



Using GitHub and Gitflow in your project

Chris Hirt
Sept. 19th, 2018



About Me



Email: chris@hirtfamily.net

CS Interim in Thailand 2015 and 2018



Who Uses Git?



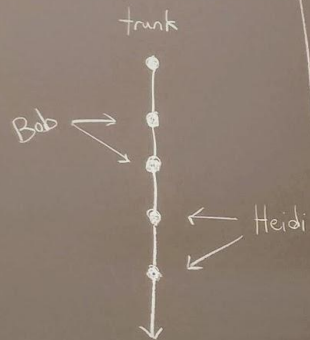
<https://xkcd.com/1597/>



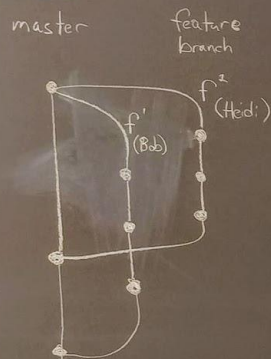
Different Git Workflows

- Centralized Repository Workflow (trunk or master branch only)
- Feature Branch Workflow
- Master and Develop Branch Workflow
- Release Branch Workflow
- Maintenance Branch Workflow (Hotfix Workflow)
- Gitflow Workflow (Uses all of the above)
- Customize and use what works best for your team and project!

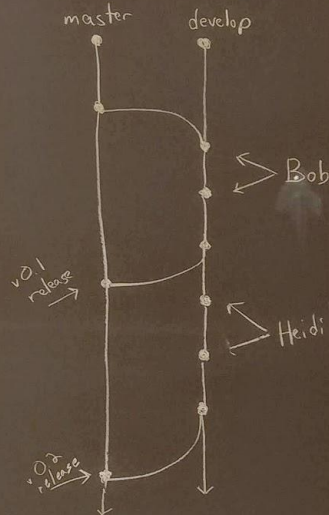
Centralized Workflow



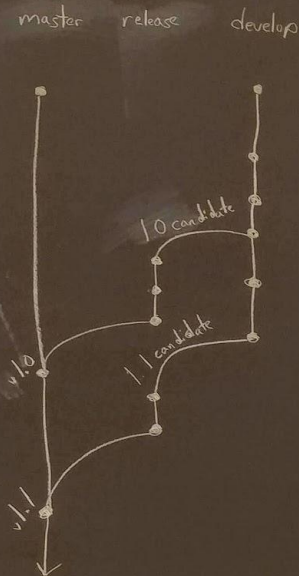
Feature Branch Workflow



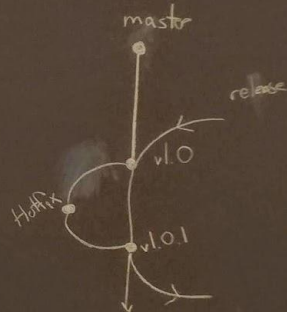
Master and Develop Workflow



Release Branch Workflow

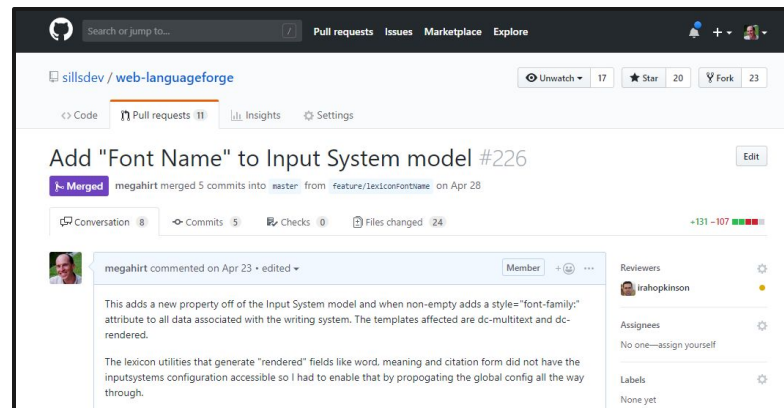


Maintenance Branch Workflow



Pull Request Workflows

- Why use Pull Requests?
 - Code Review
 - Get other's feedback while work is under development
 - Summarize and document changes
- Pull Request Workflow
 - Fork, Branch and PR Workflow - the default for open-source software and community projects
 - Same Repo, Branch and PR Workflow



<https://github.com/sillsdev/web-languageforge/pull/226>

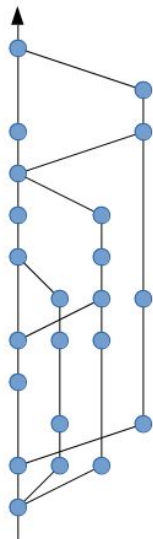


Which Workflow Should You Use?

- Feature Branch Workflow (master + feature branches only)
- Same repo Pull Request Workflow
- Don't commit to master directly!
- Who approves Pull Requests?
 - Team Lead
 - Other members
 - Yourself ?

