

Git It Together:

“An Introduction to Version Control for Revit Users”

John Pierson

Design Technology Specialist
Parallax Team



git it together

“VERSION CONTROL FOR BEGINNERS”

WWW.PARALLAXTEAM.COM

Summary

Tool customization, automation, data mining, and design iteration exploration are a few examples of why you should learn to code in the architecture, engineering, and construction (AEC) industry. Coding includes the use of text-based languages such as Python or C#, or visual programming environments such as Dynamo and Grasshopper. With the use of these tools come additional methodologies that we need to learn, such as version control or Git. You've probably heard of this process through GitHub, which is a version control platform (Dynamo is hosted on GitHub). **Essentially, version control lets you manage revisions regarding your code.** The problem with Git (version control) is that this process isn't introduced in a way that makes the most sense to a Revit user. That is where this class comes in: **We'll introduce the process and key terms, and explain why version control matters to you.** After this class, attendees will have an introductory understanding of how to use Git, and will be able to manage their files using a version control process.

Learning Objectives

- **Gain** an introductory understanding of version control processes.
- **Understand** key Git terms.
- **Learn** how to manage your files using version control.
- **Learn** how to actively contribute to open-source projects.

About the Speaker



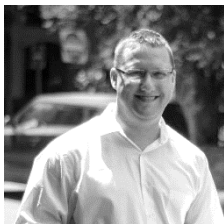
John Pierson

- **Design Technology Specialist** at Parallax Team
- **Reviteer** since 2012
- **Dynamo User** since 2014
- **Dynamo Package Developer** since 2015, (Rhythm, Bang!, DuctTape, Monocle).
- **BILTNA** Top Rated Speaker
- **Revit Certified Professional** for Architectural, Structural, MEP

The Core Team

Parallax Team is a Practice Implementation Consultancy specializing in implementation, computational design, and pretty much anything AEC related.

www.parallaxteam.com | @prlxteam



Aaron Maller
Director
@twiceroadsfool



John Pierson
Design Technology Specialist
@60secondrevit

Who are you?



What is Git?

- A Version Control Software – Meaning that it manages changes to a project without overwriting any part of that project.
- Created in 2005 by Linus Torvalds (linux dude)
- Essentially prevents user's writing over each other's files, or my personal favorite, you screwing up your code and saving over the working code.

Git is **awesome** because









- All versions are saved via a “snapshot”
- You can merge changes and review the differences of the same file that has been modified.
- Less losing work = more time for better things.

Why do we need to
know this?

Why do we need to know this?

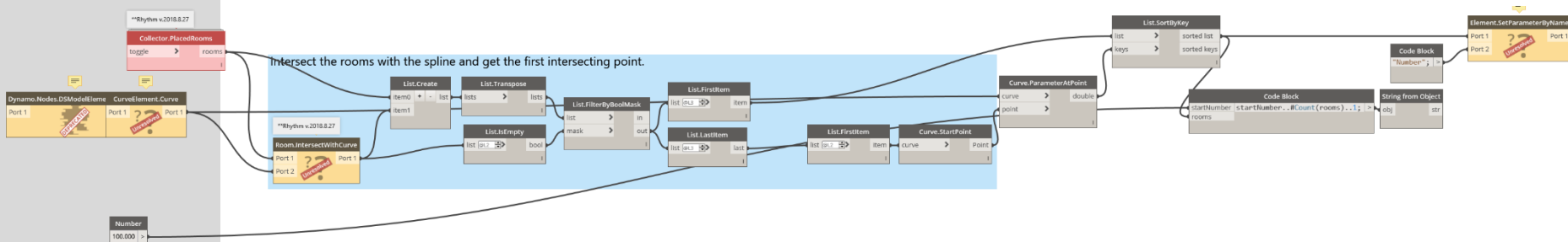
- Sometimes we suck at managing files.

rk > Prlx2 > Marketing > Lectures > 2018- AU- GitItTogether > SampleFiles > BadExample

Name	Date modified	Type	Size
 roomRenumber - Rev2(working) - Final.dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB
 roomRenumber - Rev2(working).dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB
 roomRenumber - Rev1 - testing.dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB
 roomRenumber - Final(ForReal).dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB
 larchive-roomRenumber.dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB
 roomRenumber - Rev1 - testing - Copy.dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB
 roomRenumber - Rev1 - testing - Copy (2).dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB
 roomRenumber - Rev1 - testing - Copy (3).dyn	8/30/2018 8:06 AM	Dynamo Studio W...	16 KB

Why do we need to know this?

