



# Version Control with Git and GitHub

A Short Introduction



# What is version control?

- Aka revision control, source control
- Version control: the management and tracking of changes to source code, documents, data, etc.
- Allows collaborative development
- Keeps track of **who** made a change, **when** the change was made, and **what** the change was
- Permits reverting any change and rolling back to a previous state
- Many systems available: CVS, Subversion (SVN), Perforce, git, Mercurial,...



# What is git?

- **Distributed** revision control system
  - Speed
  - Data integrity
  - Distributed, non-linear work flows
- Created in 2005 by Linus Torvalds to support the Linux kernel development
- Main characteristics:
  - the entire code and history is kept on the client (user) machine
  - users can work (make changes to code) even without internet connection
  - internet connection required only for pushing and pulling from remote server (remote repository)





# Git basics – commits

- Git keeps track of code history in snapshots
  - record of what all files look like at a given point in time
- User decides when to take a snapshot (**commit**) and what files should be included
- Allows going back and visiting any past snapshot
  - later snapshots are not lost
- A project is made out of a series of commits
- Each commit contains:
  1. information on how the files changed from previous commit (**diff**)
  2. a reference to the previous commit (**parent commit**)
  3. a **hash code** name



# Git basics – repositories

- A **repository** (or ‘repo’) is a collection of all the files and their commit history
  - contains **all** commits
  - can be local or remote
- Copying a repository from a remote server is called **cloning**
- Cloning allows teams to develop collaboratively
- **Pulling**: downloading commits that do not exist on the local machine from a remote repository
- **Pushing**: adding local changes (commits) to a remote repository



# Git basics – branches

- All commits in a repository live in some **branch**
- The main branch in a repository is called the **master branch**
- A project can have many branches
  - For example, a project that follows GitFlow will have a master branch, a develop branch, feature branches, hotfix branches, release branches
- Branches allow maintaining parallel and separate development tracks in a single project
- Development tracks can be
  - branched off
  - merged

