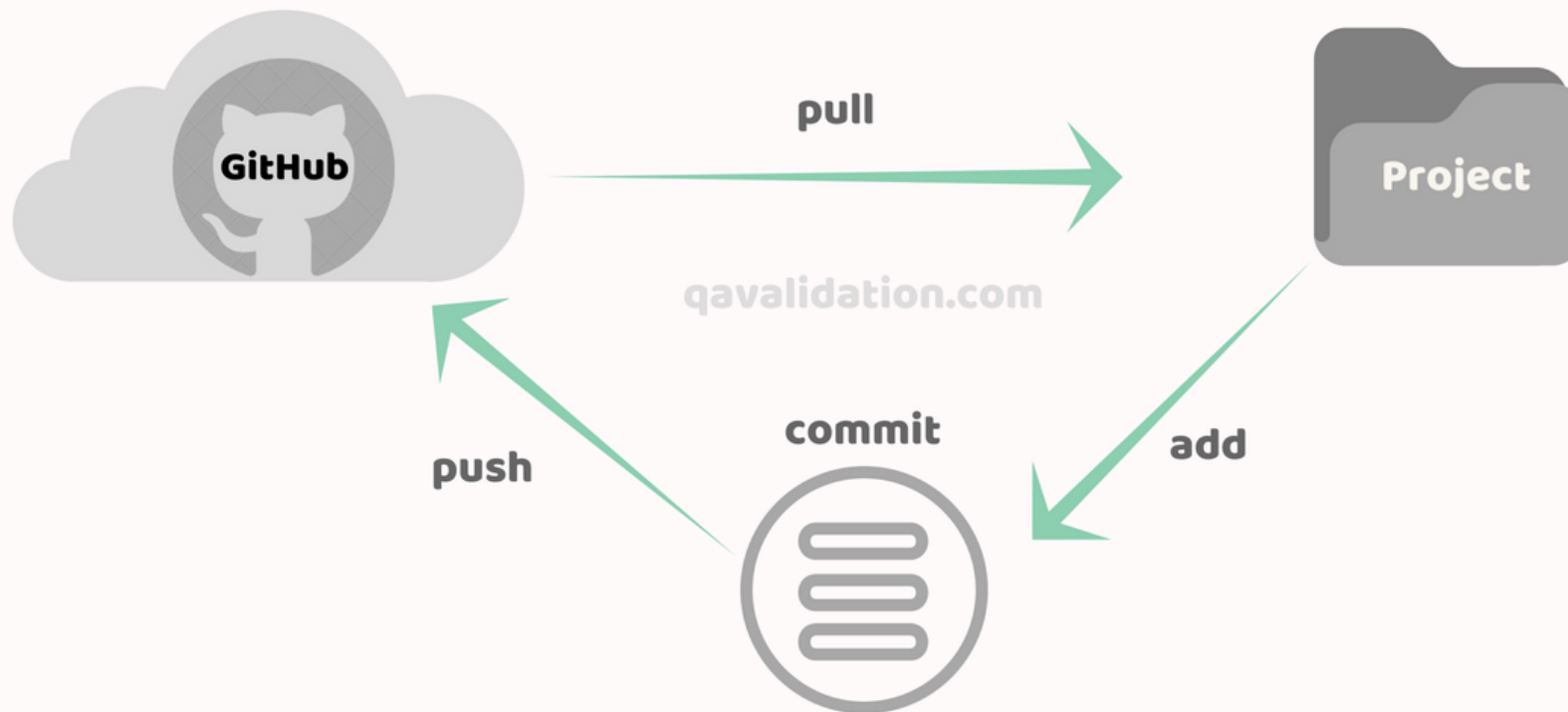
Microsoft will operate GitHub independently as a business, to keep its platform and community in place and to allow the service to “retain its product philosophy.”

As Microsoft's GitHub CEO, Nat Friedman, commented in a blog post announcing the completion of the deal, “Our vision is to serve every developer on the planet, by being the best place to build software. This is a dream opportunity for all of us at GitHub, and we couldn't be more excited to roll up our sleeves and start this next chapter.”

