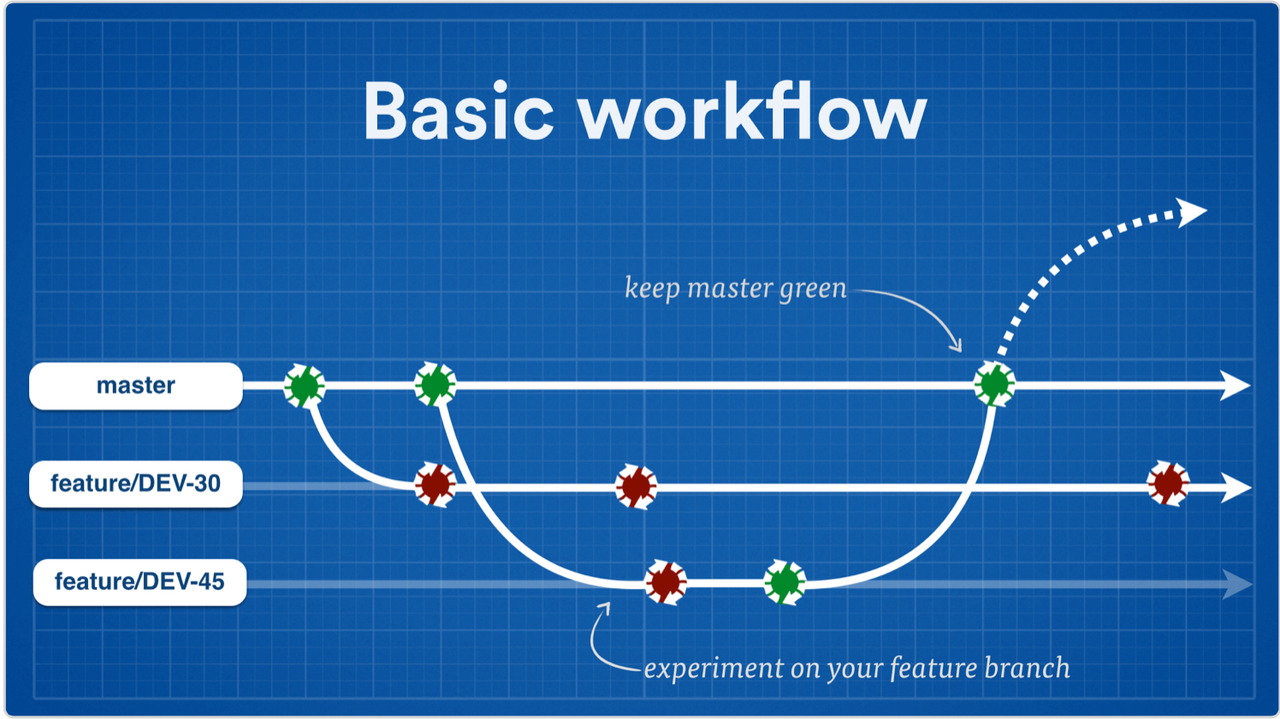
**BRANCHING PATTERNS**

**1)Feature Branch Model**

****

**Features:**

* Pull requests make it incredibly easy for your team to comment on each other’s work.
* If one gets stopped in the middle of a feature, you can open a pull request asking for suggestions from your colleagues.
* Since all the developments is done at the feature branch,master is kept clean and it has clear,unbroken codes.

**Disadvantage:**

* This development practice where every feature is developed in its own remote branch drives the process towards big integration issues .

