What testers need to know about Git.

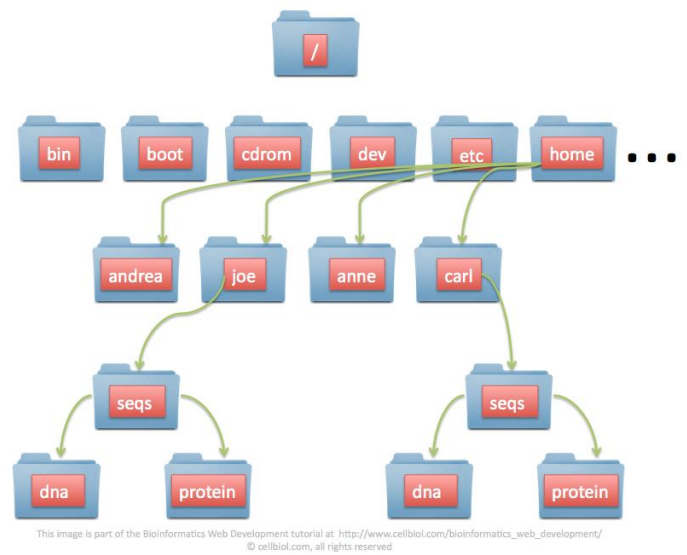
SAST Stockholm Q2 2019 – Patrik Lindström DevOps Engineer

Presentation and video of presentation will be avaible.

Code Repositories

- Store code
- Handle versions of code
- Support workflow for developers

A Filesystem with timemachine abilities and parallell universes.

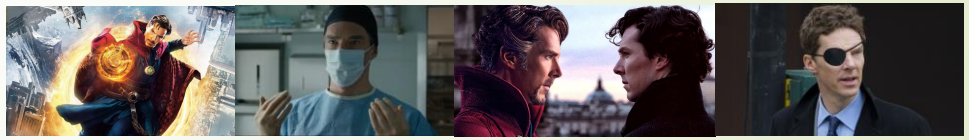


A Filesystem with timemachine abilities and parallell universes. With cloning cababilites

upstream

origin

local



What is git?

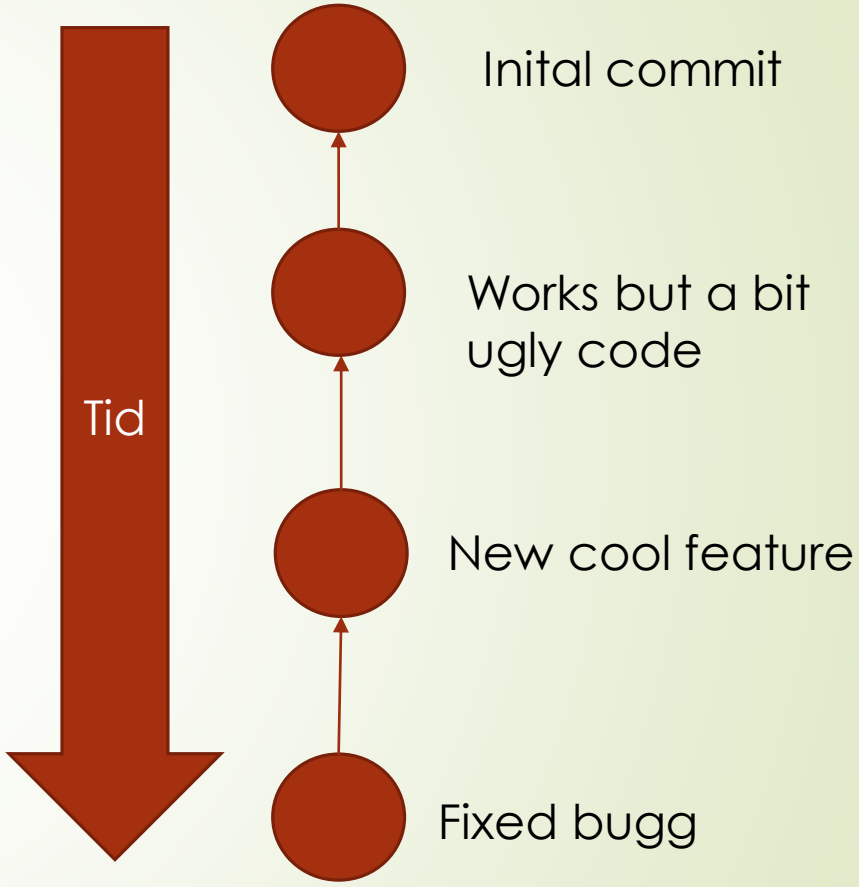
- Its a tool to handle versions of document and share them with others and see differences between them.
- Created by Linus Torwald to handle the code for the Linux kernel.
- Git is decentralized. All copies of a repositories branch contains all versions of it. Everywhere – on every developers machine.
- Git is decentralized. A local copy exist AND a cached copy of referenced cloned repository. Communications with other referenced repos are through: fetch, pull and push.
- Easy to create new branches and to merge them with other branches
- A snapshot av all files are saved in a compressed format known as a commit.
- It is not easy to learn. Weird syntax.
- Very popular. I have never heard anyone go back to eg: subversion, TFVC, CVS or ClearCase after using git.