Class

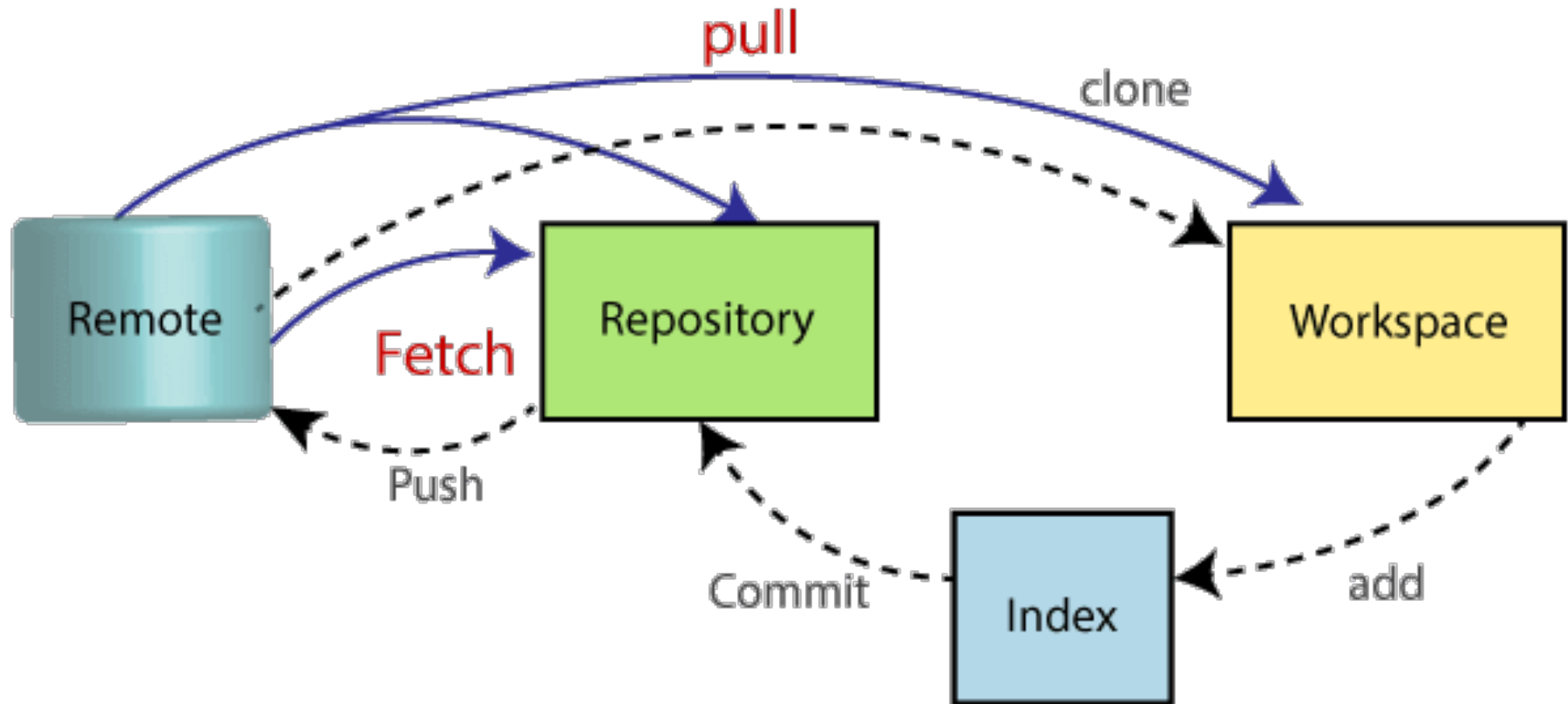
Git in GitHub

Git PUSH PULL



Class

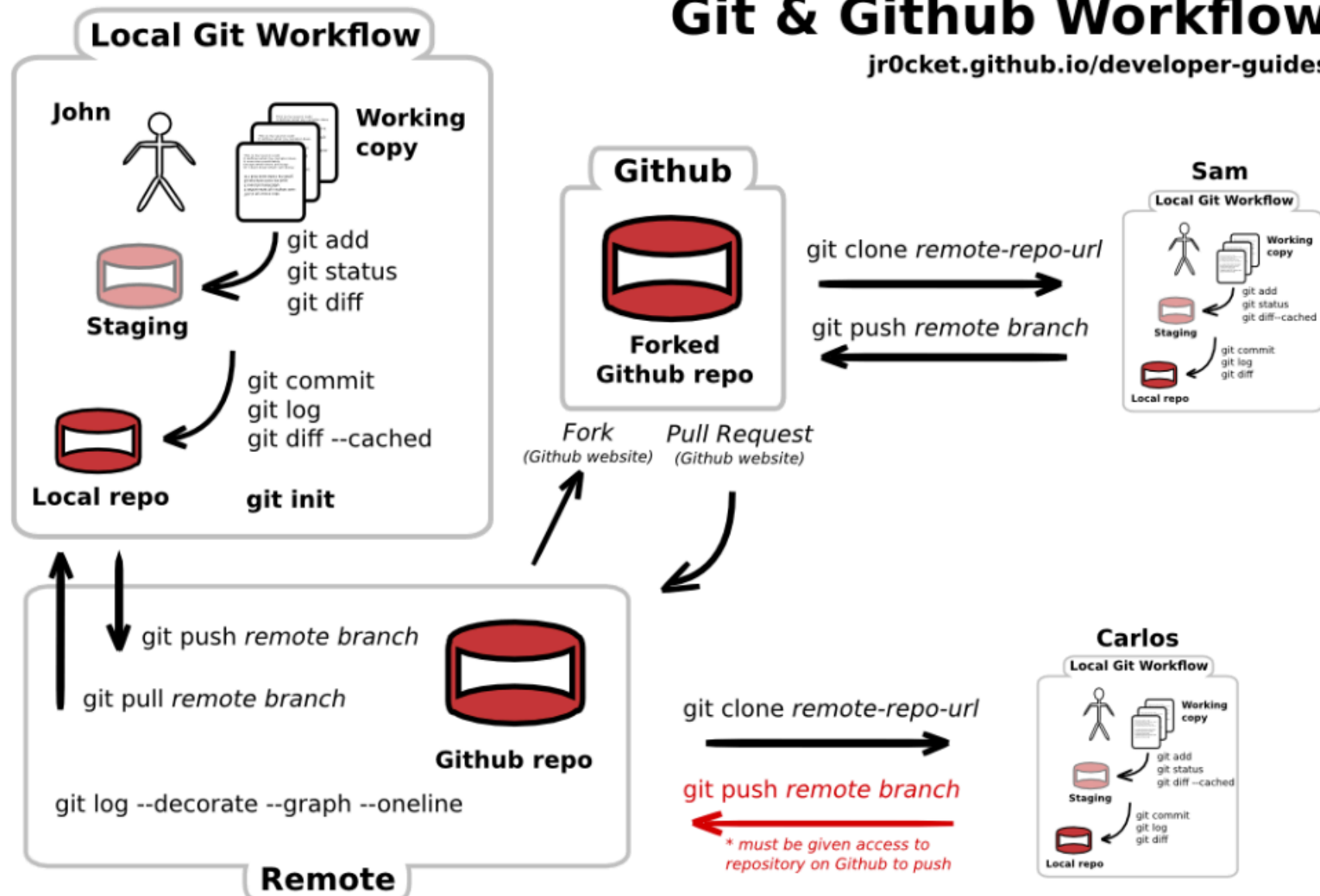
Git in GitHub



Class

Git & Github Workflow

jr0cket.github.io/developer-guides



Class

GitHub

- Web-based hosting service for version control using git
- It is mostly used for computer code
- It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features
- It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project
- As of April 2017, GitHub reports having almost 20 million users and 57 million repositories, making it the largest host of source code in the world

Class

GitHub Features

- Documentation, including automatically rendered README files in a variety of Markdown-like file formats
- Issue tracking (including feature requests) with labels, milestones, assignees and a search engine
- Wikis
- Pull requests with code review and comments
- Commits history
- Graphs: pulse, contributors, commits, code frequency, punch card, network, members
- Integrations Directory
- Unified and split diffs
- Email notifications

Class

GitHub Features (continued...)

- Option to subscribe someone to notifications by @ mentioning
- Emojis
- GitHub Pages: small websites can be hosted from public repositories on GitHub. The URL format is `https://username.github.io`
- Nested task-lists within files
- Visualization of geospatial data
- 3D render files that can be previewed using a new integrated STL file viewer that displays the files on a "3D canvas". The viewer is powered by WebGL and Three.js.
- Photoshop's native PSD format can be previewed and compared to previous versions of the same file.
- PDF document viewer

Class

GitHub as Programmer SNS & Portfolio Repository



Class

GitHub Student Developer Pack

GitHub Education

[Students](#)

[Teachers](#)

[Schools](#)

[Events](#)

[Get benefits](#)



[Home](#) / [Students](#) / GitHub Student Developer Pack

Learn to ship software like a pro.

There's no substitute for hands-on experience. But for most students, real world tools can be cost-prohibitive.

That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

[Get the Pack](#)

[Tweet](#)

[Like 56K](#)

Before you receive access to the offers, we'll need to [verify that you are a student](#).

Teachers, researchers, faculty, and staff are not eligible for the Pack, but can [get free and discounted access to GitHub](#).