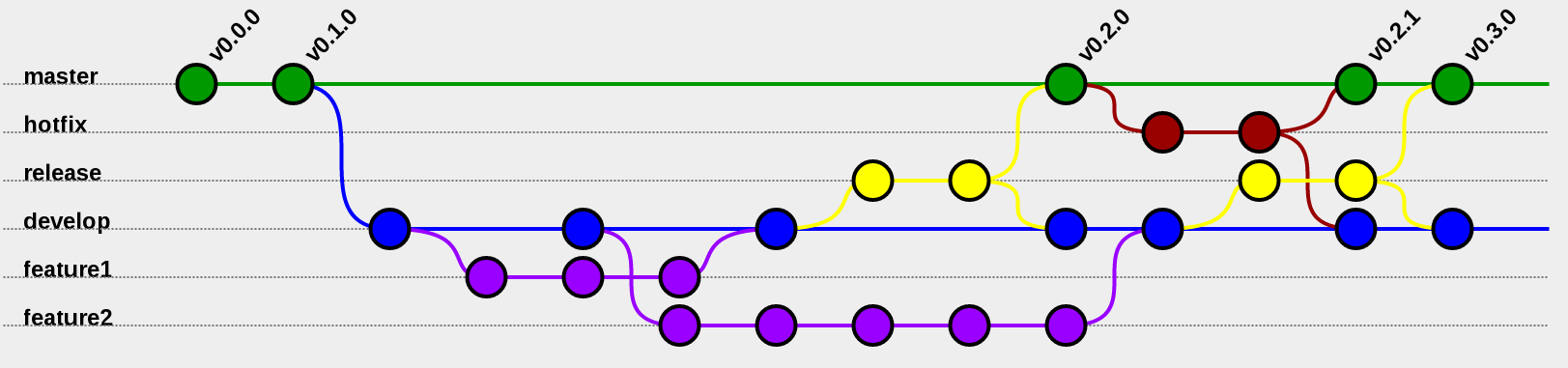
**2)Github Flow:**



**Features:**

* Followed when there is no complex features for the developer.
* Once developer finish a feature or finish a bugfix it is immediately promoted to production version.
* Better suited to continuous Integeration where changes can be quickly made and easily deployed, sometimes multiple times a day.
* When merged to master deployment starts automatically.
* If your code is having only one version in production at all times , you may use github-flow.

**3)Git Flow**

****

**Features:**

* For larger teams, it’s often beneficial to assign more specific roles to different branches.
* The Gitflow Workflow is a common pattern for managing feature development, release preparation, and maintenance.
* Gitflow is used when one needs to continuously support previous versions in production while developing the next version.

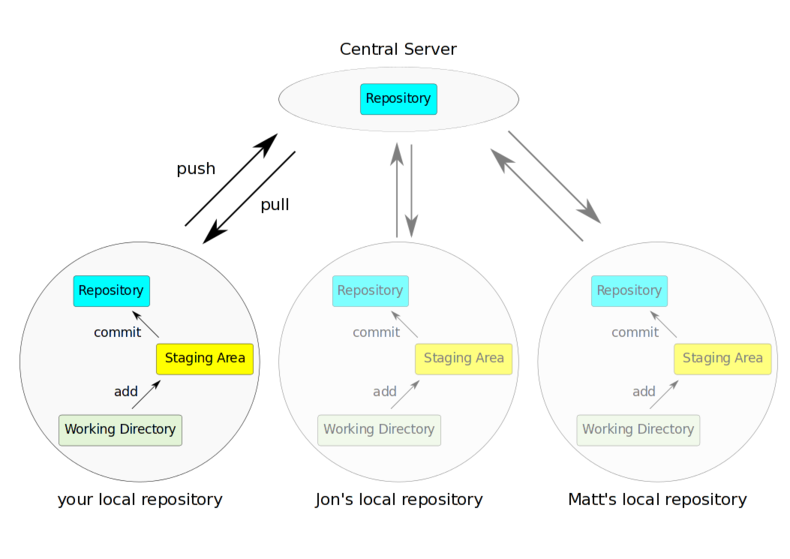
**Disadvantage:**

**Complexity**

* The primary reason for moving away is that the git-flow process is hard to deal with in a continuous (or near-continuous) deployment model.

**4) Centralised Workflow**

* It gives every developer their own local copy of the entire project.
* When the central server goes down or slow it does not affect the files during work since the codes are referenced using local repository.



* When needed, we prefer to undo an entire feature, by reverting at most single merge commit.
* Committing new changes can be done locally without anyone else seeing them. Once you have a group of changes ready, you can push all of them at once.

**Disadvantage:**

* If project contains many large, binary files that cannot be easily compressed, the space needed to store all versions of these files can accumulate quickly.
* If project has a very long history (50,000 changesets or more), downloading the entire history can take an impractical amount of time and disk space.