Science librarian training - Git & GitHub

Lesson - outline

1. **Today's goal:** To introduce Git and GitHub; to familiarize librarians with major concepts and terms; to demonstrate a cycle of contributing to a project; to give librarians an opportunity to commit a change to GitHub. (Rick)
    a. Ask audience: "How many of you have used Git or GitHub? How many have used it via the command line?
2. **What is version control? Why use it?** (Rick)
    a. Show Bret Davidson's slides 3-6 here.
    b. Add these ideas:
        i. Three very common types of version control:
            1. Undo command
            2. Track changes in Word
            3. Google Docs history.
        ii. Can use GitHub to contribute to a collaborative project, or to start your own project from an existing one
3. **What are Git and GitHub?** (Rick)
    a. Git is software for tracking and managing changes to a collection of files (a "repository")
    b. GitHub is a company that provides tools for using Git. Two main ones:
        i. GitHub web hosting service
        ii. GitHub Desktop, an application for using GitHub on your computer.
    c. The GitHub web service also provides tools for managing Git-based projects:
        i. Issues-tracker
        ii. Wiki
        iii. Membership management
        iv. Social media
        v. Activity graphs/visualizations
4. **What's going on? An analogy**: Mastering the cake recipe (Jamie)
    a. Terms to use:
        i. *Repository*
        ii. *Clone*
    b. Other concepts to introduce/demonstrate:
        i. *Pull request*
        ii. That a person moderates pull requests to the repository
5. **Set-up**:
    a. Download and install GitHub Desktop - we will help (Rick leads, all help librarians at their computers)

b. Clone our existing science-librarian-training repository to your GitHub Desktop:
    i. Browse to the UCBerkeleyRDM GitHub repository.
    ii. Click the science-librarian-training link
        1. What's all this? Brief tour of the GitHub repository page…
            a. Repository contents: Folder and a readme file
            b. README file display - GitHub can preview certain types of file right on screen.
            c. Most recent "commit," or change to repository. More on that later.
            d. Tabs along top: Code, issues, wiki...
    iii. Click "Clone or Download" (green) button and select "Open in Desktop"
    iv. Clone the repository to the desktop of your computer; a folder will be created there.

6. **A first look at GitHub Desktop** (which is now open on your screen)
    a. GitHub is a version control system. We can see every version of our files listed in a column.
        i. Creation of the first file in the repository is at the bottom of column. (click)
        ii. The contents of the original version of the file README.md appear to the right
        iii. Some information about this version of the file appears above the content:
            1. Created by Jamie, four days ago, etc. - This is the "commit", the record of the change.
                a. The top line is a description of the commit: "Create README.md"
        iv. Most recent change to the repository is at the top of the column. (click)
            1. Jamie has added a commit comment: "Update workshop agenda 1."
            2. We can see the contents of the file.
                a. The line in green highlight has been added.
                b. If Jamie were to have deleted a line, it would appear in red.
                c. This view is called the "diff" - it shows the difference between this version and the previous one.
        v. The column that displays every commit to the repository is called the "history" or "log".

7. So where are my files?
    a. Return to GitHub repo. Click 'Workshop 1' folder. Reveal 'Workshop 1 Agenda' file, the README file for this folder, and a "Bibliographic data" folder.
    b. Use diagram to explain relationship between GitHub repository, GitHub desktop repository, and files on disk.
        i. Draw an arrow connecting the repo on GitHub to the one on GitHub Desktop. Label the arrow "clone".
    c. The 'Workshop 1 Agenda' file that we've seen in the log is this 'Workshop 1 Agenda' file. So how do we edit it?

d. Have librarians find the files on the computer. (HINT: They are in the folder created on their desktop when we copied ("cloned") the GitHub repo to GitHub Desktop.
e. A few terms (define each):
    i. Local repository
    ii. Remote repository
    iii. Origin

8. **Demonstrate the full cycle of making a change:** Edit 'Workshop 1 Agenda'
    a. Sync first, so you know you are working with the most recent copy of the file. Sync pulls the most recent version of the repository from GitHub to your GitHub Desktop.
    b. Create a new branch -- 'Rick's branch'
        i. Describe the screen that appears
            1. No changes...yet. We will see how this changes in a moment.
            2. Graph at top comparing Master branch to Rick's branch
        ii. Define terms:
            1. Branch
            2. master branch
            3. Working branch (aka feature branch, topic branch)
    c. Make an edit to 'Workshop 1 Agenda' file
        i. Find the file in finder, open it in text editor
        ii. Make a change.
        iii. Save.
    d. Review GitHub Desktop screen - describe what's changed
        i. '1 change'
        ii. File listed
        iii. Diff of file
        iv. Summary box: describe what I did:
            1. "Modify agenda"
    e. Commit this change to Rick's branch in my local repository
    f. Review screen now.
        i. Empty Uncommited changes screen.
        ii. Click History tab: there's our commit.
        iii. Notice how graph has changed:
            1. There's now an event -- a circle -- on Rick's branch that is not yet in the master.
    g. Publish your changes.
    h. Make a pull request
        i. OK commit message.
        ii. Add description if desired.
        iii. Note: from Rick's branch to master
        iv. Click Send pull request button.
    i. Switch to GitHub repo in browser

        i. Show and click Pull request button
        ii. See that the pull request is listed. Click its name
        iii. There's my commit.
        iv. Auto merge ok?
        v. Accept pull request
        vi. Add a comment, if you want
9. **Now, librarians get to make a change.**
    a. In Bibliometric data folder there's a csv file.
        i. Show on GitHub.
        ii. Note how GitHub can preview the file
        iii. Review file. Show librarian names.
    b. Find file on local computer (all)
    c. ASSIGNMENT: Librarians each edit file to add their middle name or initial.
    d. Make pull request.
    e. Show resulting pull requests on GitHub.
10. **Questions, discussion**
11. Fin