

An Overview of

OpenSpace

Branching and Merging

How we use (or should be using)
git branches
to add features and fix bugs

Eric Myers
February 2017

Our process is well described by
"A successful Git
branching model"

By Vincent Driessen

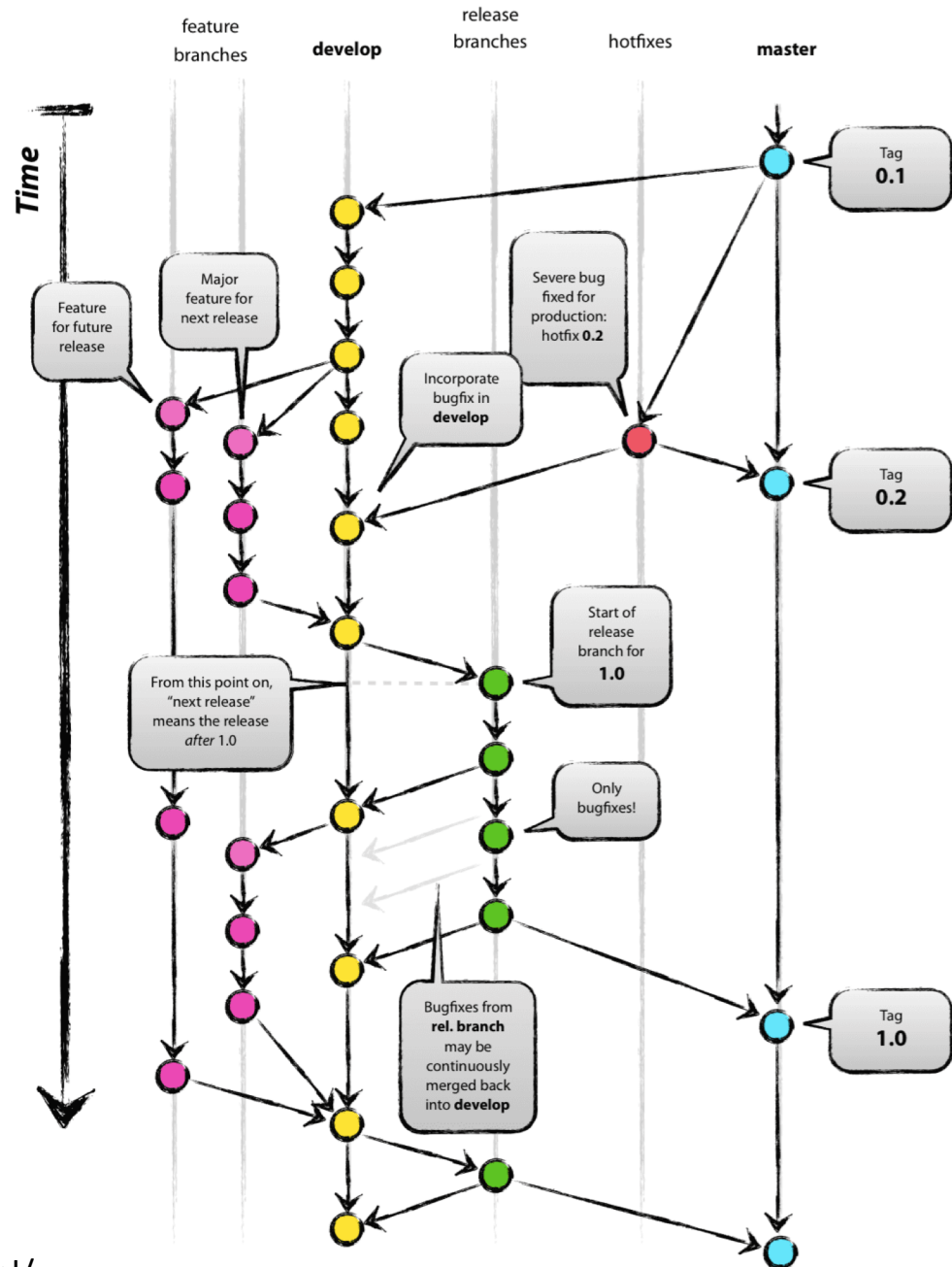
Our source code is kept on GitHub.com

Each feature is developed on a git
branch [on developer's machine(s)]

Two important branches:
 master is production code
 develop is main development

Developers work on "feature"
branches, which get merged into the
develop branch, which *on occasion* is
merged into the **master** branch (and
tagged)

