Central Repository Workflow

Central Repository

Local Pushed Changes
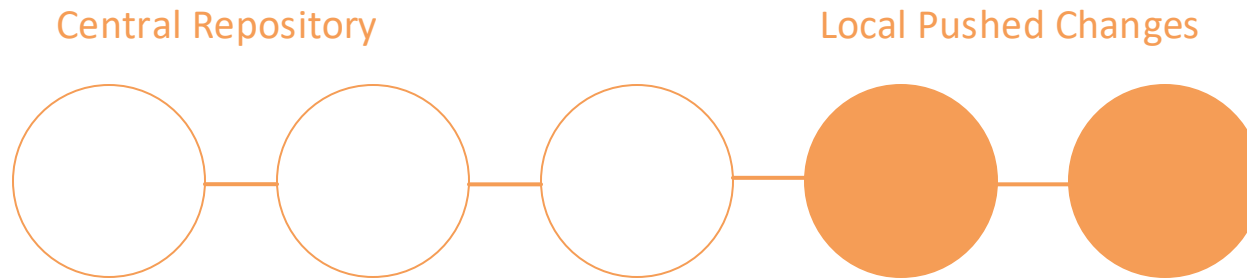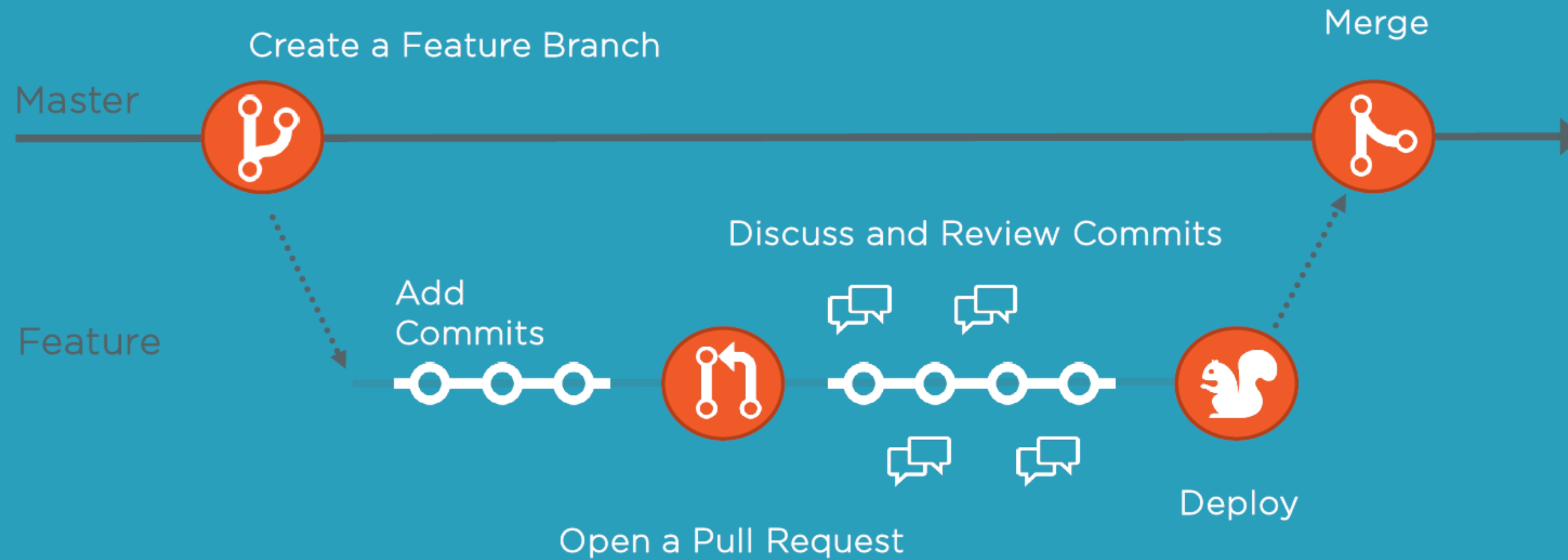
In a centralized model, user pulls all the changes from a central repository and saves them locally Pushes local changes to central repository.

GitHub Flow

Master — Create a Feature Branch — Merge

Feature — Add Commits — Open a Pull Request — Discuss and Review Commits — Deploy

Gitflow

So here we're showing the different branches as lines with time flowing left to right.
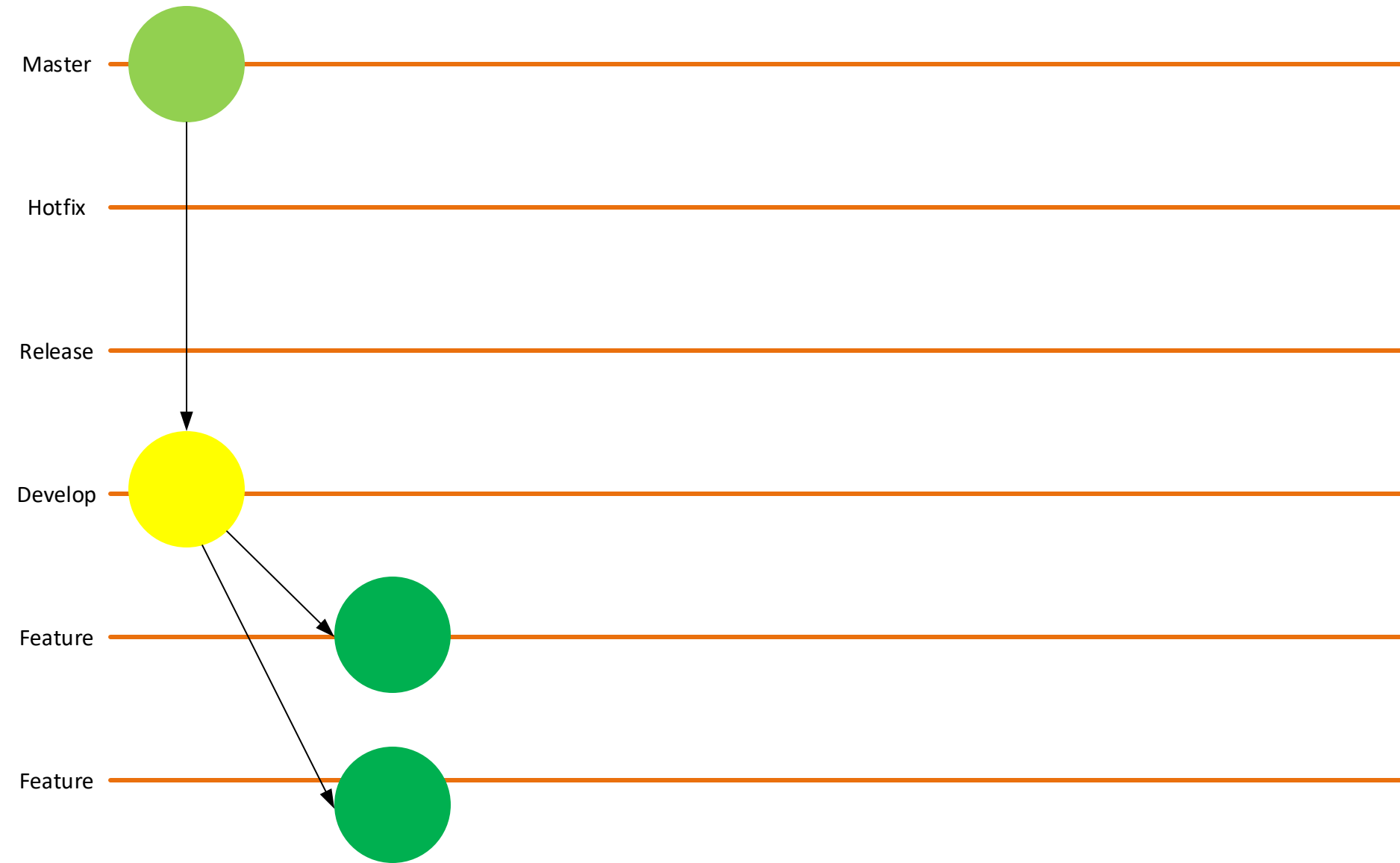At the top here we have master and at the end we have two feature branches with the various other branches we use in-between.
There will be a commit on the master branch
We'll create the develop branch