- We've all had the “But it worked yesterday” experience.



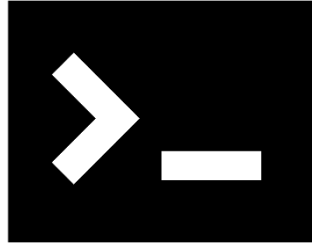
Why do we know

- We
wha

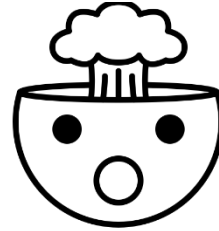
er that “does



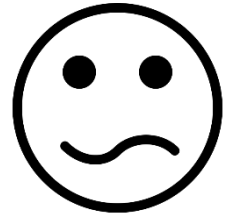
Why don't we all know how to use this?



command line
(primarily)

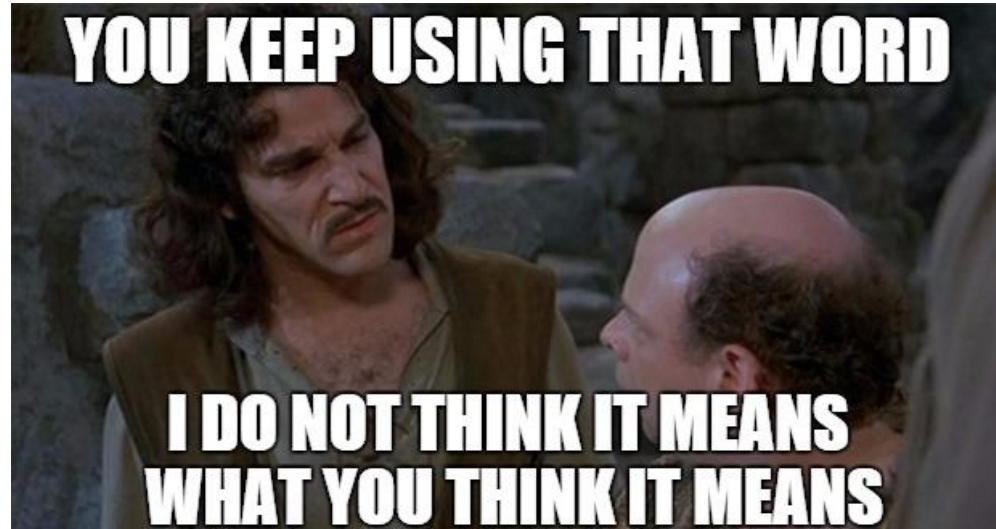


you don't know
what you don't
know



not the most
intuitive thing
with new terms
and processes

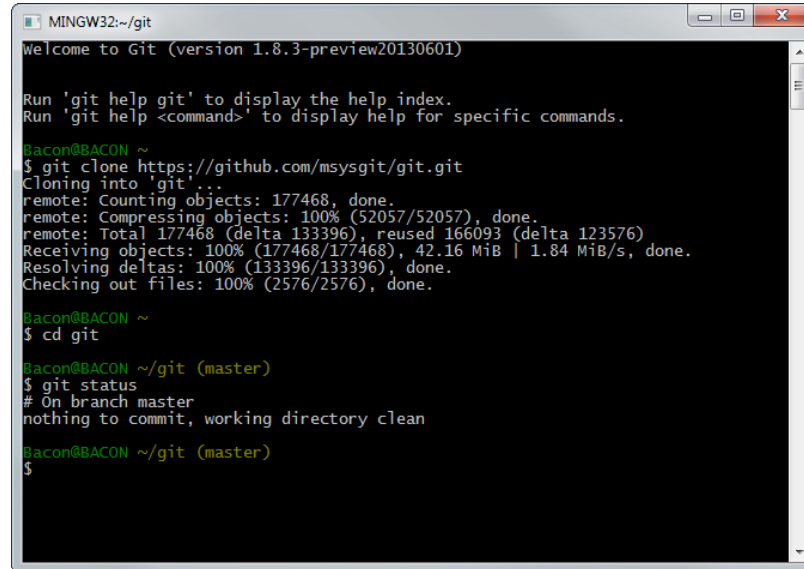
Key Terminology



Key Terminology

Command Line

Interface for text-based input for Git. (*not how we are going to be doing this today*)



```
MINGW32:~/git
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

Bacon@BACON ~/git (master)
$
```