Linear vs Non-Linear History (Rebase vs Merge)

Non-linear history

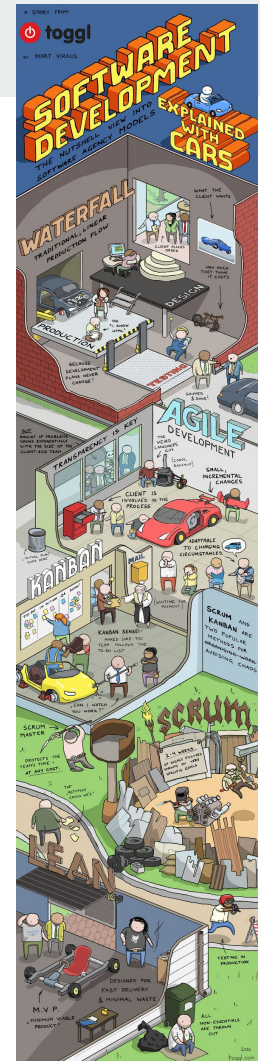
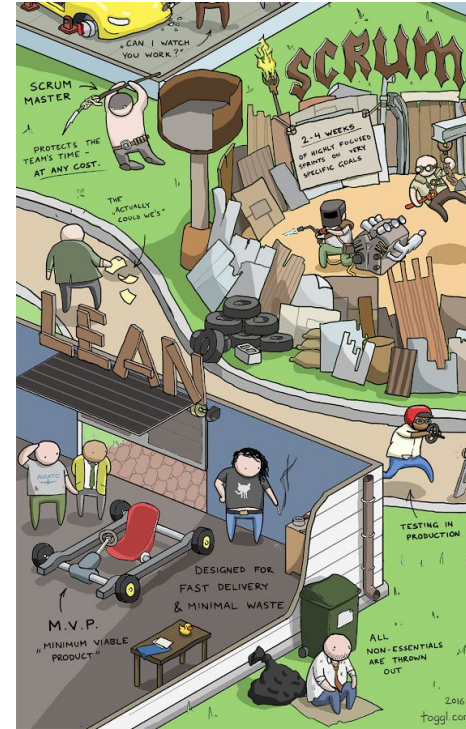
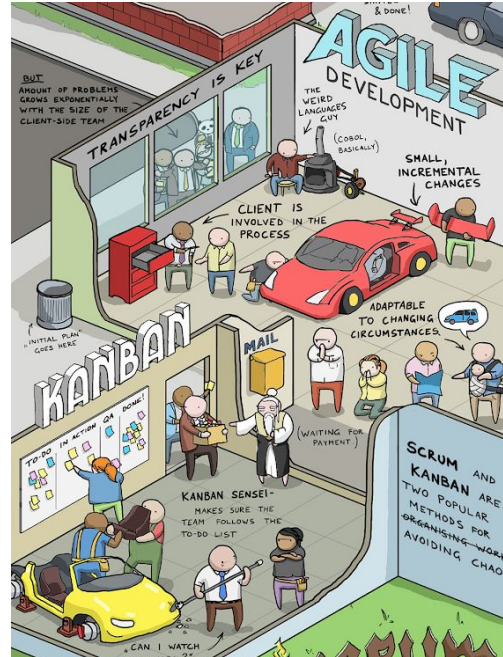
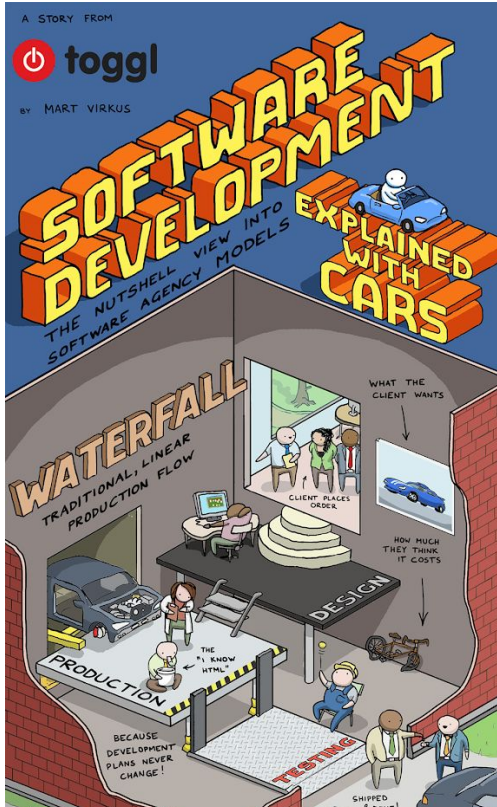


Linear history



- **Rebase Strategy:**
 - Leads to a cleaner repo / linear history
 - Rewrites history (can be dangerous / destructive)
 - Requires you to deal with conflicts earlier (potentially)
 - Should only be used on local branches
 - Harder to do
 - Better for short-lived branches
- **Merge Strategy:**
 - Is non-destructive to history
 - Creates separate merge commits
 - Requires merges from master along the way
 - Better for long-lived branches

A funny cartoon poking fun at the different software development process models



<https://toggl.com/developer-methods-infographic/?ref=9ag>