Master

Hotfix

Release

Develop

Feature

Feature

Gitflow
Develop to Feature

Master

Hotfix
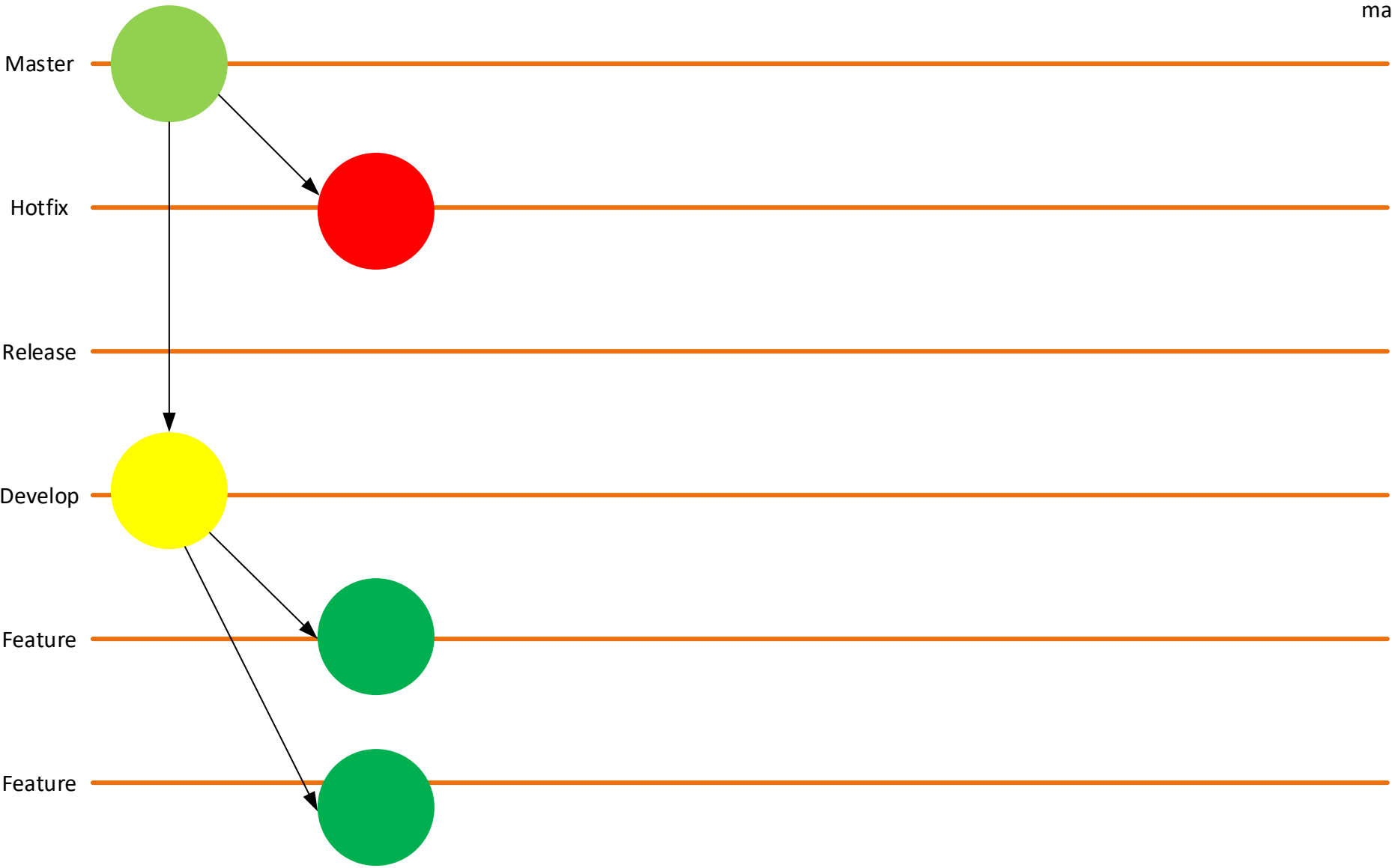
Release

Develop

Feature

Feature

From the develop branch, we'll branch off and
create any feature branches where all the work
goes,
but there will be occasions when you want to
make minor changes, and those minor changes
can just go on the develop branch.

Gitflow
Hotfix

So, work is continuing on these branches. Master's been released at some point and then, somebody realizes that there's a bug.

from master, we can create a hotfix branch and we can do the fix for master on that branch. That fix gets checked back into master.