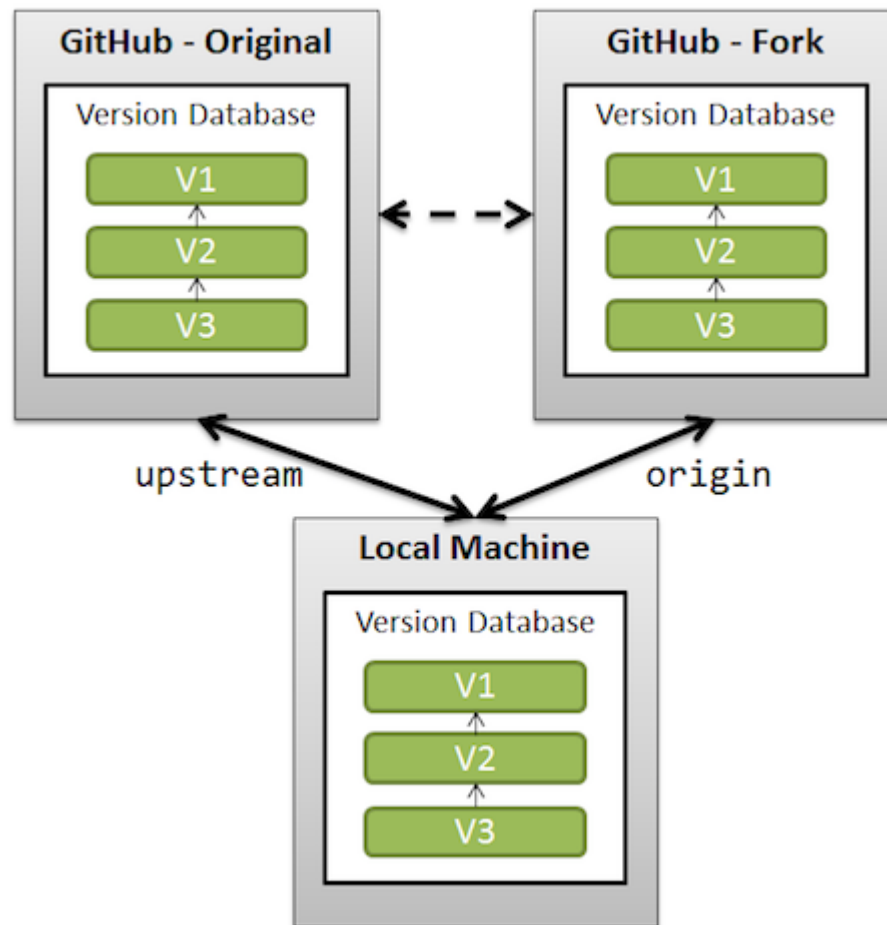


# What is GitHub?

- Largest web-based git repository hosting service <https://github.com/>
- Founded in 2008
- Promotes open source, but also has an Enterprise Edition for businesses
- Offers all Git distributed version control functionality
- Additional functionality:
  - User interface
  - Documentation
  - Bug tracking
  - Feature requests
  - Pull requests



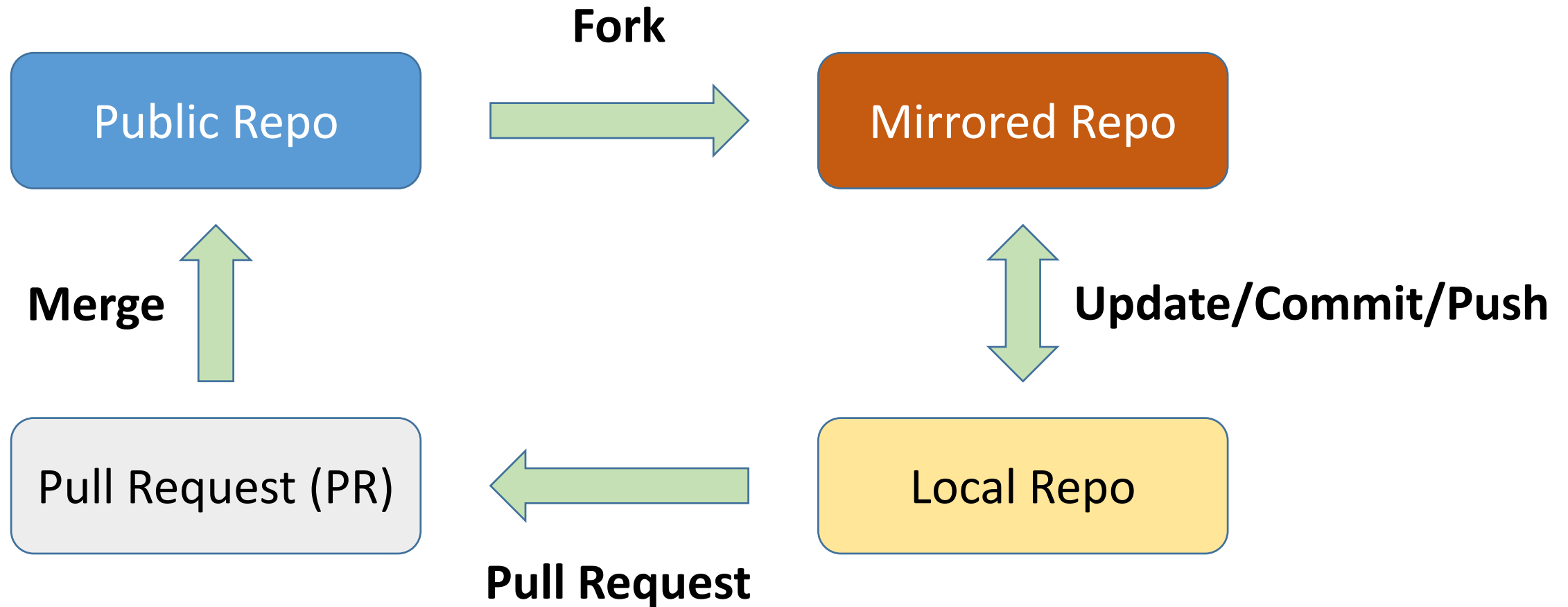
# GitHub basics – forking







# GitHub basics – pull requests



# SourceTree GUI client



The screenshot shows the SourceTree interface with several red callout boxes pointing to specific features:

- Current project:** Points to the 'chronos' repository tab at the top.
- Local branches:** Points to the 'develop' and 'master' branches in the left sidebar.
- Remote branches:** Points to the 'feature' branch under the 'origin' remote in the left sidebar.
- Log message for selected commit:** Points to the commit message 'Minor formatting fixes in co-simulation code (no functional changes)' in the commit log.
- Selected file in commit:** Points to the file path 'src/tests/vehicle/test\_HMMWV\_cosimulation/TerrainNode.cpp' in the file list.
- Commit hash code:** Points to the commit hash '3005aa7' in the commit log.
- Who made the commit?:** Points to the author 'Radu Serban <serban@wisc.edu>' in the commit log.
- When was the commit done?:** Points to the date 'Saturday, November 5, 2016 2:00:00 PM' in the commit log.
- Selected commit:** Points to the selected commit row in the commit log.
- File differences (wrt parent commit):** Points to the diff view showing code changes in 'TerrainNode.cpp'.

Graph	Description	Date	Author	Commit
●	Merge branch 'feat' into origin/HEAD	6 Nov 2016 7:39	Radu Serban <serban@wisc.edu>	d52dd49
●	Minor formatting fixes in co-simulation code (no functional changes)	5 Nov 2016 14:00	Radu Serban <serban@wisc.edu>	3005aa7
●	Merge branch 'develop' into feature/cosim_rigid_tire	5 Nov 2016 13:46	Radu Serban <serban@wisc.edu>	378e24e
●	Fix bug in full vehicle co-simulation code related to update of triangular s	5 Nov 2016 13:45	Radu Serban <serban@wisc.edu>	860c013
●	Merge branch 'develop' of https://github.com/projectchrono/chrono into	4 Nov 2016 16:18	Radu Serban <serban@wisc.edu>	302f8be
●	Add picture and documentation for Euler - Bernoulli beam	4 Nov 2016 15:54	Alessandro Tasora <tasora@wisc.edu>	38a517e
●	Fix a bug in the ABAQUS file loader that prevented loading tetrahedrons	4 Nov 2016 15:19	Alessandro Tasora <tasora@wisc.edu>	24f53ee
●	New page for FEA tutorials. Populated with many demo subpages	4 Nov 2016 15:17	Alessandro Tasora <tasora@wisc.edu>	a534625