<http://nvie.com/posts/a-successful-git-branching-model/>



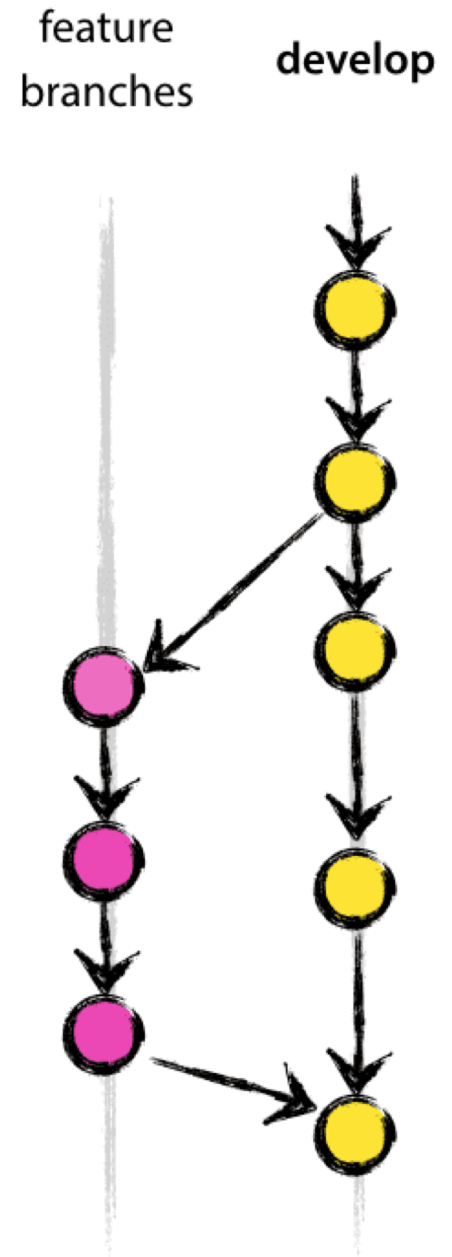
Feature Branches

Feature branches may be short lived or exist for a longer time – as long as it takes to complete the full feature.

Feature branches should **focus on only ONE feature**, not a mix of different things. A developer (or team) might work on several different branches, not just one branch that they "own". Other developers may contribute as needed.

Feature branches generally are branched from `develop`, and will later be merged back into `develop`.

Long-lived feature branches should be merged into `develop` periodically, ... AND they should periodically pull from `develop` to incorporate the latest work from others.



OpenSpace/OpenSpace Branches on GitHub

The screenshot shows the GitHub repository page for OpenSpace/OpenSpace. The repository is currently on the 'master' branch. A red arrow points to the 'Branch: master' dropdown menu, which is open, showing a list of branches. The 'develop' branch is highlighted in blue. The list of branches includes 'develop' and several feature branches like 'feature/NewAtmosphere', 'feature/downloadmanager', 'feature/dynamicRootGraph', 'feature/galaxy', 'feature/globebrowsing', 'feature/iSWA', 'feature/kameleonvolume-merge', and 'feature/kameleonvolume'. The 'develop' branch is the current branch. The repository has 3,932 commits, 21 branches, 9 releases, 15 contributors, and is licensed under MIT. The latest commit is db46b8 on Dec 14, 2016.

OpenSpace / OpenSpace

Unwatch 35 Unstar 36 Fork 7

Code Issues 78 Pull requests 1 Projects 2 Wiki Pulse Graphs Settings

The official GitHub repository for the open-source project OpenSpace <http://OpenSpaceProject.com> Edit

3,932 commits 21 branches 9 releases 15 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

Switch branches/tags

Find or create a branch...

Branches Tags

develop

feature/NewAtmosphere

feature/downloadmanager

feature/dynamicRootGraph

feature/galaxy

feature/globebrowsing

feature/iSWA

feature/kameleonvolume-merge

feature/kameleonvolume

Latest commit db46b8 on Dec 14, 2016

ch 'develop'	2 months ago
4k.xml configuration file for high resolution screenshots	5 months ago
scene folder; remove lodesritest and restructure lodglobes...	2 months ago
oul changes of moving LogLevel out of LogManager	2 months ago
oul changes of moving LogLevel out of LogManager	2 months ago
on and Scale factories from base module to OpenSpace core	2 months ago
time monitoring of StatsCollector data	7 months ago
pendent kernels in OsirisRex	2 months ago
ised shaders for postRender pass	2 months ago
oul changes of moving LogLevel out of LogManager	2 months ago
yright header	2 months ago

Release Branches

This is a new proposal,
we have not yet done this.