Master

Hotfix

Release

Develop

Feature

Feature

Gitflow
Hotfix To Master

So, work is continuing on these branches. Master's been released at some point and then, somebody realizes that there's a bug.
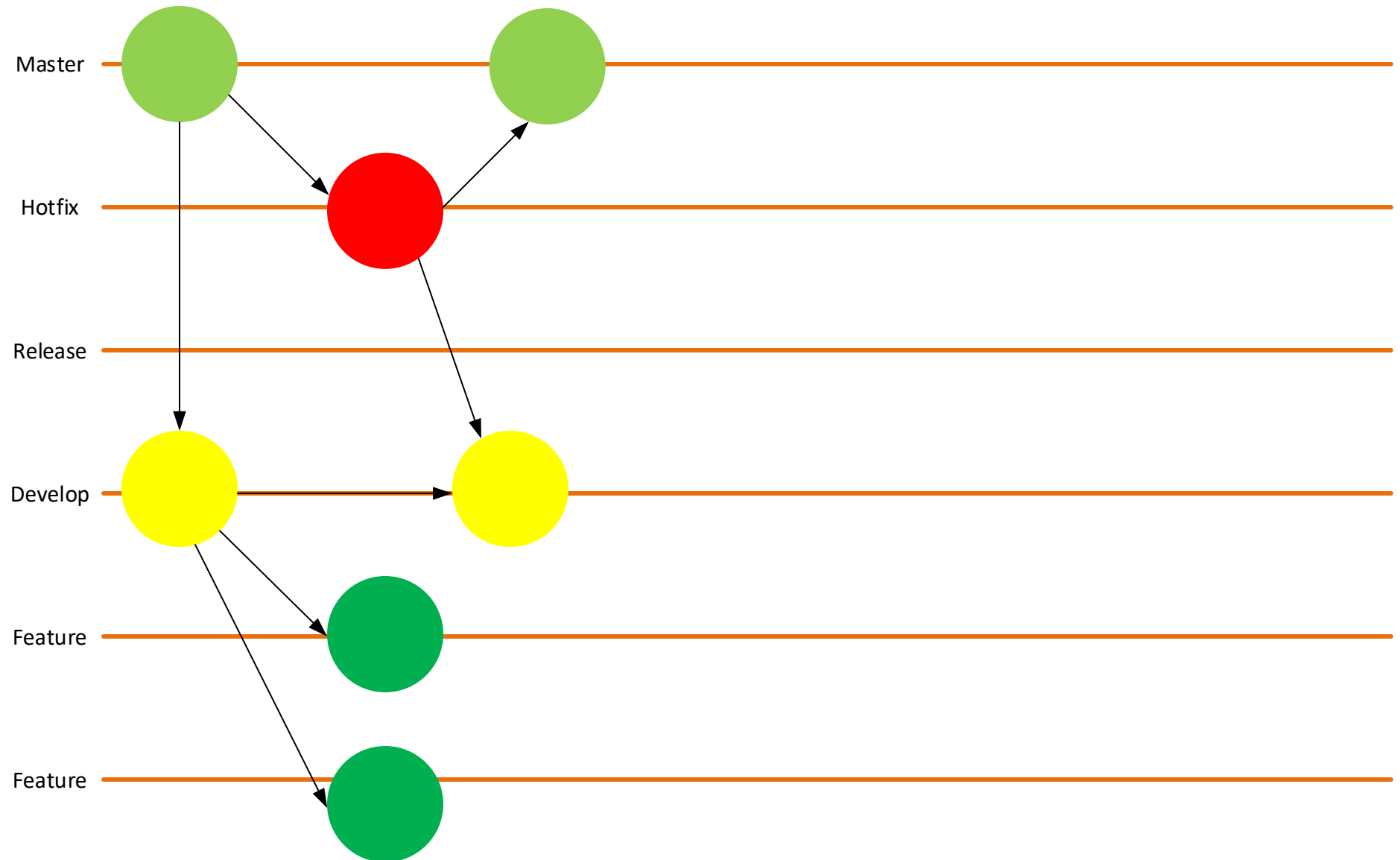
from master, we can create a hotfix branch and we can do the fix for master on that branch. That fix gets checked back into master.

Master

Hotfix

Release

Develo
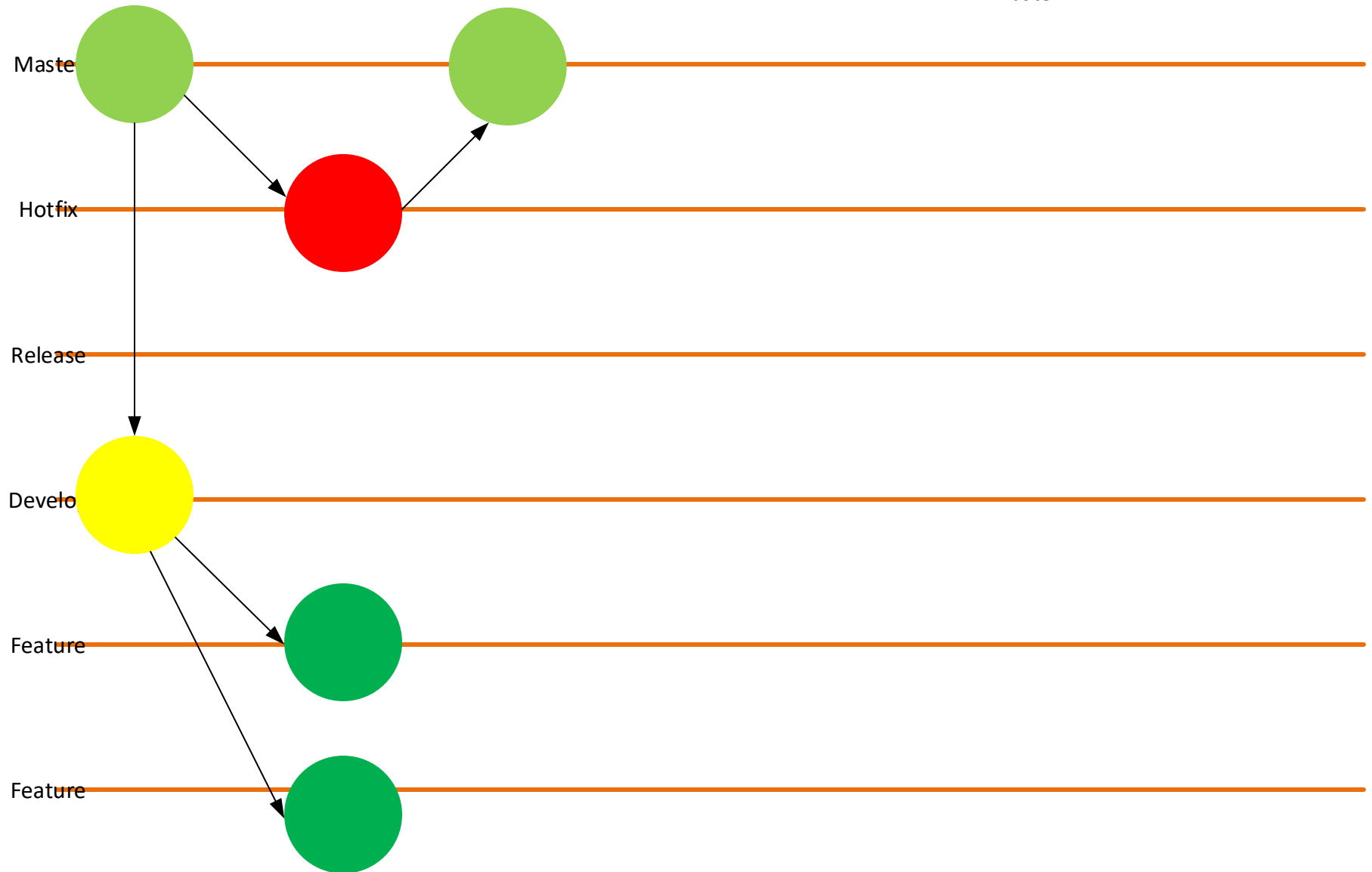
Feature

Feature

Gitflow
Hotfix To Develop

We also need to make sure that fix is also checked into develop.

