Introduction to Team Development

Proprietary vs Open-Source

Open Source

Public

No direct management

Derived projects

Proprietary

Private

Rigorous management

Market share, competition

Intellectual property protection

Types of open-source licenses

Permissive licenses

Provide software as-is, with no warranties

Use and alter as you wish, at your own risk

Permissive Licenses	Copyleft Licenses
Berkley Software Distribution(BSD) MIT Apache 2	GNU Public License (GPL) Mozilla Public License (MPL) Eclipse Public License (EPL) Common Development and Distribution License (CDDL)

Types of open-source licenses

Copyleft Licenses

Extra requirements to the permissive license

Source code must be included in distributions of binaries

Permissive Licenses	Copyleft Licenses
Berkley Software Distribution(BSD) MIT Apache 2	GNU Public License (GPL) Mozilla Public License (MPL) Eclipse Public License (EPL) Common Development and Distribution License (CDDL)

Types of open-source licenses

Copyleft Licenses

Source code must be available under same terms under which code was originally obtained

Additional restrictions cannot be placed upon licensee's exercise of the license

Permissive Licenses	Copyleft Licenses
Berkley Software Distribution(BSD) MIT Apache 2	GNU Public License (GPL) Mozilla Public License (MPL) Eclipse Public License (EPL) Common Development and Distribution License (CDDL)

Version Control

Version control systems manage changes to codebases

Keep track of what, by whom, when, and why changes were made

Useful for individuals, but especially useful for teams where multiple people are making changes at the same time

Git is the most popular version control system today

Git

Free & open-source, installed locally

Git code repository can be local to a single computer, or synced up to an online repository for wider access

Multiple remote, cloud-based repositories available, including:

GitHub, GitLab, BitBucket

These are third-party services built on top of Git technology

Git can be used in both open-source & proprietary projects

Summary of basic Git CLI commands

Creating a local repository in Git:

Create a repository: ***git init***

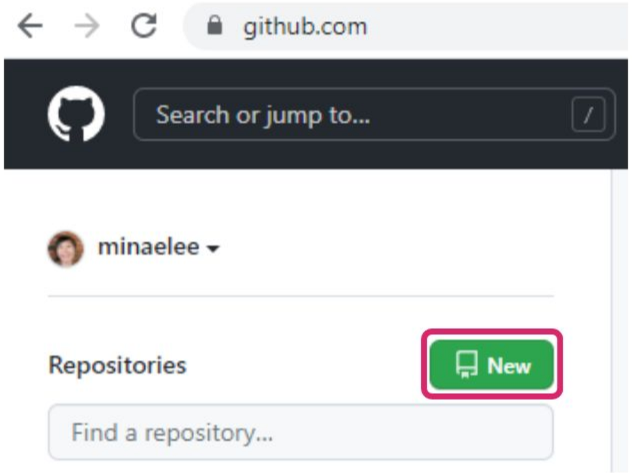
Clone a repository: ***git clone [url]***

Add files to a repository's staging area:
git add [file]

Commit files to a repository:
git commit –m "[descriptive commit message]"

Summary of basic Git CLI commands

Creating a remote repository in GitHub:



Contributing to team projects

1. Clone repository: ***git clone [url]***

2. Develop, test, and commit:
git add [filename], git commit –m [message]

3. Create a branch: ***git branch [branch name]***

4. Push to forked repository: ***git push [branch name]***

5. Send a pull request: ***git pull***

6. Maintainer reviews and merges: ***git merge [branch name]***

Infrastructure & tools for team development

Source Code Management and Version Control (e.g. Git)

Bug/Issue tracking (e.g. Jira, GitHub)

Continuous integration (GitHub Actions, BuildBot, Jenkins)

Mailing lists

GitHub and GitLab provide popular all-in-one solutions