<https://gitforwindows.org/>

Key Terminology

Repository/Repo

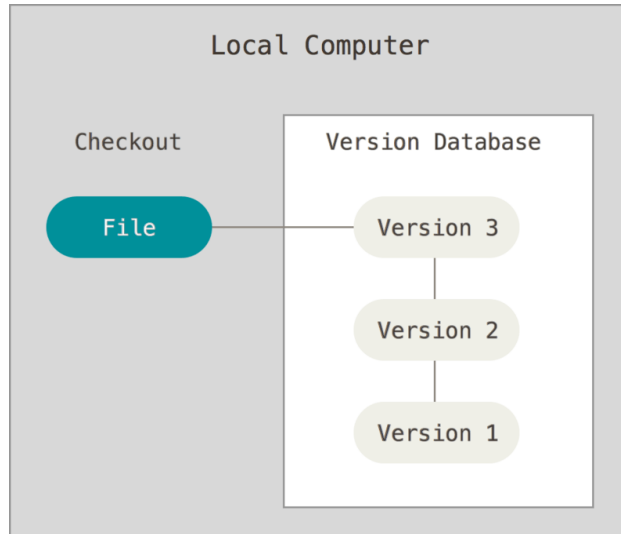
Storage space where your projects live.
(similar to a Revit central file)



Key Terminology

Version Control

Instead of manually saving multiple versions, Git saves “snapshots” of every point in time for the life of the project. (similar to Revit’s backup folder, but better.)



Key Terminology

Commit

This is the beautiful part. A commit is you pushing your changes to Git. Each time you commit to the repo, Git takes a snapshot to give you a checkpoint to go back to if needed. (similar to a sync with central, but also way better)

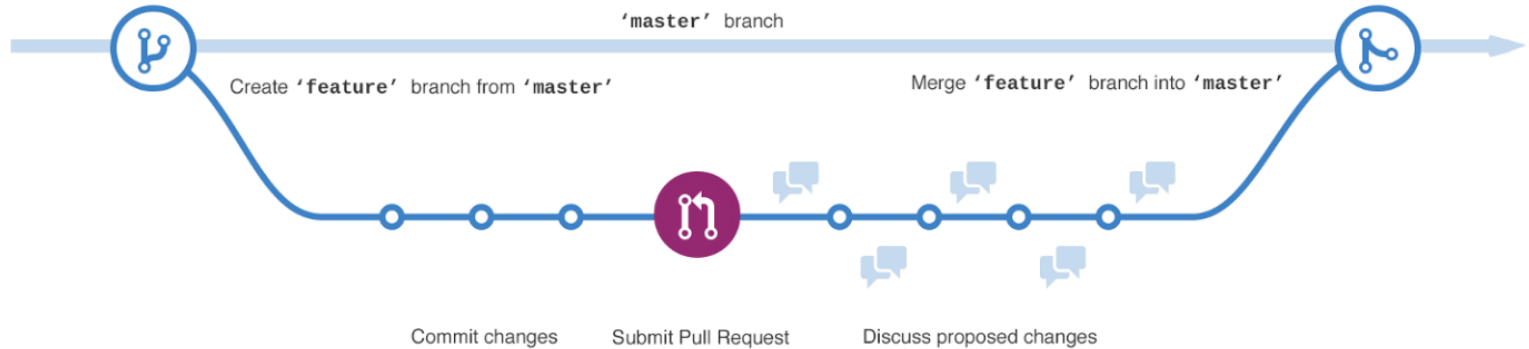


Sending your changes off to the repo!