Master

Hotfix

Release

Develop

Feature

Feature

Gitflow
Hotfix To Develop

We continue work on the feature branch and that feature branch will eventually be checked back in to develop.

Master

Hotfix

Release

Develop
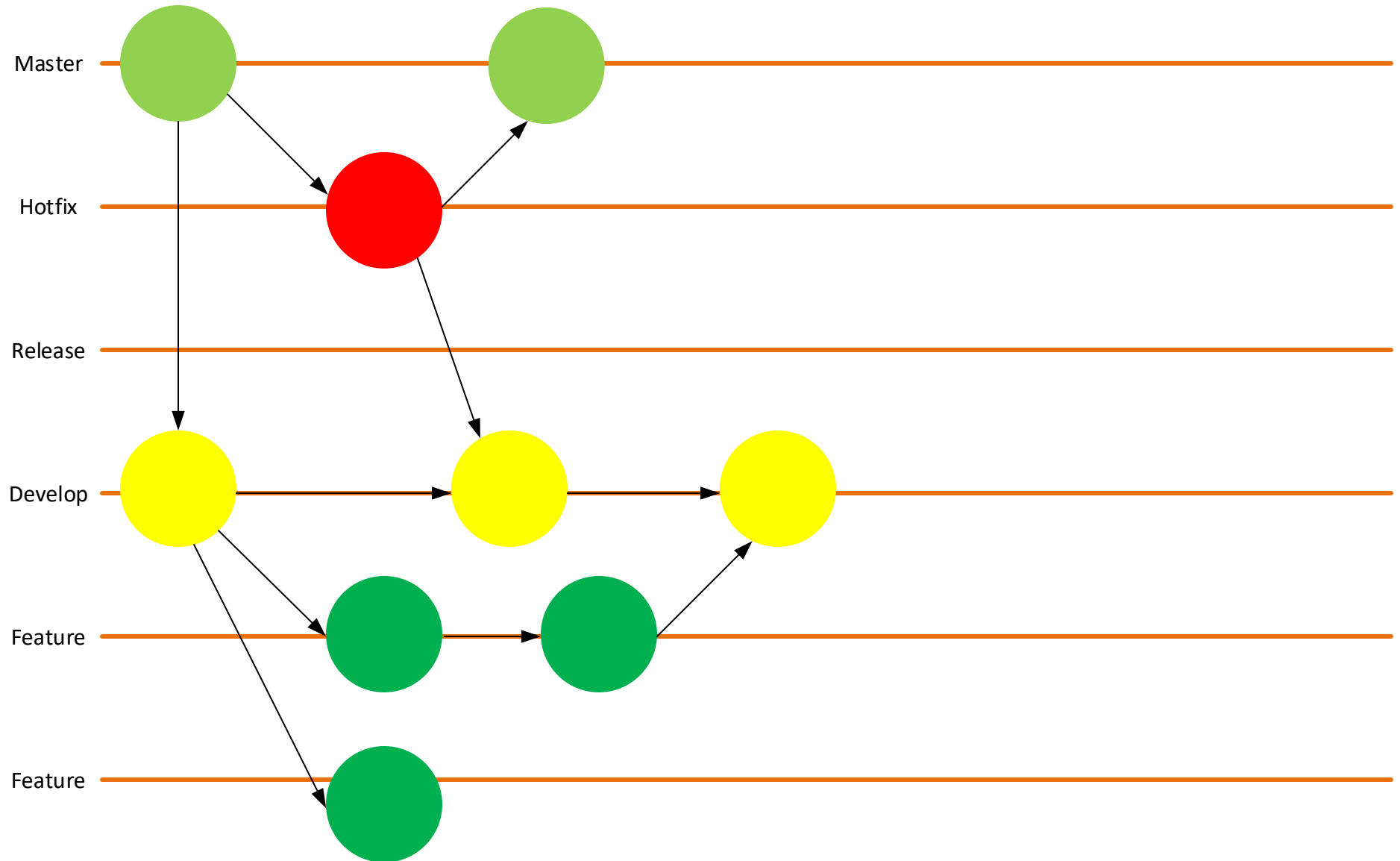
Feature

Feature

Gitflow
Hotfix To Develop

We continue work on the feature branch and that feature branch will eventually be checked back in to develop.

