

---

---

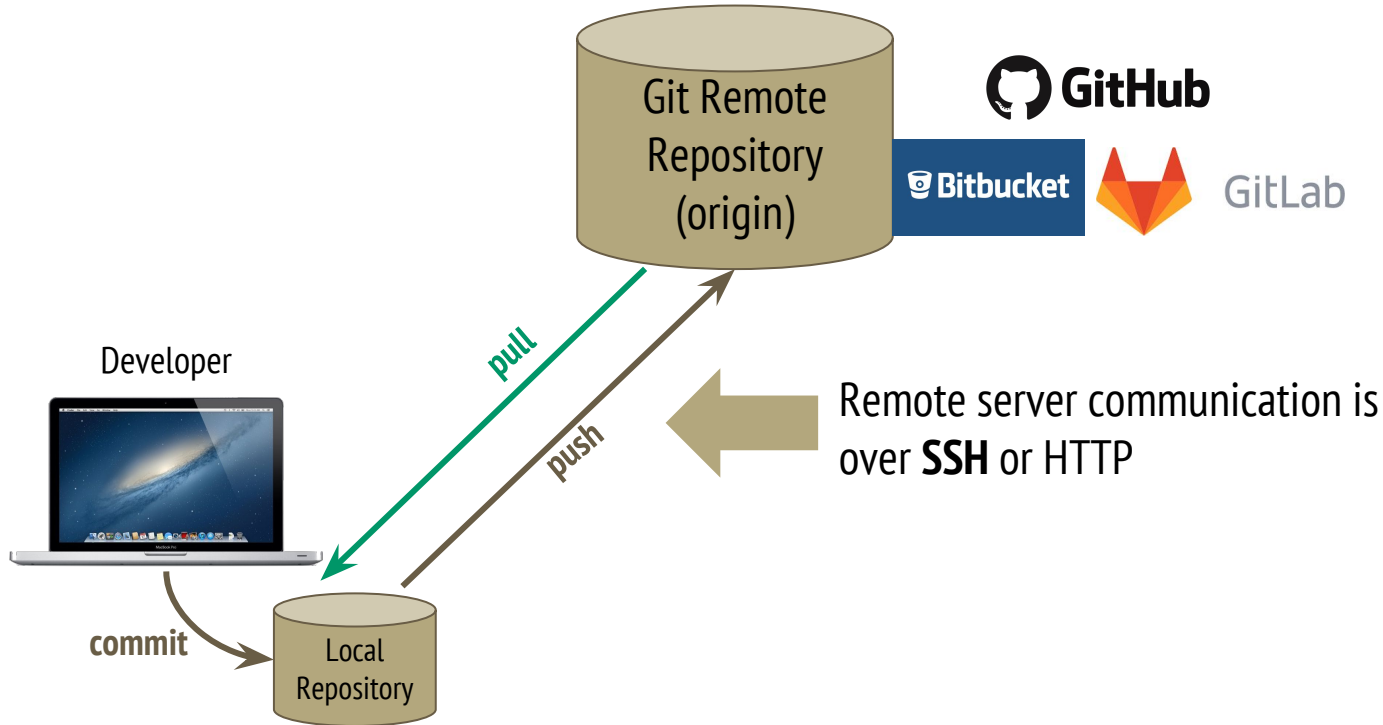
# **GIT Overview**

— By Sunit Parekh & Amit Agarwal —

---

---

# Git, a distributed version control system



# Git Authentication

- Default authentication with remote is SSH based
- User/password is possible, but hardly anyone uses

# Git Basic Commands

- init
- clone
- add, rm, mv
- commit
- push
- pull

---

# Basic Commands Demo

- `git clone git@github.com:sunitparekh/java-rdbms-unittest.git`
- `git add .`
- `git commit -m "demo changes"`
- `git pull --rebase`
- `git push`

# Git Branch Commands

- fetch
- branch
- checkout
- merge
- pull
- push

---

# Branch Command Demo

- `git fetch`
- `git branch -a`
- `git checkout -b test`
- `git commit -m "test commit to branch"`
- `git push`

# Why Git?

- Centralised vs Distributed
  - perform all operations without connected to central server
- Git feels blazing fast, as most of the operations are local
- Easy branching & merging
  - Create local branches without affecting others
  - Merge locally any branch and test easily
- Ignore semantics are easier and more flexible
- Pull Request feature in Git