Hands on demo

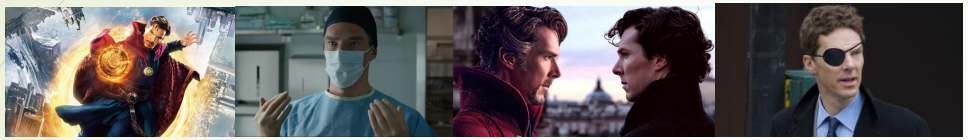
- A long and stormy night
- Trying cmdline and graphical tool
- See how different tools interact
- Be amazed how files come and dissapear.

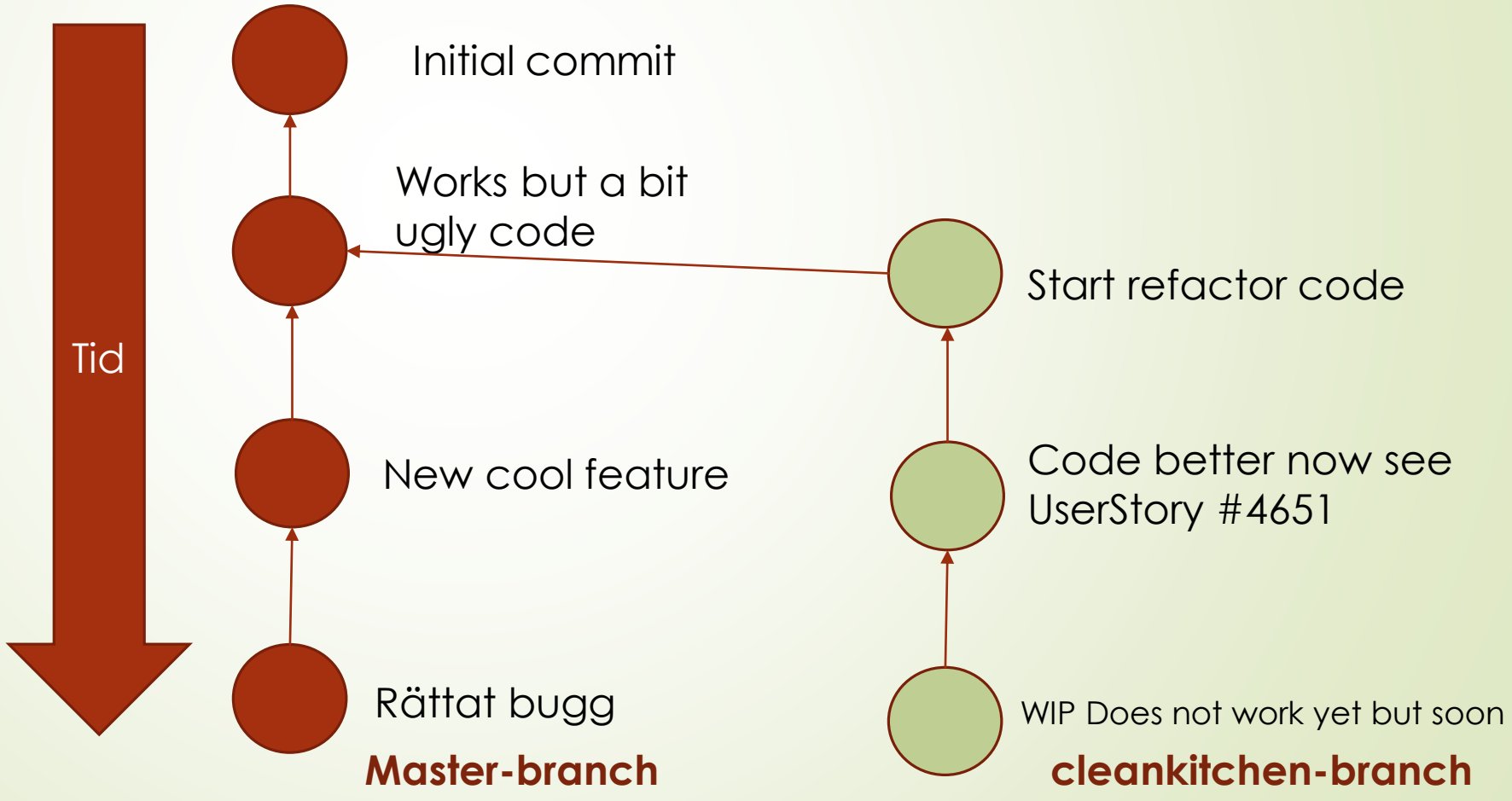
Time machine

- Move between commits.