●	New page in the manual, about finite elements. New subpages with the li	4 Nov 2016 15:16	Alessandro Tasora <tasora@wisc.edu>	8348735
●	More consistent cross hyperlinks	4 Nov 2016 15:15	Alessandro Tasora <tasora@wisc.edu>	59b364d
●	Added table of contents for large pages of manual	4 Nov 2016 15:14	Alessandro Tasora <tasora@wisc.edu>	475e503
●	Remove orphan page that polluted the "Related pages" in doxygen tabs	4 Nov 2016 15:13	Alessandro Tasora <tasora@wisc.edu>	21f6002
●	Fix bug where ANCF shell elements were double counted in contact surfa	4 Nov 2016 14:23	Radu Serban <serban@wisc.edu>	c1f793d
●	Fix bug in extracting contact material properties for deformable tires	4 Nov 2016 13:10	Radu Serban <serban@wisc.edu>	c7e0087

```
Commit: 3005aa763e5528b5926a0ddaf5ab1813da3fc23f [3005aa7]
Parents: 378e24ed9f
Author: Radu Serban <serban@wisc.edu>
Date: Saturday, November 5, 2016 2:00:00 PM

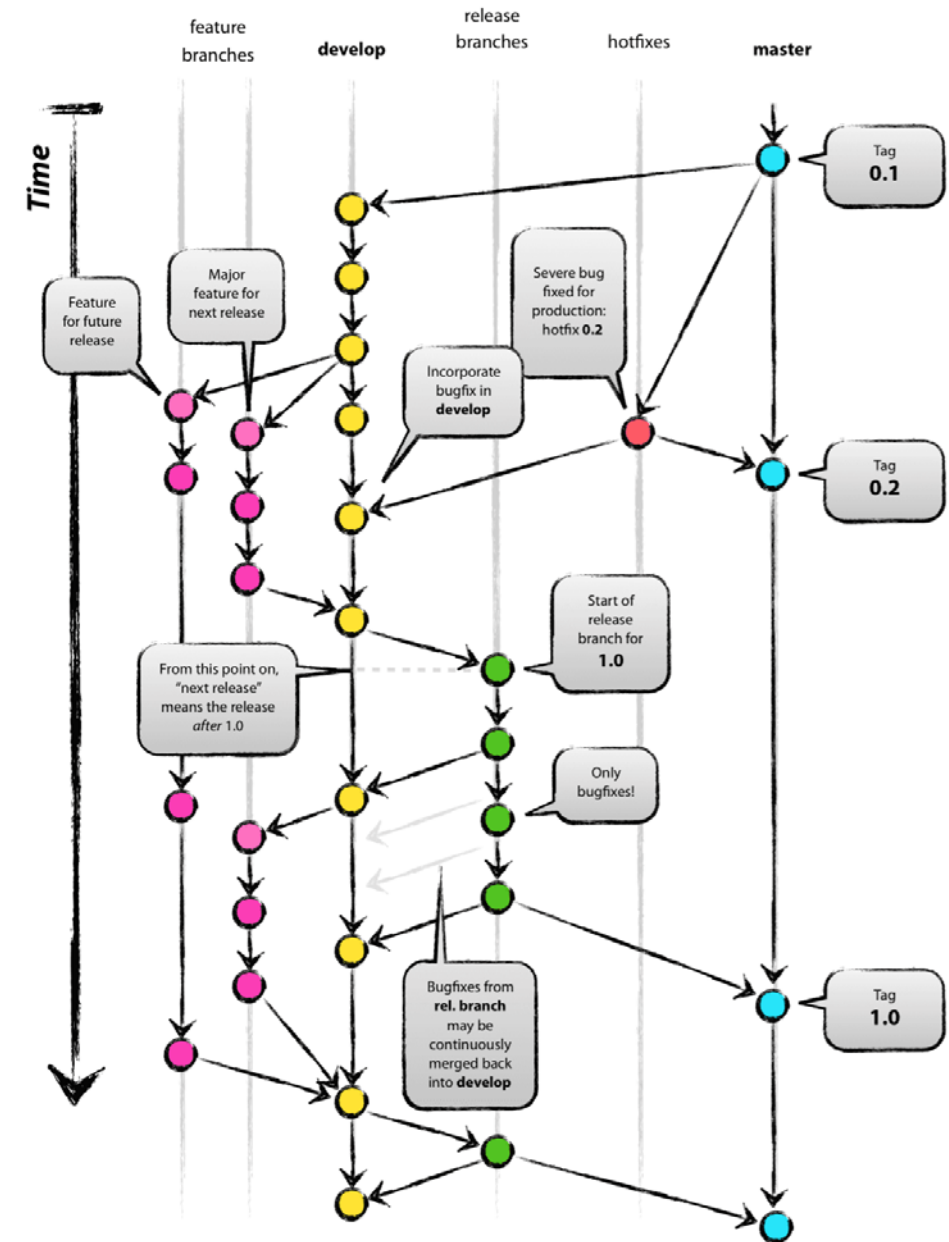
Minor formatting fixes in co-simulation code (no functional changes)

src/tests/vehicle/test_HMMWV_cosimulation/TerrainNode.cpp
src/tests/vehicle/test_HMMWV_cosimulation/TireNode.cpp
src/tests/vehicle/test_HMMWV_cosimulation/VehicleNode.cpp
src/tests/vehicle/test_HMMWV_cosimulation/test_VEH_HMMWV_Cosimulat...
```



# GitFlow

- Proposed by Vicent Driessen  
<http://nvie.com/posts/a-successful-git-branching-model/>
- A development model based on Git





# Git/GitHub resources

- Git official website: <https://git-scm.com>
- GitHub guides: <https://guides.github.com>
- Git tutorials and training: <https://www.atlassian.com/git/tutorials/>
- Git cheat-sheet: <https://services.github.com/kit/downloads/github-git-cheat-sheet.pdf>
- GetFlow cheat-sheet: <http://danielkummer.github.io/git-flow-cheatsheet/>
- Git GUI clients: many options (see <https://git-scm.com/download/gui/linux>)
  - Windows/Mac: Sourcetree (by Atlassian)
  - Linux/Windows/Mac: SmartGit, git-cola