Master
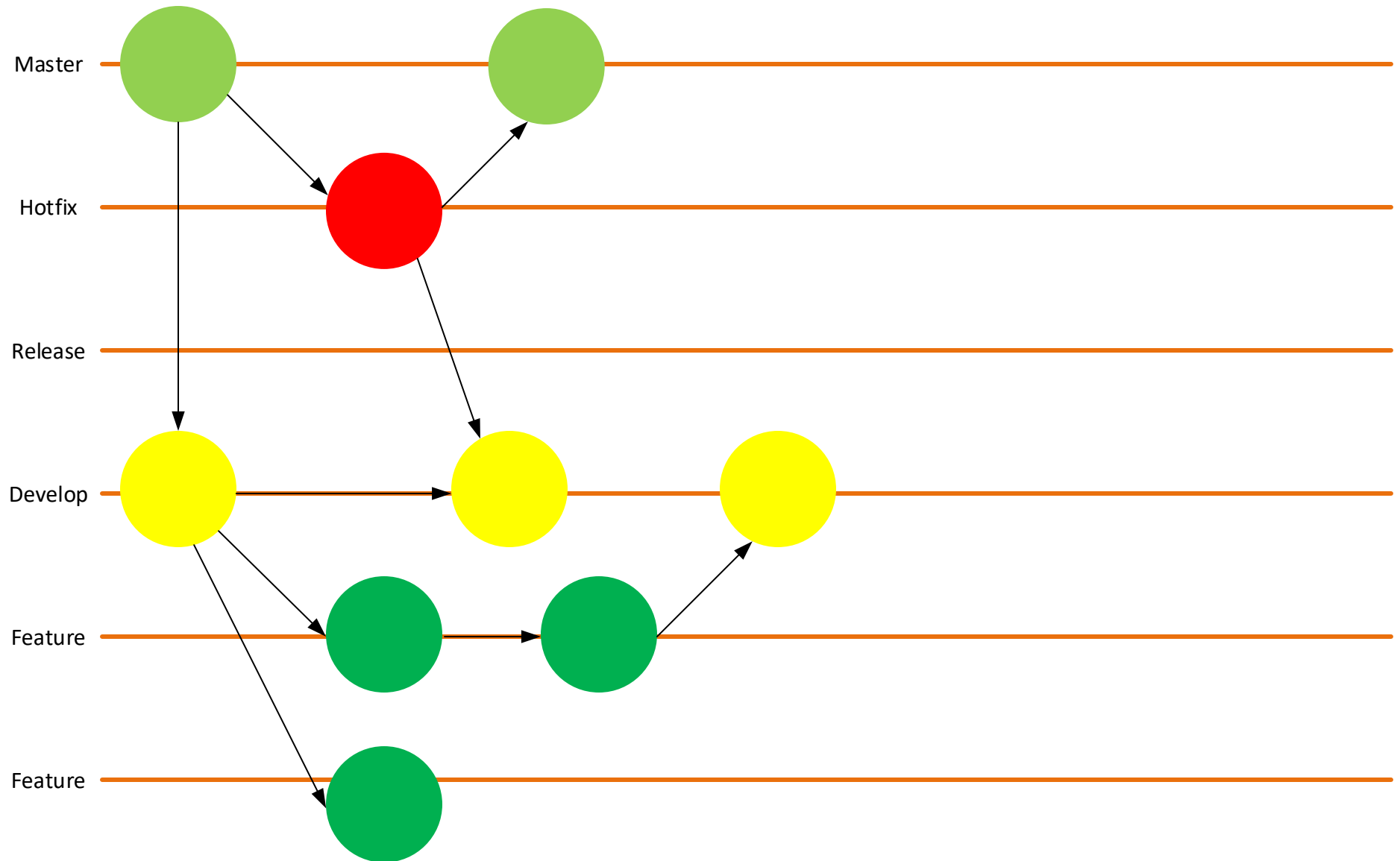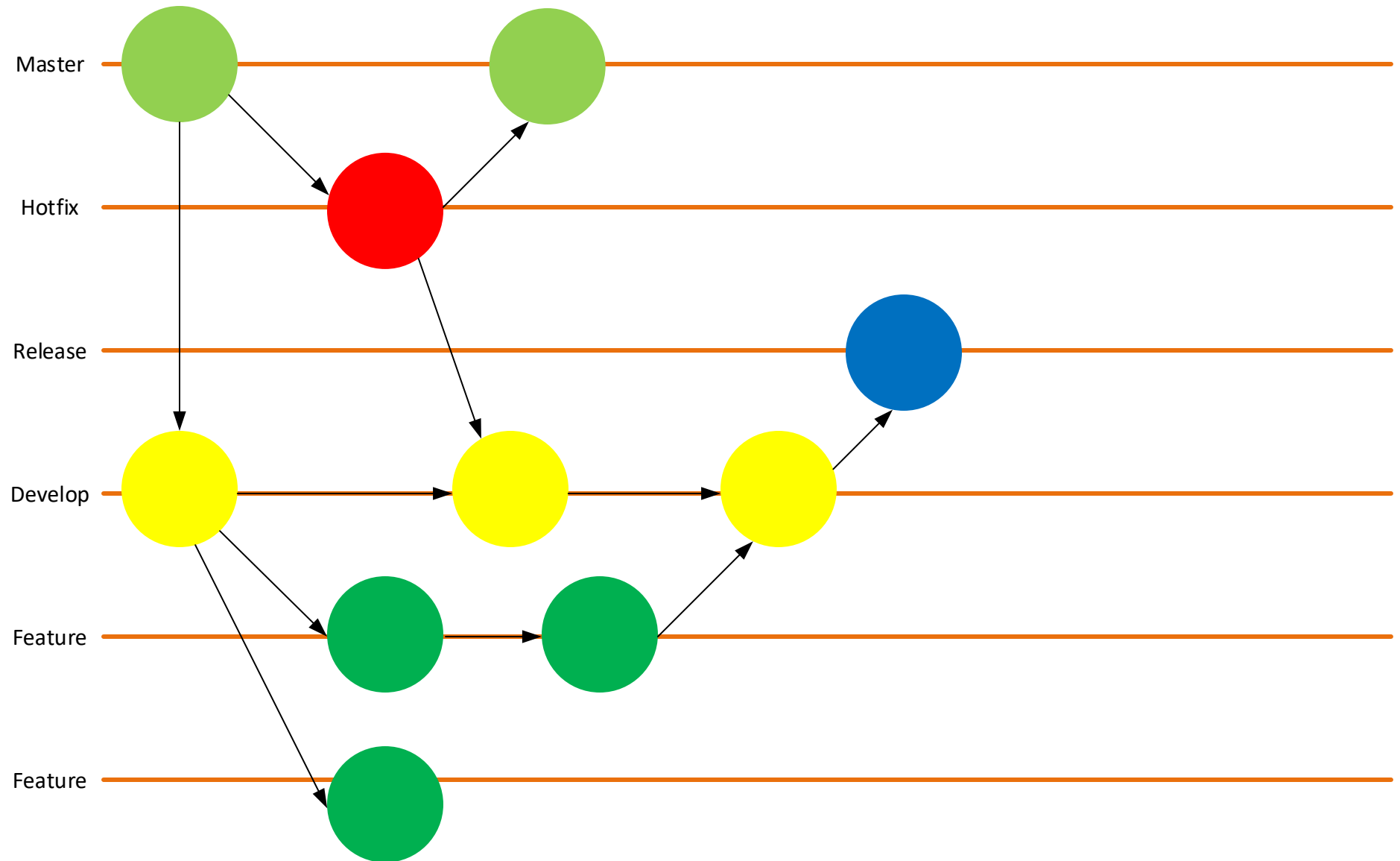
Hotfix
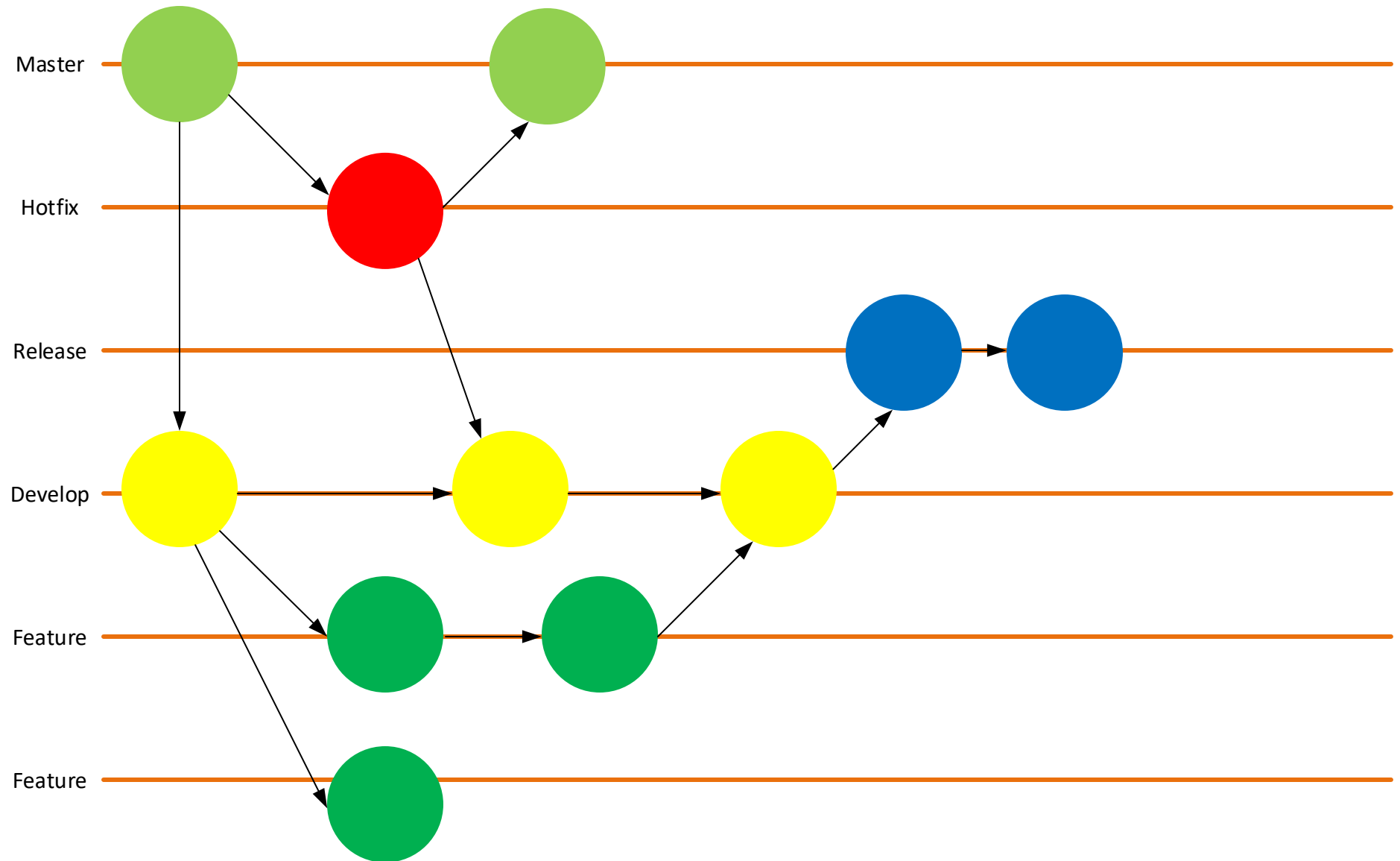
Release