Parallell universes- branches





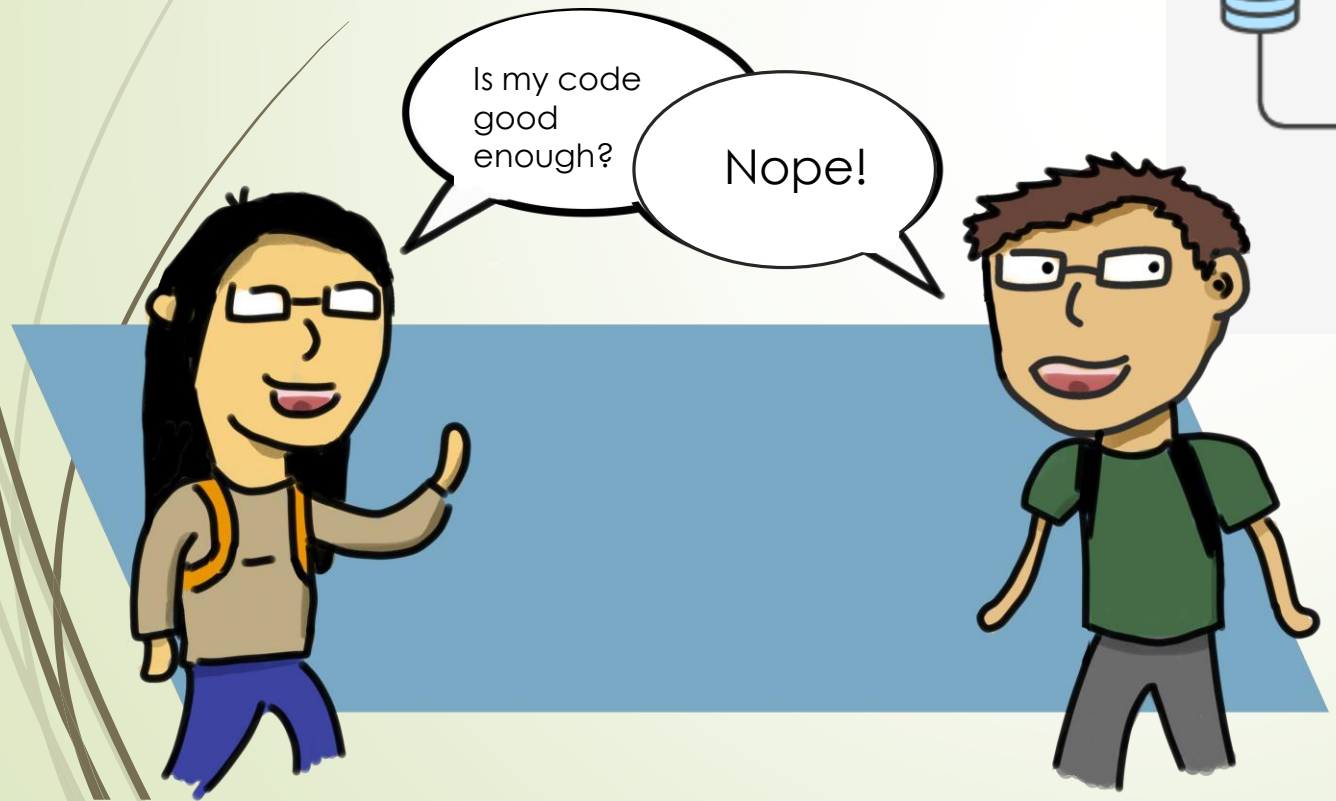
Links to clones

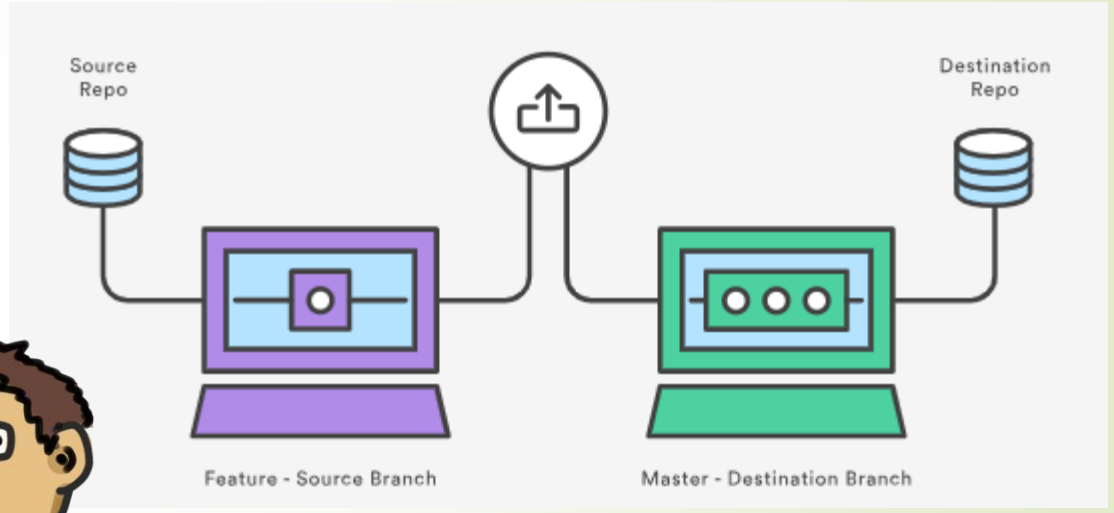
origin

Push/Pull

local

Pull request – social interaction