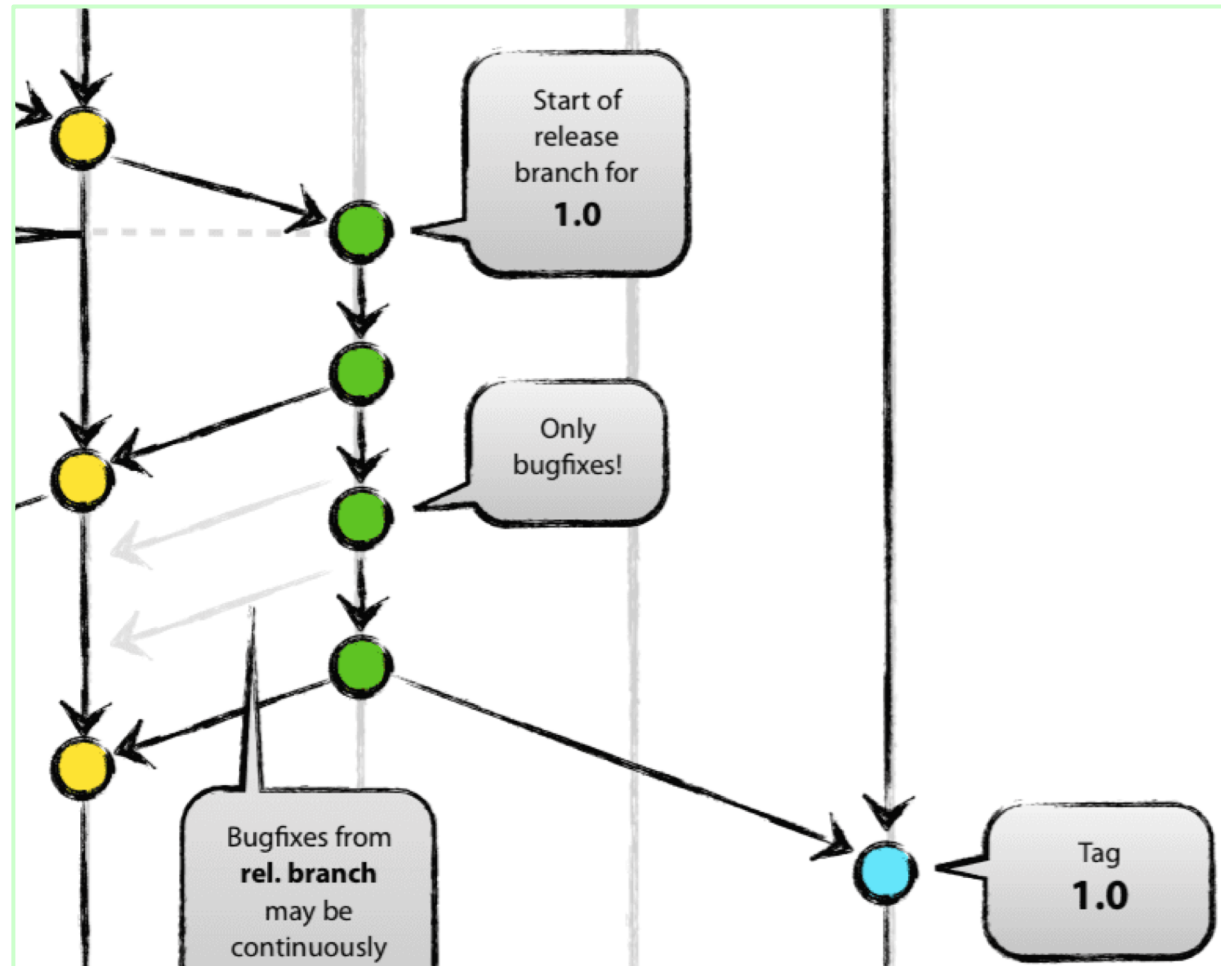
A release branch is really a "release candidate," to get ready to release a new tagged production version on master.

To begin, branch off of develop.

Test it thoroughly, on all platforms, in all ways.

Fix any bugs.

At the end, merge the results into master
AND
merge the fixes back into develop.



