OSPO Explainer Transcript: Version Control and Releasing Code

URL: https://ospo.library.jhu.edu/ospo-explainers/

Slide 1: Hello, and welcome to the Johns Hopkins University Open Source Programs Office Explainer Series, a collection of bite-sized videos tackling a variety of open-source essentials. Want to suggest a new topic? Drop us a line at ospo@jhu.edu to make a request. Today's explainer? Version Control and Releasing Code.

Slide 2: A *version control system* is a combination of technologies and practices for tracking and controlling changes to a project's files, in particular to source code, documentation, and web pages.

Slide 3: If you're writing code and saving it locally - even if it's just one script - it is worth it to move to a version control system. Version control can: track changes to your code over time; help you experiment without fear of breaking things; and allow others to see and cite your exact methods.

Slide 4: From Karl Fogel's book "Producing Open Source Software" - "If you don't already have an opinion about which version control system your project should use, then choose Git, and host your project's repositories at GitHub." *Git* is the de facto version control system standard in open source. The *GitHub* hosting platform provides free hosting and convenient tooling that is built around Git. GitHub allows you to work privately as long as you need and easily transition your project to publicly viewable when you're ready to share or accept contributions.

Slide 5: New to Git and GitHub? There are lots of great resources at Hopkins and beyond to help you get started:

- The OSPO provides access to GitHub Enterprise at no cost, with easy signup on our website: https://ospo.library.jhu.edu/services/github-enterprise/
- JHU Data Services provides Git and GitHub workshops each semester: https://dataservices.library.jhu.edu/training-workshops/
- GitHub provides free training resources: https://docs.github.com/en/get-started/start-your-journey/git-and-github-learning-resources and https://github.com/git-guides
- The Software Carpentries self-directed Version Control with Git for novices: https://swcarpentry.github.io/git-novice/

Slide 6: Once your code is in GitHub or another hosting platform like GitLab or Bitbucket, you can help make it more discoverable and citable by using **releases.** A release is a snapshot of your code at a specific point in time.

Slide 7: Software projects may handle release management in a variety of ways depending on their needs and the size or complexity of their project.

For larger projects, we recommend following a release guide such as the one in Karl Fogel's *Producing Open Source Software* (https://producingoss.com).

Smaller projects can follow the simpler set of steps shared on the next few slides.

Slide 8: For smaller projects, it's a good idea to create a release:

- Before submitting a paper
- After fixing important bugs or making significant code changes
- When sharing code with collaborators
- For major milestones in your analysis

Even if you're not planning on further changes to your code, a release gives your code an identifiable version that you and others can reference precisely.

Slide 9: If you're using GitHub, follow these steps to create a release:

- Go to your repository and click on **Releases** in the right sidebar
- In Releases, select **Draft a new release**
- Click on the **Tag** field, and enter a tag/release number for your version.
- Give your release a title and a description. Your description should include what this version does, changes since the last release, which paper/analysis it supports (if any), and updated installation/usage instructions.
- Hit **Publish**!

Slide 10: A note about tagging / version numbers

- If you'll only have one release, you can tag it v1.0.0
- If you're planning on ongoing development, consider **semantic versioning**, which follows the pattern: major.minor.patch (e.g. v2.3.6)
  - Increment the first number and set the second and third to zero if you have "breaking" changes that require users to modify code to upgrade, e.g. 2.3.6>3.0.0
  - Increment the second number and set the third to zero if you've added new features that don't break existing functionality, e.g. 2.3.6>2.4.0
  - Increment just the third number for bug fixes and small improvements without new features, e.g. 2.3.6>2.3.7

Slide 11: If you're using a different hosting platform for your code, check the documentation for those platforms for instructions on creating a new release:

GitLab: https://docs.gitlab.com/ee/user/project/releases/

Bitbucket: https://support.atlassian.com/bitbucket-cloud/docs/use-tags-and-releases/

SourceForge: https://sourceforge.net/p/forge/documentation

Slide 12: Our best piece of advice is, don't wait for the "perfect" project - every project can benefit from version control, no matter how small or personal. Take the first step and create a repository for the code you're working on right now!

Slide 13: Have any questions about what you've just learned? Ask the JHU Open Source Programs Office for more information. You can reach out to us online at ospo.library.jhu.edu or send an email to ospo@jhu.edu.