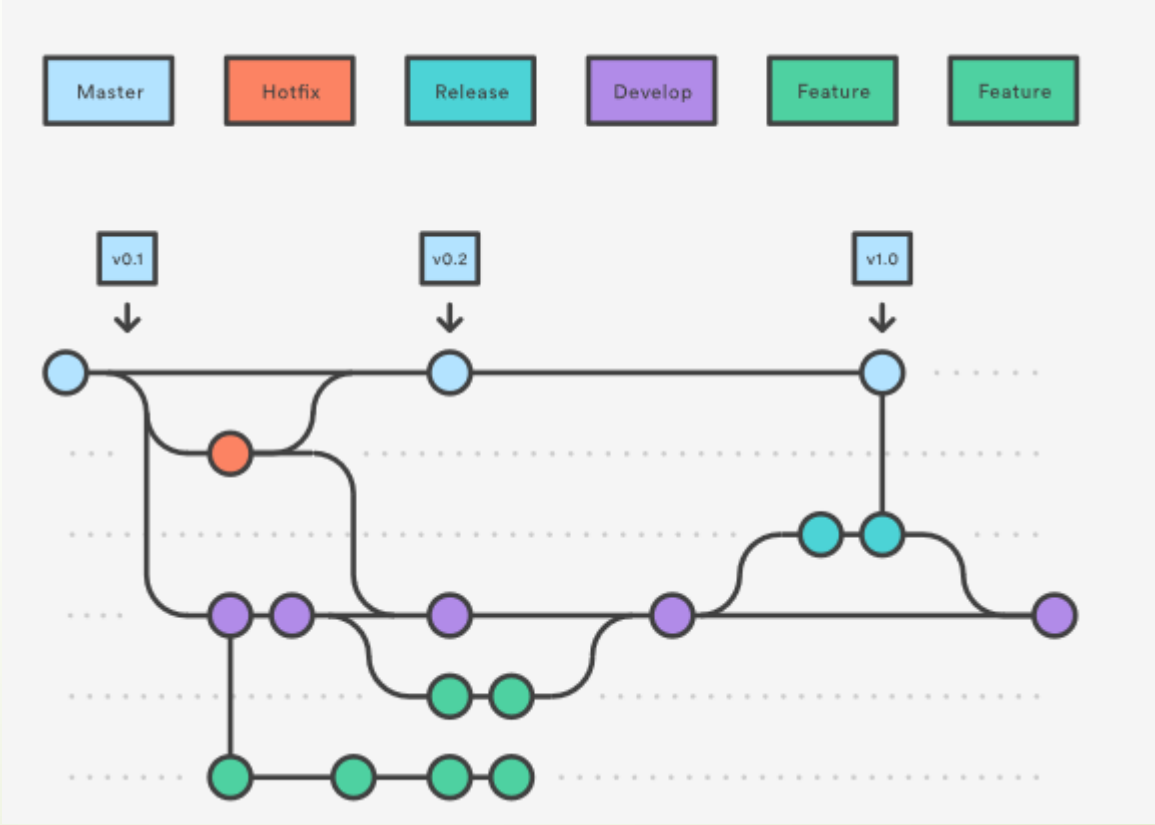


Branchstrategies how to cooperate with others

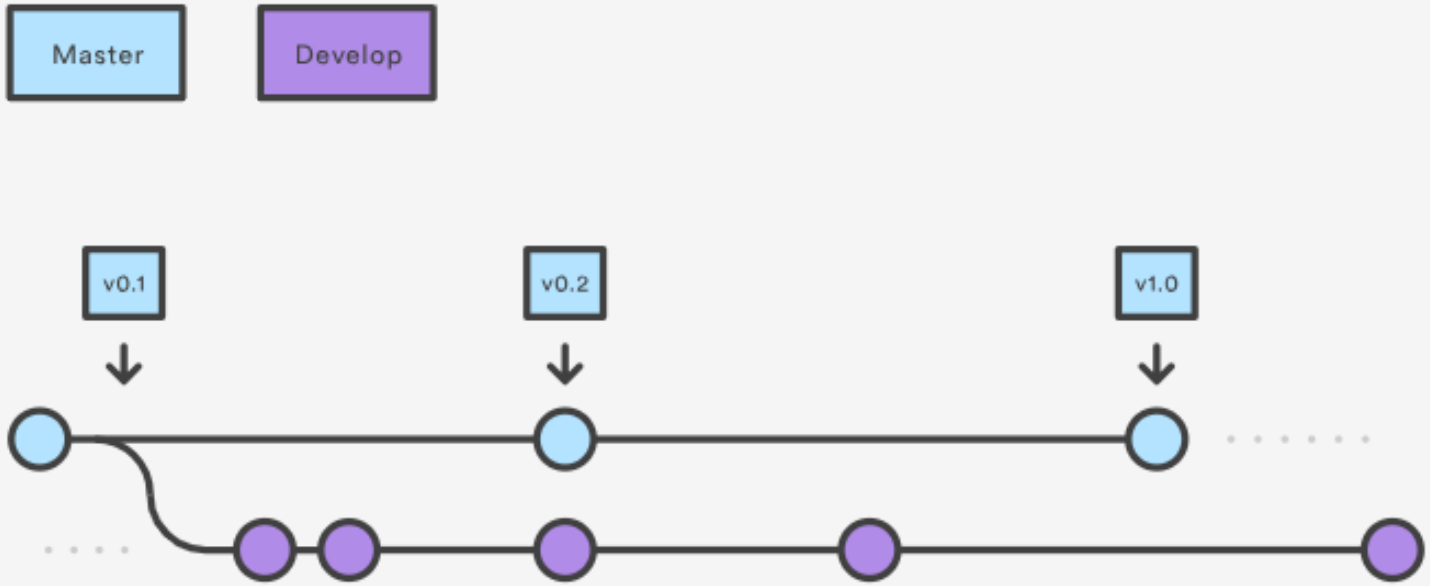
- Trunk based development
- Githubflow
- **Gitflow**
- Others