Develop

Feature

Feature

Gitflow
Develop To Release

At some point, we decide we are ready to do a release, and so from the develop branch, we create a release branch. We can make any more changes on this release branch that we need.

Master

Hotfix

Release

Develop

Feature

Feature

Gitflow
Develop To Release

So, if we are testing and those tests turn up bugs, we can fix the bugs on the release branch.

Master

Hotfix
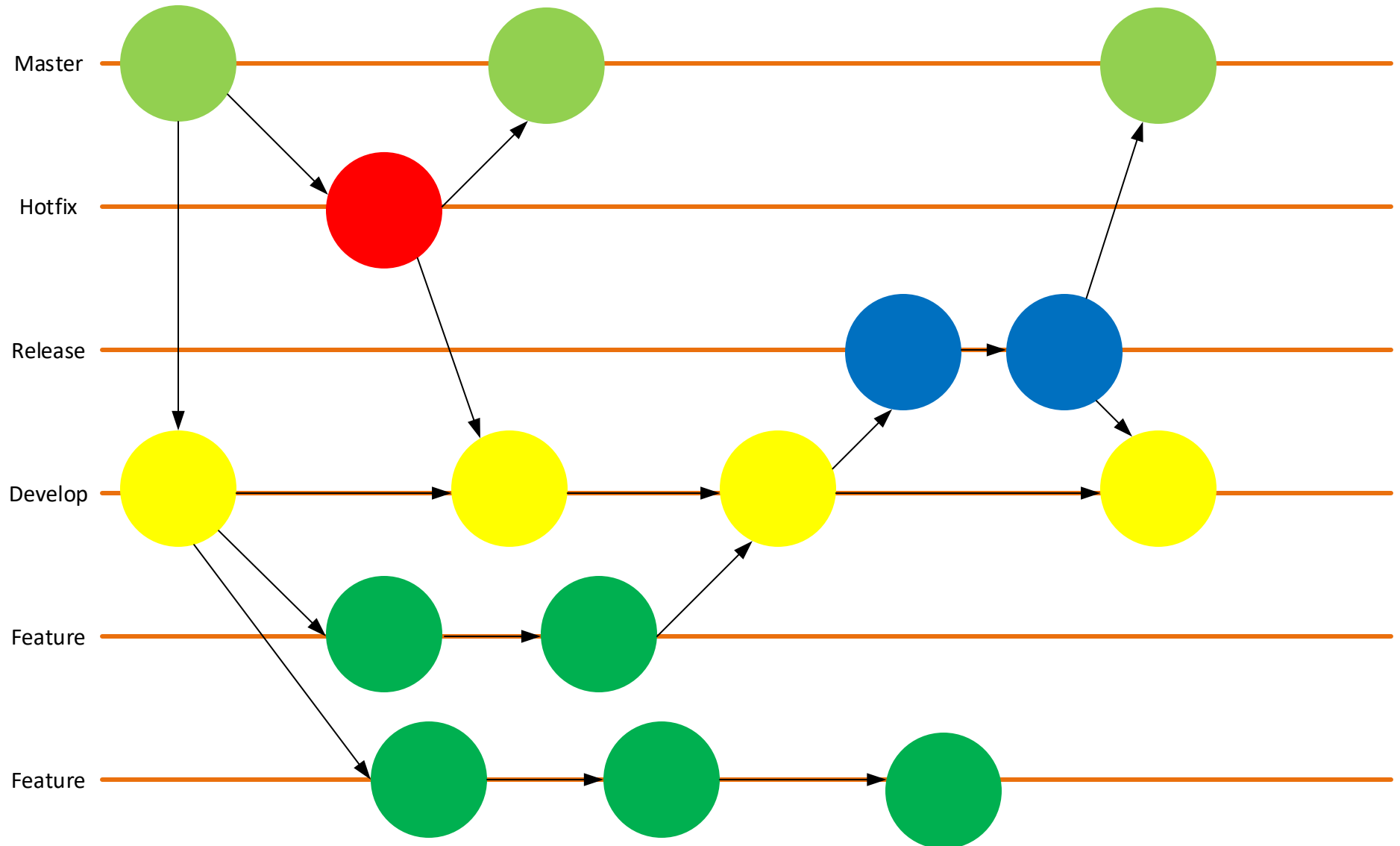
Release

Develop

Feature

Feature

Gitflow
Release To Master and Develop

Once the release goes live, those changes are checked into the master branch and back into the develop branch. So we never lose any of these changes.