Key Terminology

Branch

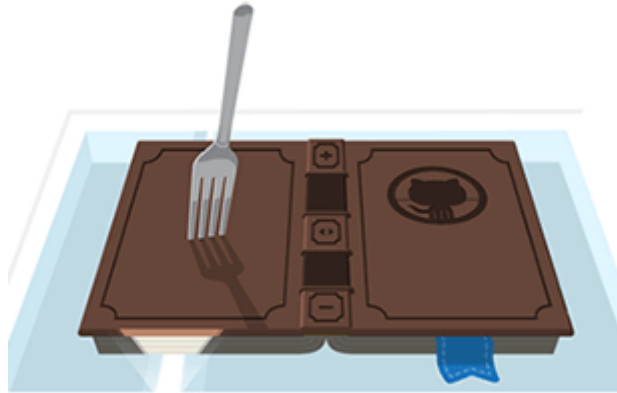
This allows people to work on their own version of the project before pushing the changes back to the main project. (similar to a Revit local file)



Key Terminology

Fork

Your very own copy of a project separate from the source project with the option to push your changes to the source if you want to.



Key Terminology

Pull Request










A way of submitting contributions to an open project. This is done by making changes on your fork and generating the pull request (PR).



Key Terminology

.gitignore

When you are managing a project with version control, there are definitely going to be files you don't want to “sync” to the repo. Rather than removing these files from the local repo we can tell github to ignore them by extension!

 .git	10/25/2018 9:52 AM	File folder	
 _dataset	8/14/2018 7:15 AM	File folder	
 _resources	10/25/2018 7:43 AM	File folder	
 Presentation	10/26/2018 1:03 PM	File folder	
 Presentation-Videos	10/26/2018 1:03 PM	File folder	
 SampleFiles	10/26/2018 1:04 PM	File folder	
 .gitignore	10/25/2018 7:52 AM	GITIGNORE File	1 KB
 LICENSE	10/25/2018 7:36 AM	File	2 KB
 README.md	10/25/2018 7:36 AM	MD File	1 KB

This AU Class was...
built using Github

This AU Class was... built using Github

The screenshot shows the GitHub repository page for `johnpierson/AU2018-GitItTogether`. The page includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and repository statistics (2 Unwatch, 1 Star, 0 Fork). The main content area displays the repository description, a list of commits, and a table of files.

https://github.com/johnpierson/AU2018-GitItTogether

Search or jump to...

Pull requests Issues Marketplace Explore

johnpierson / AU2018-GitItTogether

Unwatch 2 Star 1 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

This is the repository for the AU presentation on version control. Edit

Manage topics

11 commits 1 branch 0 releases 1 contributor View license

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Commit Message	Time Ago
johnpierson Update README.md	Latest commit 60cd3e4	6 minutes ago
Presentation-Videos	Home Image, various updates	16 minutes ago
Presentation	Home Image, various updates	16 minutes ago
SampleFiles/BadExample	Additions + Updates	4 days ago
_resources	Home Image, various updates	16 minutes ago
.gitignore	Recategorization	4 days ago
LICENSE	Create LICENSE	18 minutes ago
README.md	Update README.md	6 minutes ago

The landing page with all the data

#AU2018

Resources

- Markdown
 - <https://guides.github.com/features/mastering-markdown/>
 - <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
- Misc
 - This exact presentation's repo.
<https://github.com/johnpierson/AU2018-GitItTogether>
 - <https://guides.github.com/activities/hello-world/>
- License
 - Choose a license
<https://choosealicense.com/>
 - Creative Commons
<https://creativecommons.org/>

Now, Let's **Build** Some Workflows. (and use version control)

- Start a Github Project
- Establish ReadMe and .gitignore
- Choose a license
- Build cool stuff and manage it.
- Collaborate: *AKA Have other users mess with it.*

Workflows to Choose From



Read Family
Category
without Revit



Load View
Templates from
File



Modify Your (or
your coworker's) Revit
Ribbon

thank you.

NOW, GO DO AWESOME THINGS

WWW.PARALLAXTEAM.COM

@prlxteam | @twiceroadsfool | @60secondrevit



AUTODESK®

Make anything™

Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product and services offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2018 Autodesk. All rights reserved.