Gitflow

- https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow



How it works

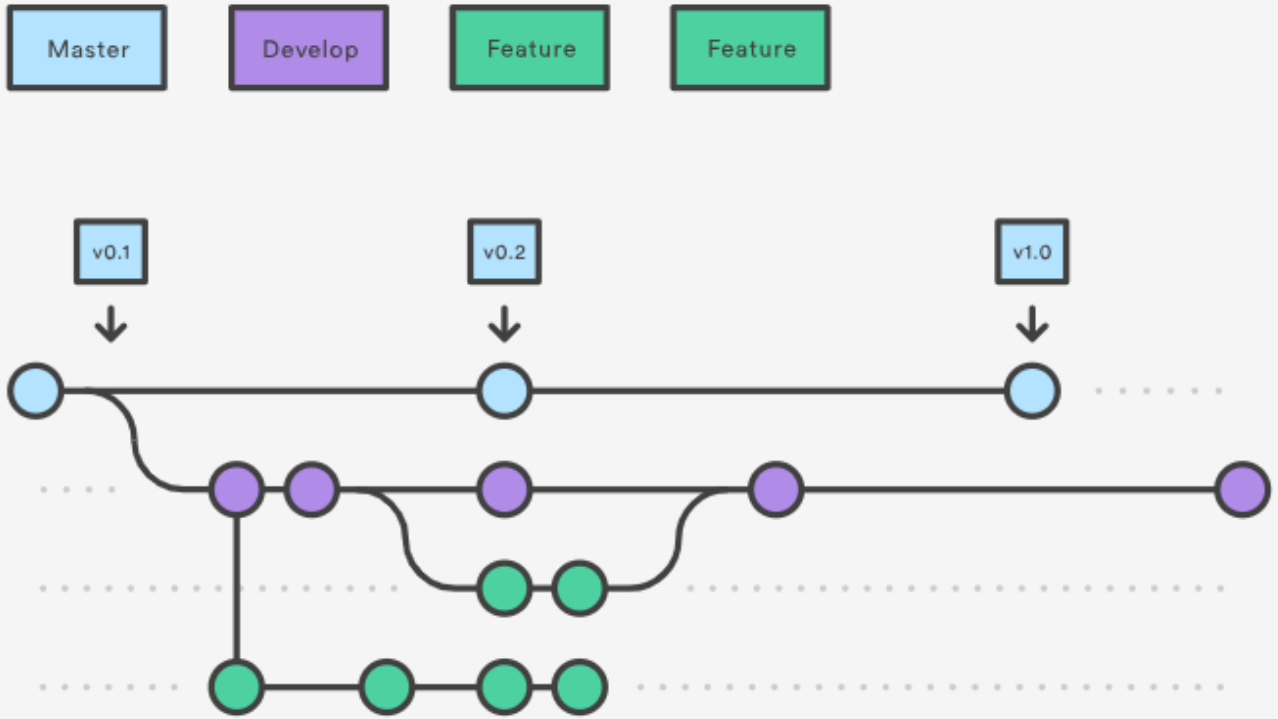


Develop and Master Branches

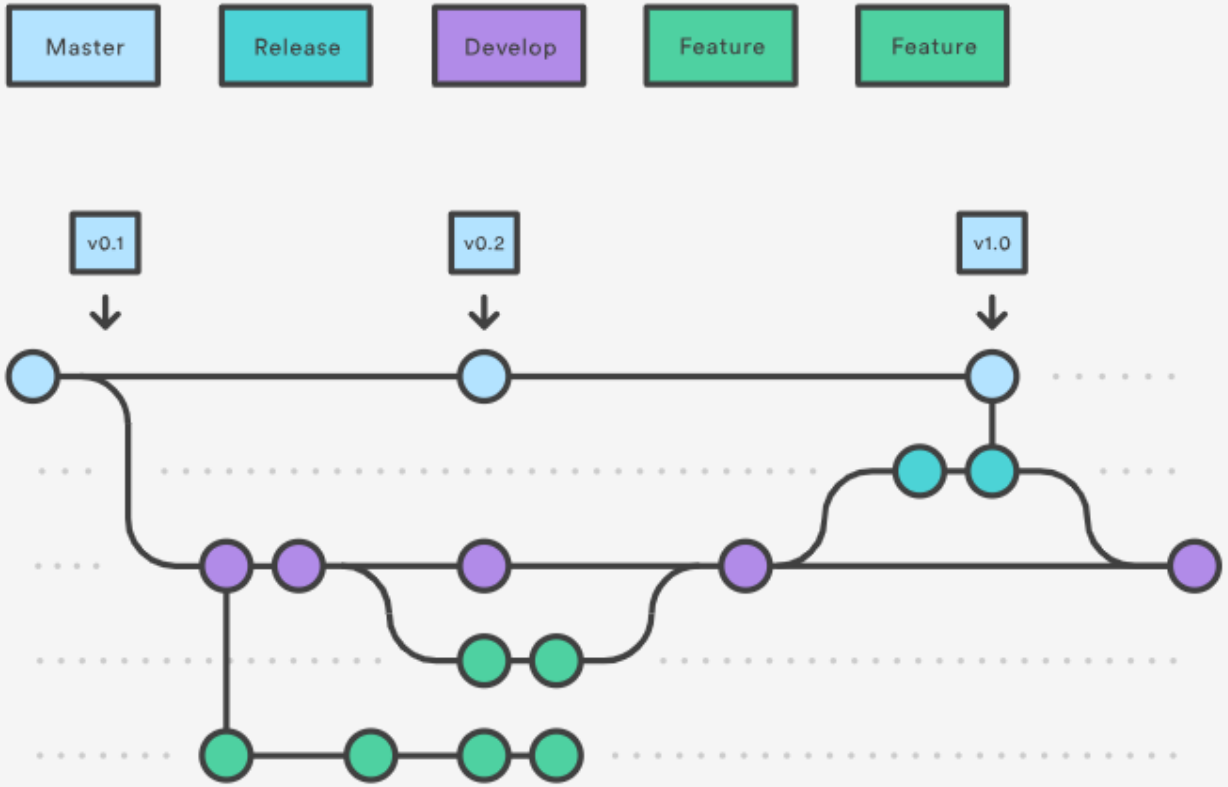
Instead of a single `master` branch, this workflow uses two branches to record the history of the project. The `master` branch stores the official release history, and the `develop` branch serves as an integration branch for features. It's also convenient to tag all commits in the `master` branch with a version number.

Feature Branches

Each new feature should reside in its own branch, which can be pushed to the central repository for backup/collaboration. But, instead of branching off of `master`, `feature` branches use `develop` as their parent branch. When a feature is complete, it gets merged back into develop. Features should never interact directly with `master`.