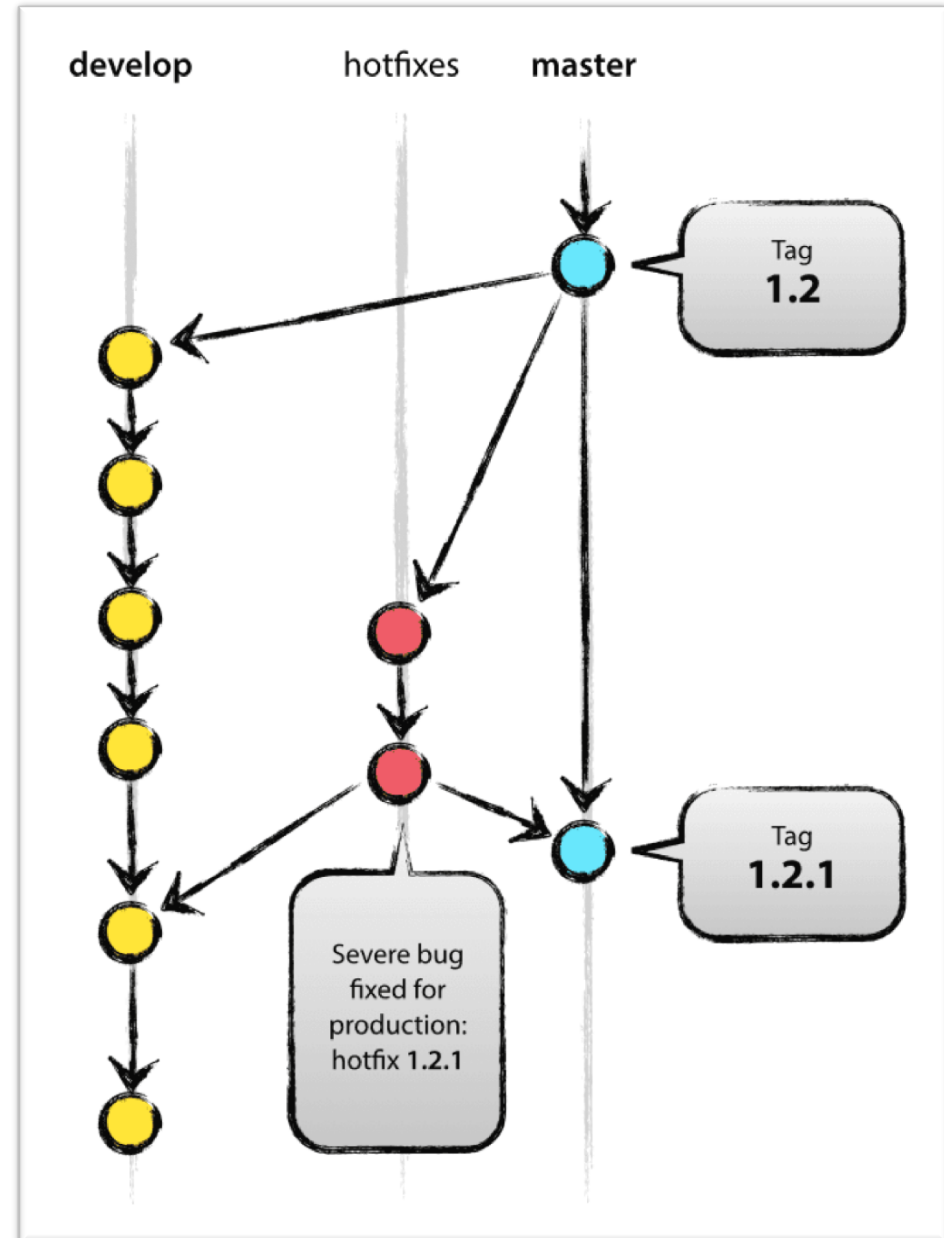
Fixing Bugs

Developers generally fix bugs in their own branches as they go along. That's what developers do.

A serious bug in a production version requires a "hotfix".

As with a release branch, corrections are merged into `master`, AND back into `develop`.

Less serious bugs can be fixed by branching off of `develop`, or a feature branch, then merging back into `develop`.



Guidelines (for developers)

Developers should always `merge` `develop` into their feature branch when the feature branch is ready to be merged with `develop`. Resolve any conflicts or problems there, then make a `git pull` request to merge the feature branch into `develop`.

Do not use `git rebase` on a public branch. And all the branches on GitHub are public branches. Instead, use `git merge`.

Bugs should be reported as "issues" on GitHub, and I suggest bug-fix branches be named after the issue (eg. "issue173").