Master

Hotfix

Release

Develop
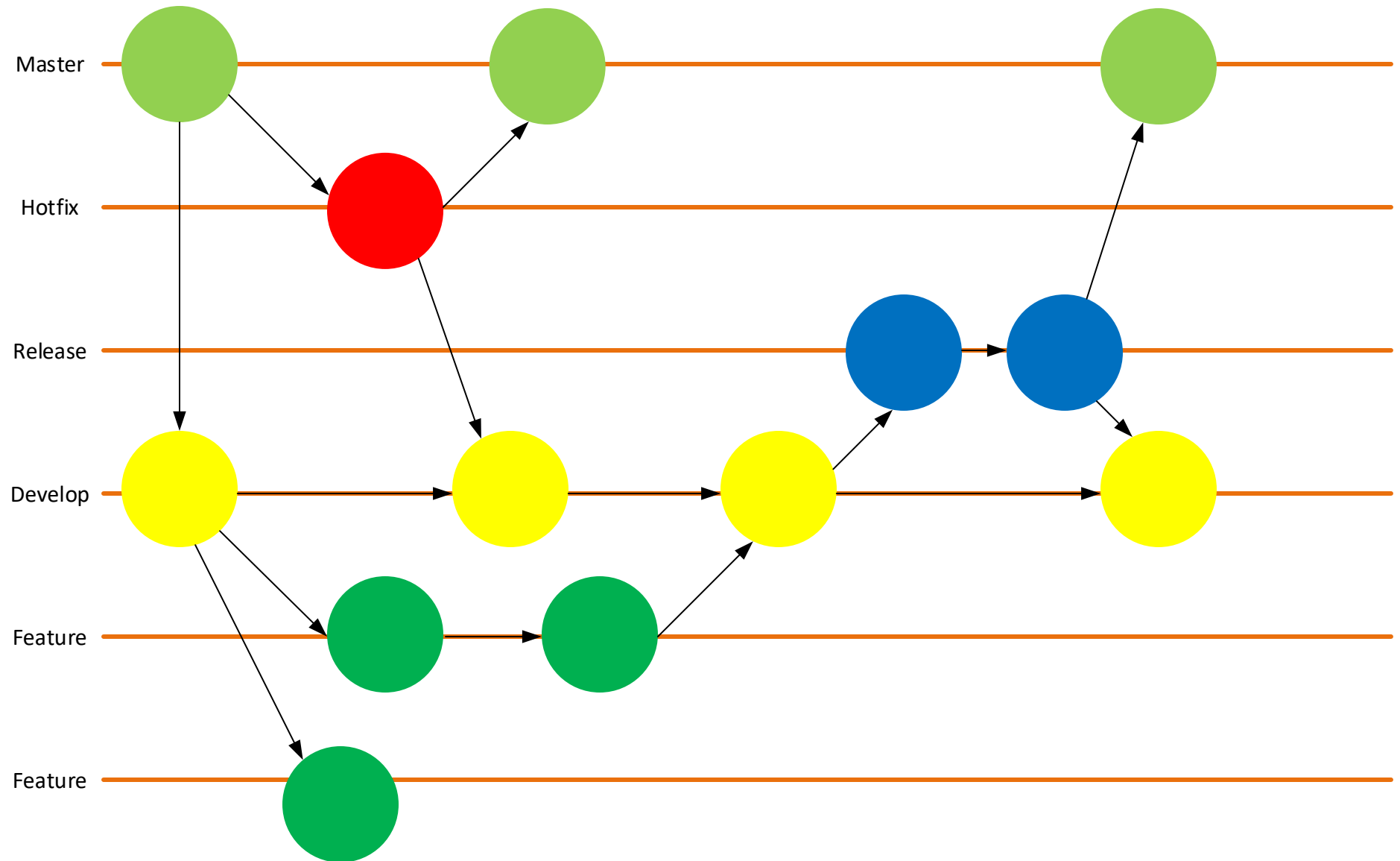
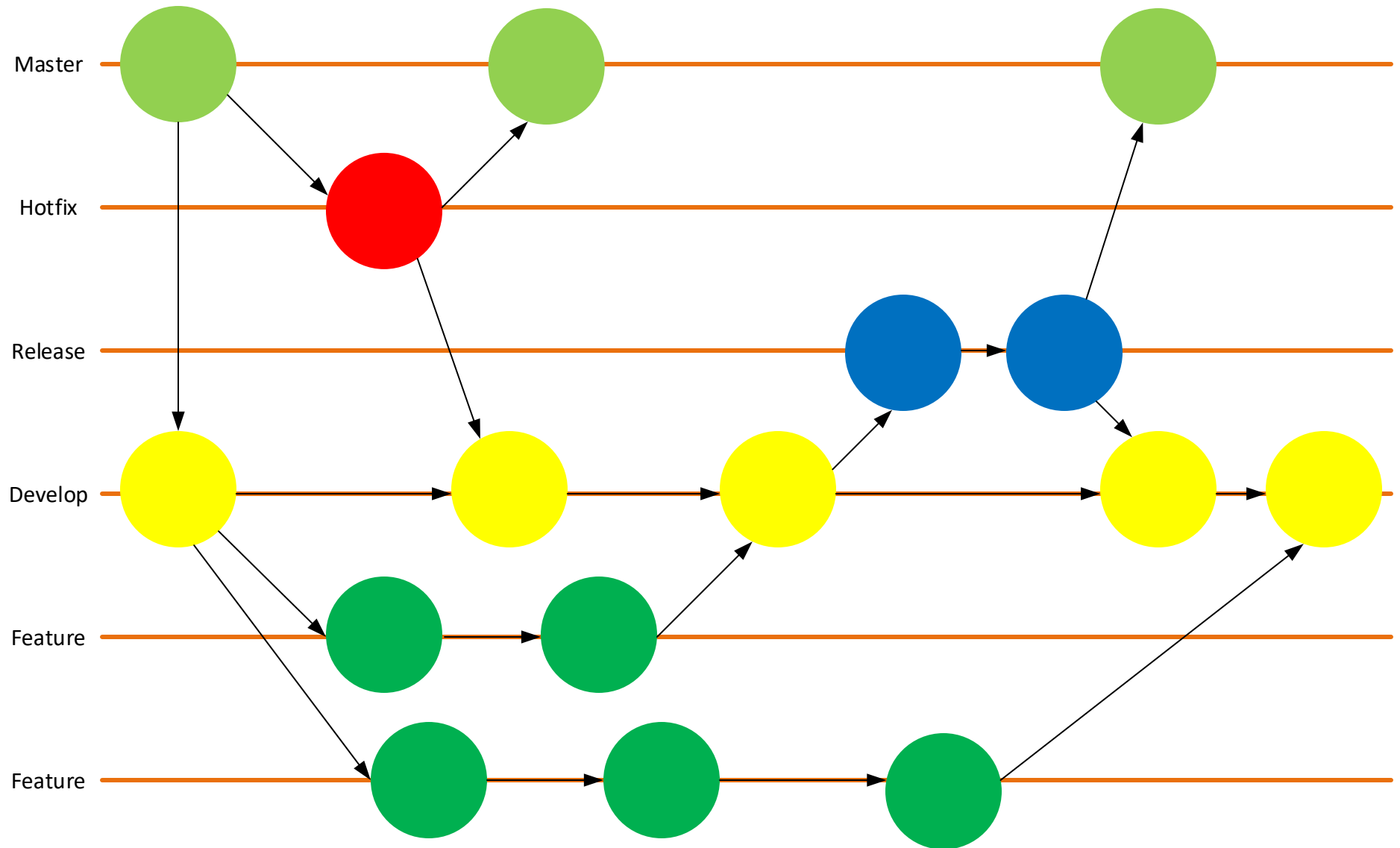Feature

Feature

Gitflow
New Feature

Still working on the feature branch.