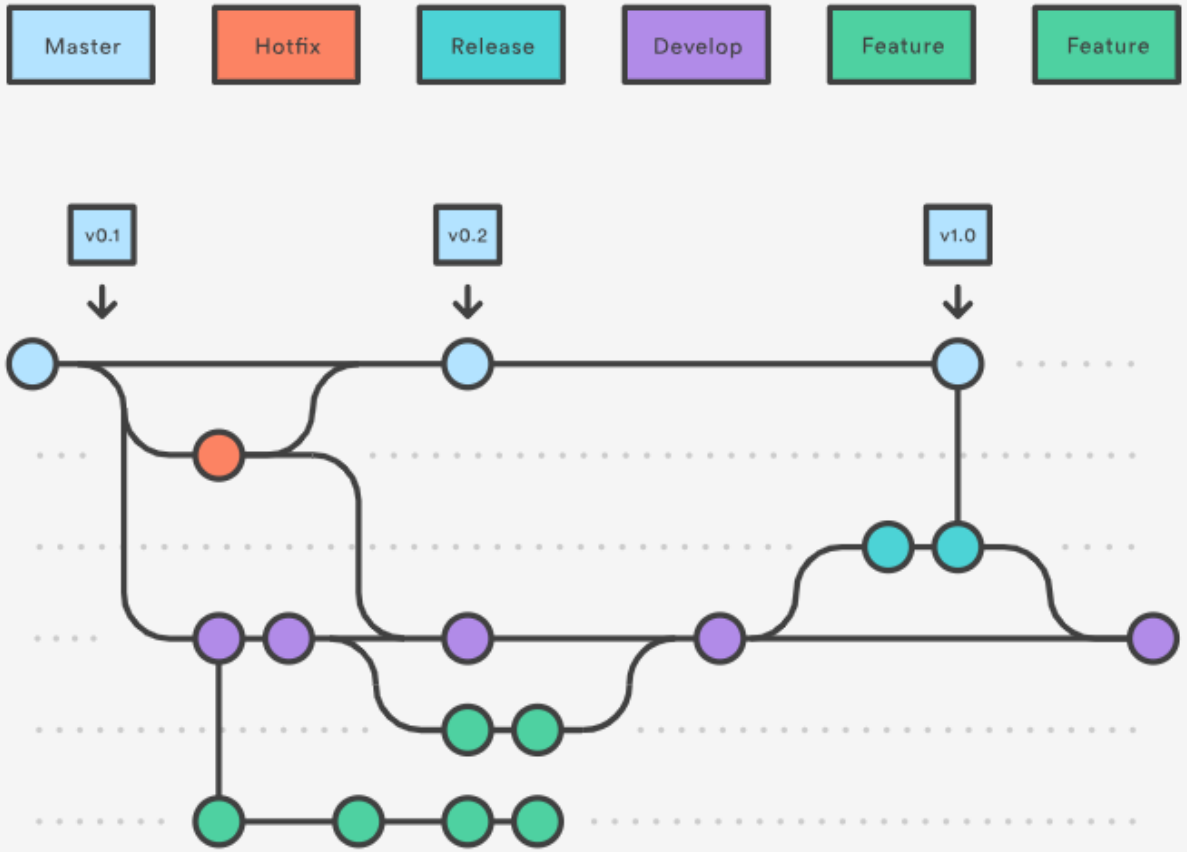


Release Branches

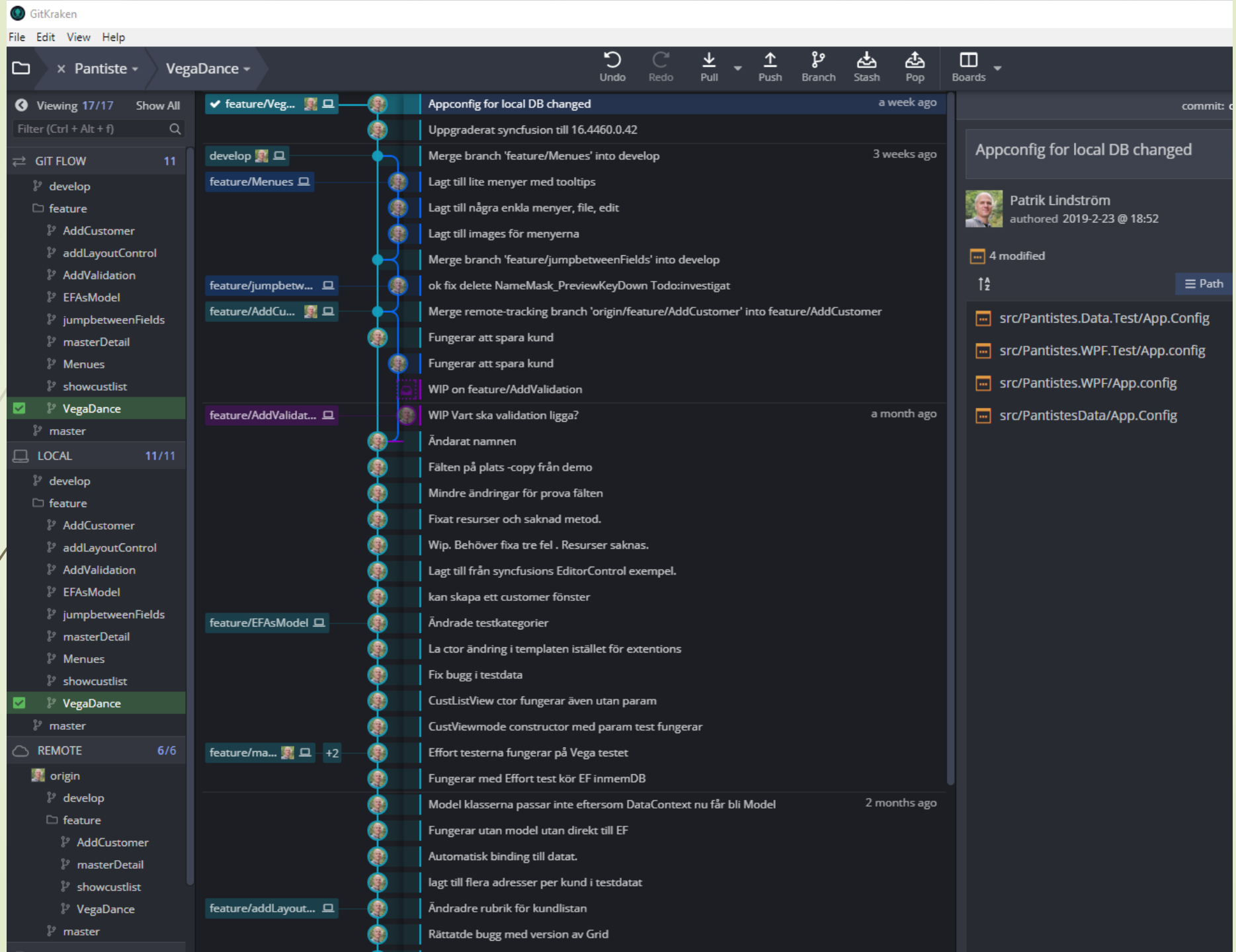


Once `develop` has acquired enough features for a release (or a predetermined release date is approaching), you fork a `release` branch off of `develop`. Creating this branch starts the next release cycle, so no new features can be added after this point—only bug fixes, documentation generation, and other release-oriented tasks should go in this branch. Once it's ready to ship, the `release` branch gets merged into `master` and tagged with a version

Hotfix Branches



Maintenance or `"hotfix"` branches are used to quickly patch production releases. `Hotfix` branches are a lot like `release` branches and `feature` branches except they're based on `master` instead of `develop`. This is the only branch that should fork directly off of `master`. As soon as the fix is complete, it should be merged into both `master` and `develop` (or the current `release`


GitKraken
File Edit View Help
Pantiste
VegaDance
Undo
Redo
Pull
Push
Branch
Stash
Pop
Boards
Viewing 17/17
Show All
Filter (Ctrl + Alt + f)
GIT FLOW 11
develop
feature
AddCustomer
addLayoutControl
AddValidation
EFAsModel
jumpbetweenFields
masterDetail
Menues
showcustlist
VegaDance
master
LOCAL 11/11
develop
feature
AddCustomer
addLayoutControl
AddValidation
EFAsModel
jumpbetweenFields
masterDetail
Menues
showcustlist
VegaDance
master
REMOTE 6/6
origin
develop
feature
AddCustomer
masterDetail
showcustlist
VegaDance
master
feature/Veg...
Appconfig for local DB changed
a week ago
Uppgraderat syncfusion till 16.4460.0.42
develop
Merge branch 'feature/Menues' into develop
3 weeks ago
feature/Menues
Lagt till lite menyer med tooltips
Lagt till några enkla menyer, file, edit
Lagt till images för menyerna
Merge branch 'feature/jumpbetweenFields' into develop
feature/jumpbetw...
ok fix delete NameMask_PreviewKeyDown Todo:investigat
feature/AddCu...
Merge remote-tracking branch 'origin/feature/AddCustomer' into feature/AddCustomer
Fungerar att spara kund
Fungerar att spara kund
WIP on feature/AddValidation
feature/AddValidat...
WIP Vart ska validation ligga?
a month ago
Ändarat namnen