Never use `git push` on `develop` or `master`. Instead, `push` your work to the feature branch on GitHub, then create a `git pull request` on GitHub and let the Codemasters pull into `develop` or `master` after sufficient testing.

Above and Beyond

I think of feature branches like spacecraft flying around the mothership (`develop`), and sometimes docking. And in this analogy we should remember there is a difference between "rendezvous" and "docking".

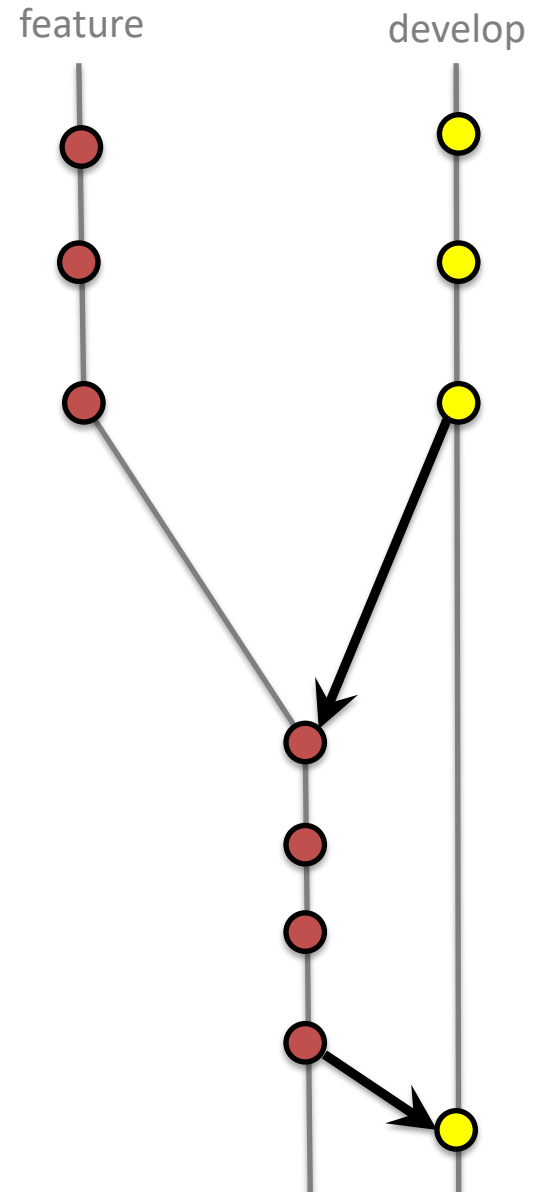
"Rendezvous" is getting close. The feature branch is getting close to ready to be pulled into `develop`. In the diagrams we've studied the feature branch now moves closer to the `develop` branch.

Before you can dock, you have to be ready. Pull from the `develop` branch, resolve any conflicts, and test the merged code.

- It must BUILD on all 3 platforms (use Jenkins)
- It must WORK on all 3 platforms.

Then create a [git pull request](#) on GitHub. The actual git merge/pull into `develop` is "docking".

NEVER BREAK the `develop` branch!