Master

Hotfix

Release

Develop

Feature

Feature

Gitflow
New Feature to Develop
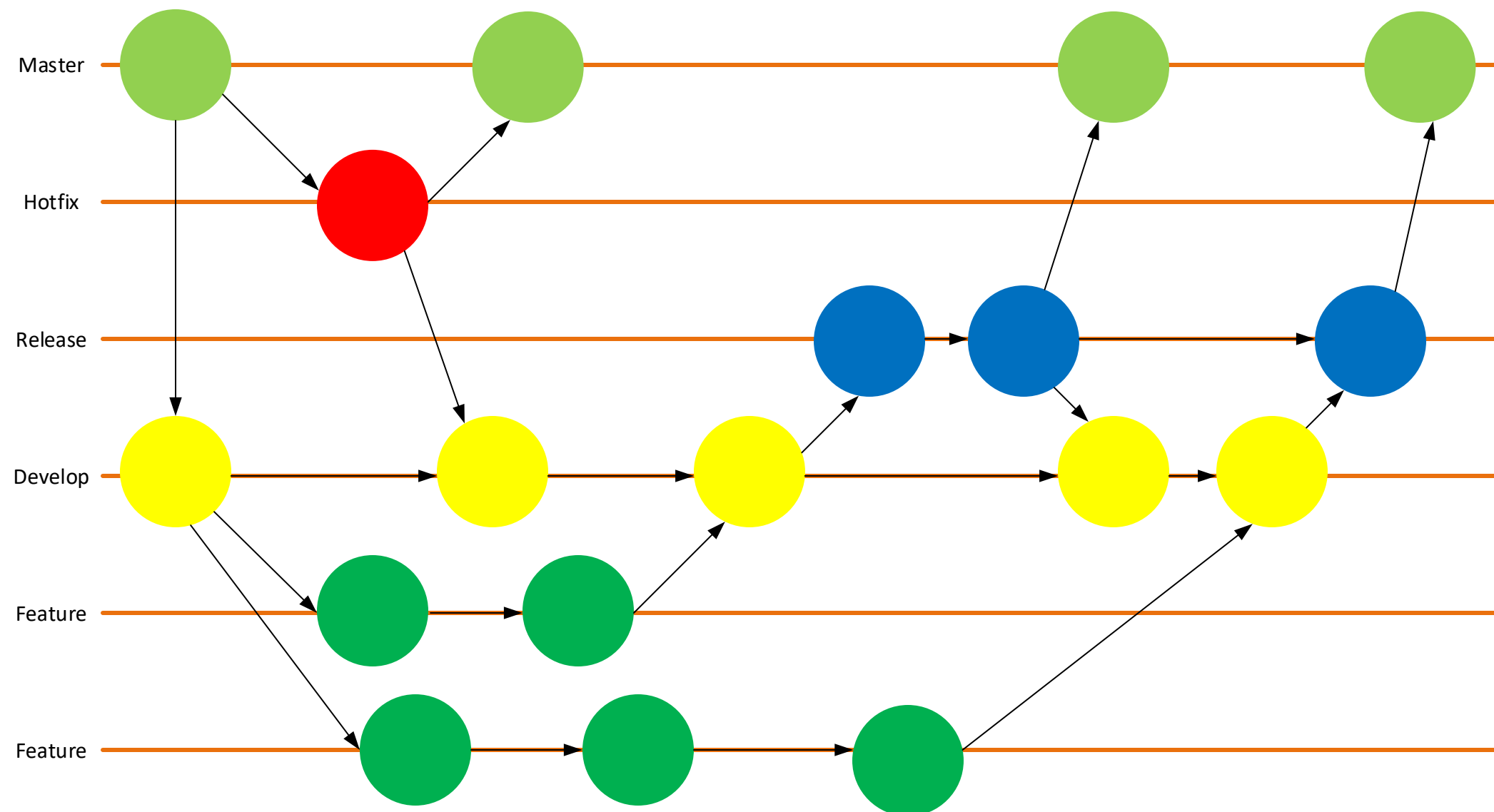
Still working on the feature branch.

Master

Hotfix

Release

Develop

Feature

Feature

Gitflow
New Feature to Develop

when that's ready to go live, we'll check those changes into a release branch, do any more work on the release branch, and then, from the release branch, check those changes back into master

Master

Hotfix

Release

Develop

Feature

Feature

Gitflow is a set of scripts that extend Git
Install it on your local machine.
It's possible to use standard Git commands.
The scripts under the covers just run a set of standard commands
Makes scripts much easier to manager
So while it would be possible to learn all the standard commands and
the syntax for the commands needed to use Gitflow, it's much easier
to just install Gitflow and to use the Gitflow commands.

So, these commands have to be installed separately to Git. And on the
Wiki for Gitflow, there are notes on how to do this installation, and
you'll find there are notes for each of the operating systems.
https://github.com/nvie/gitflow/wiki/Installation

Clone the Gitflow repository: git clone –recursive https://github.com/nvie/gitflow.git

Instsall On Your Machine Contribute\msysgit-install.cdm "c:\Profram Files\Git\"

For windows machines: grab various dlls that contain APIs that GitHub needs.

Copy those dlls under C:\Program Files\Git\bin

So the first thing to do is to clone the Gitflow repository.
git clone –recursive https://github.com/nvie/gitflow.git
next step is to install Gitflow.
It's already installed on my machine in I already cloned it into my C:\
dev repository , so it will ask me if I want to reinstall it.

I'm going to say yes. At this point, as you can see, it deletes the old
files and then copies Gitflow into the correct location on this machine.
The next step, if you're running on Windows machines, is to grab
various dlls that contain APIs that GitHub needs. These APIs are
already in place on a Linux system, but they're not in place on a
Windows system. And it tells you here where to go and get these dlls
from. So, I've already downloaded these dlls. So I already have these
in a download directory on my machine. So getop. exe, libiconv2. dll,
and libintl3. dll, and I'm simply going to copy these into my Program
Files Git bin folder. And once that's done, Gitflow is now installed.