Fälten på plats -copy från demo
Mindre ändringar för prova fälten
Fixat resurser och saknad metod.
Wip. Behöver fixa tre fel . Resurser saknas.
Lagt till från syncfusions EditorControl exempel.
kan skapa ett customer fönster
feature/EFAsModel
Ändrade testkategorier
La ctor ändring i templaten istället för extentions
Fix bugg i testdata
CustListView ctor fungerar även utan param
CustViewmode constructor med param test fungerar
feature/ma... +2
Effort testerna fungerar på Vega testet
Fungerar med Effort test kör EF inmemDB
Model klasserna passar inte eftersom DataContext nu får bli Model
2 months ago
Fungerar utan model utan direkt till EF
Automatisk binding till datat.
lagt till flera adresser per kund i testdatat
feature/addLayout...
Ändradre rubrik för kundlistan
Rättatde bugg med version av Grid
commit:
Appconfig for local DB changed
Patrik Lindström
authored 2019-2-23 @ 18:52
4 modified
Path
src/Pantistes.Data.Test/App.Config
src/Pantistes.WPF.Test/App.config
src/Pantistes.WPF/App.config
src/PantistesData/App.Config

Many commands

- Demo of git bisect

- Finds what commit that breaks the system and introduce the bug.
- Scenario you come up with new great test. It works on first commit but not last. You want to find out what code change breaked the test: git bisect
- I will go through the git bisect cmd in a video that will be avaible for you.

Other uses than coding

- Music composing: https://github.com/CMAA/nova-organi-harmonia
- Food recepie: http://forkthecookbook.com/
- Wiki pages: Both TFS (Azure DevOps), Github has this.
- HR documents see : https://github.com/OctopusDeploy/People
- Legal documents: https://github.com/seriesseed/equity https://blog.abevoelker.com/gitlaw-github-for-laws-and-legal-documents-a-tourniquet-for-american-liberty/
- Software requirements
- Test plans
- Test scripts

Ad finem

- Understand how git work
- Learn some git commands – git init, git commit –m"Hello"
- Suggest to store shared documents in markdown in git instead of word in sharepoint. (If you are dealing with hard core developer and wants them to deal with shared documents.)
- Extra points for knowing about markdown format.