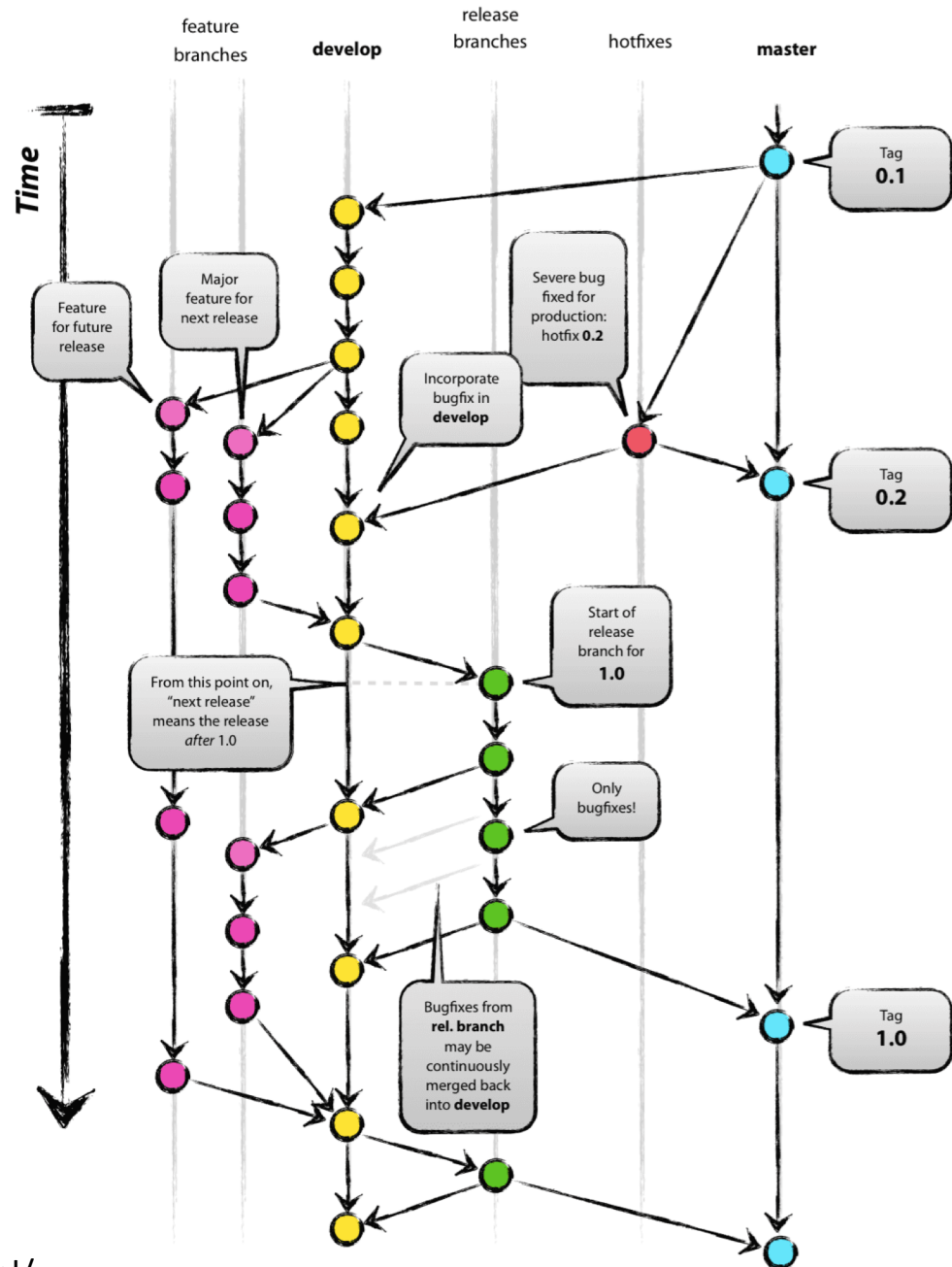


"A successful Git branching model"

By Vincent Driessen

This picture should now make a lot more sense.

The article should now make sense – read it (if you wish).



Branch Status

Branch Name	Linux	Win	MacOS	Last Commit	A B	Primary Developers
master				14Dec2016		Everybody
develop				9Jan2016	5 3	Everybody
feature/openvr				24Jan2016	8 3	Erik Sunden
feature/scenegraph-refactor				15Dec1016	3 3	Emil Axelson
feature/globebrowsing				6Dec2016	0 23	Kalle & Erik
feature/scenegraph				7Nov2016	68 197	Emil Axelson
feature/dynamicRootGraph				4Nov2016	56 197	Jonathas Costas, Emil Axelson
feature/kamelianvolume				29Oct2016	60 209	Emil Axelson
feature/NewAtmosphere				7Oct2016	6 235	Jonathas Costas
feature/downloadmanager				28Sep2016	31 258	Alex Bock
feature/starnames				20Jul2016		Matthew Territo
feature/galaxy				3Jun2016		Emil Axelson
feature/screenspaceimage				4Apr2016		Alex Bock

These 3 columns are Jenkins BUILD status. There should be 3 more columns for TEST status.

OpenSpace Releases

<u>Name</u>	<u>Version</u>	<u>Release Date</u>	
Prerelease-5	0.1.0	2015-05-14	Pluto-Palooza event at AMNH
Prerelease-7	0.2.0	2015-07-08	New Horizon's closest approach to Pluto
Prerelease-8	0.3.0	2016-04-14	CCMC
Prerelease-9	0.4.0	2016-05-31	IPS / Eurovision
Prerelease-10	0.5.0	2016-09-22	Kulturnatten
Prerelease-11	0.6.0	2016-12-09	AGU

Executables can be downloaded for the first two from

http://openspaceproject.com/?page_id=352