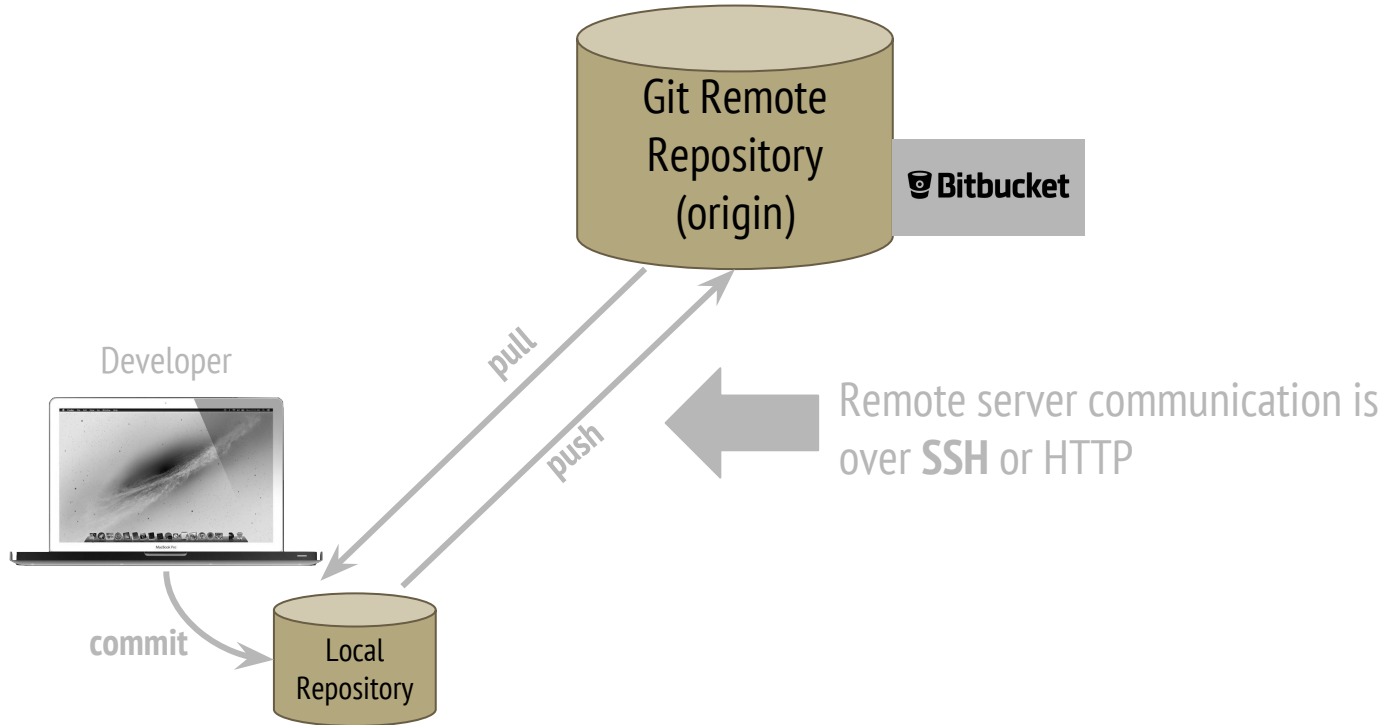
# Don't like Command Line Interfaces !!!

- ungit <https://github.com/FredrikNoren/ungit>
- **\*SourceTree** <https://www.sourcetreeapp.com/> (Most Popular)
- TortoiseGit <https://tortoisegit.org/>
- GitKraken <https://www.gitkraken.com/>
- GitHub Desktop <https://desktop.github.com/>

# Tools Demo

- SourceTree <https://www.youtube.com/playlist?list=PLpL2ONl1hMLtLY1Y7YJNcA5zumvaITLYs>
- ungit <https://www.youtube.com/watch?v=hkBVAi3oKvo&feature=youtu.be>

# Always remember, local and remote



# SVN to Git mapping

|   | Subversion                                   | Git  |
|---|--|--|
| Get copy of code first time from central server | <b>checkout</b>                              | <b>clone</b>   |
| Get updates of code from central server         | <b>update</b>                                | <b>pull</b>  |
| Committing your changes                         | <b>commit</b>                                | <b>commit</b> (local)<br><b>push</b> (remote)                            |
| Status  | <b>stat</b>                                  | <b>status</b>  |
| File operations                                 | <b>add</b><br><b>rm</b><br><b>mv</b>         | <b>add</b><br><b>rm</b><br><b>mv</b>                                     |
| Branching                                       | <b>copy</b><br><b>switch</b><br><b>merge</b> | <b>branch</b> (local)<br><b>checkout</b> (local)<br><b>merge</b> (local) |