Class

GitHub Student Developer Pack

- GitHub launched a new program called the GitHub Student Developer Pack to give students free access to popular development tools and services
- GitHub partnered with Bitnami, Crowdfunder, DigitalOcean, DNSimple, HackHands, Namecheap, Orchestra, Screenhero, SendGrid, Stripe, Travis CI and Unreal Engine to launch the program

Summary

CVS → SVN → Git & GitHub

GitHub for

- Access to global open source softwares

- Make portfolio for your career management

- Contribution to global projects

- Enjoy rich student developer benefits

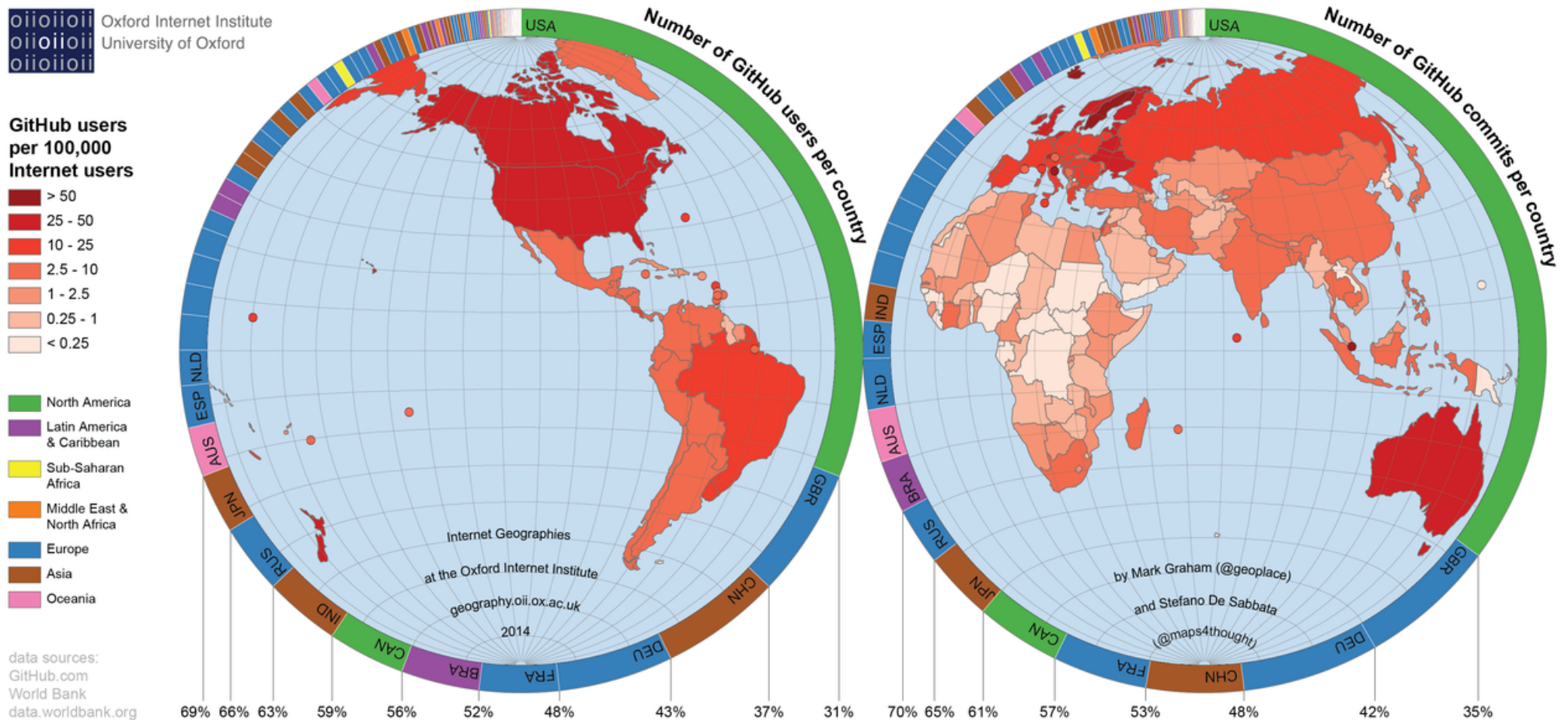
Reference

“소셜 코딩으로 이끄는 GitHub 실천기술”
오오츠카 히로키 지음, 제이펍

“GitHub Guides”
<https://guides.github.com/>

“Cheat Keys in GitHub”
<https://education.github.com/git-cheat-sheet-education.pdf>

Discussion



GitHub | Mapping collaborative software

Homeworks

“Hello World” via GitHub

<https://guides.github.com/activities/hello-world/>

Pre-Homeworks for Lecture 9

Visit news and videos,

List the tools mentioned in the news and videos.