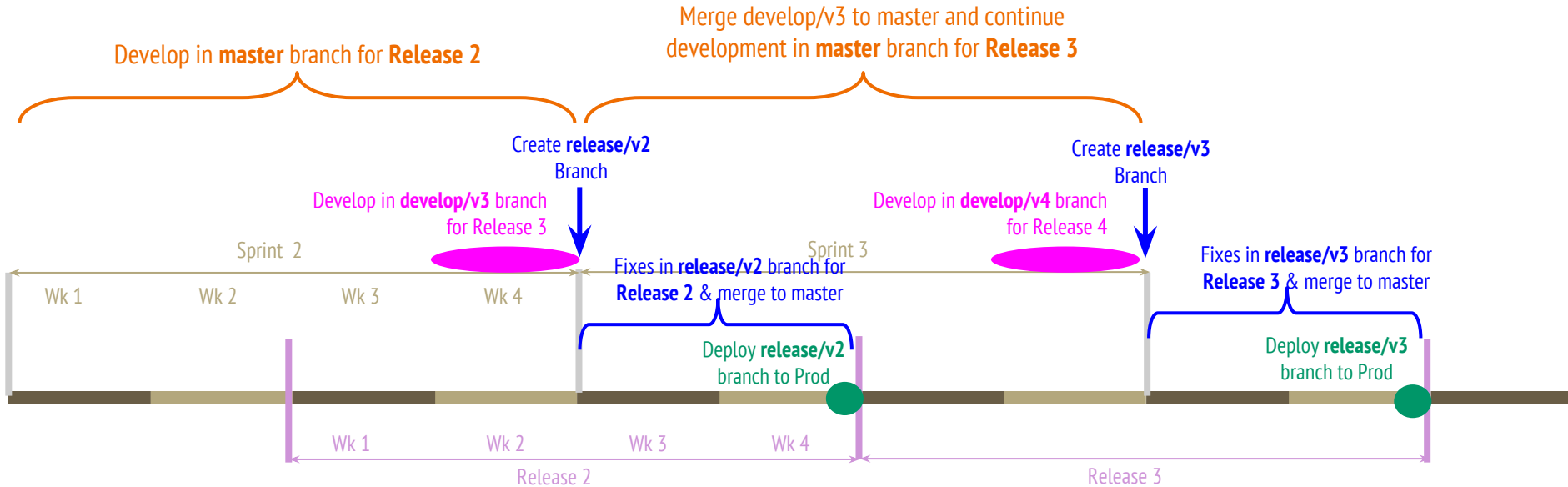
# Challenges with Git

- Higher learning curve than Subversion
  - add, commit, push, pull.... more commands to learn
  - New commands available e.g. stash
-

# Git Workflows

- 3 branch model
- GitFlow

# 3 branch model with Release and Sprint lifecycle



**Release:** Fixed time based cycle for developed software to be deployed to production.

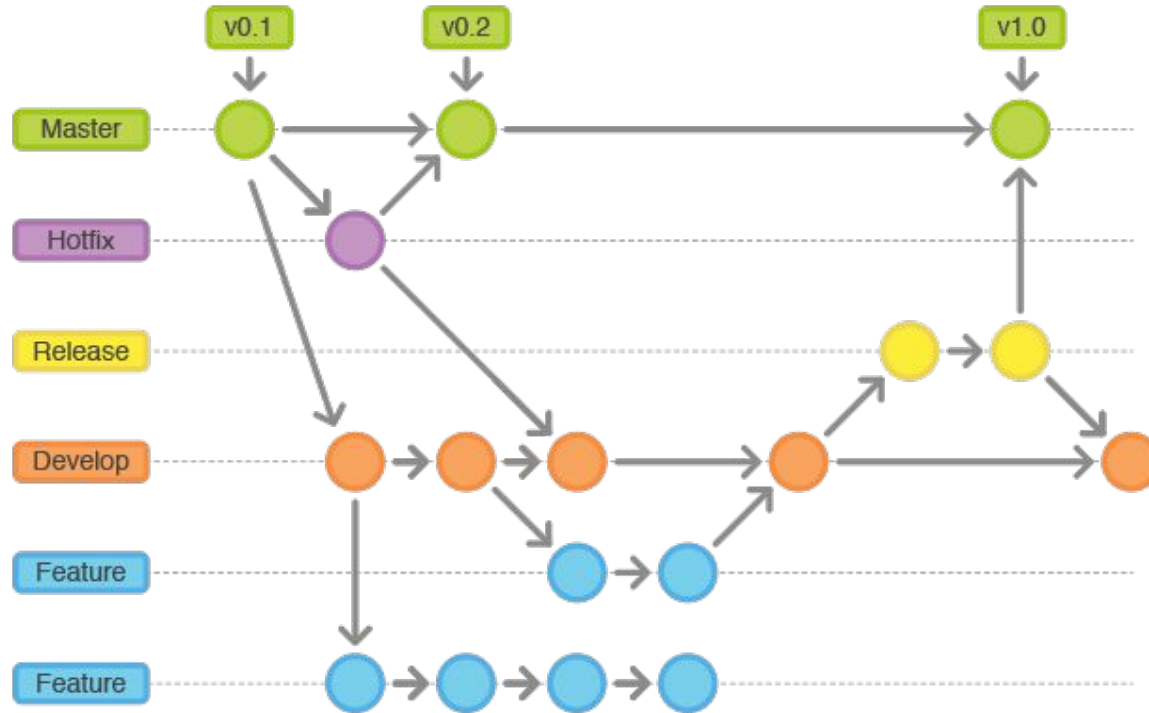
**Sprint:** Development cycle for a release. At end of sprint, development for release should be complete and spillover should be moved to next release.

3 branching model explained in detail:

Video <http://bit.ly/branch1234>

Slides <http://bit.ly/3-branch-model>

# GitFlow





# Learn Git Online

- Try Git <https://try.github.io> (MUST Recommended)
- uDemy course on Git

# References

- Git Workflows <https://www.atlassian.com/git/tutorials/comparing-workflows/>
- Video Tutorials <https://git-scm.com/videos>