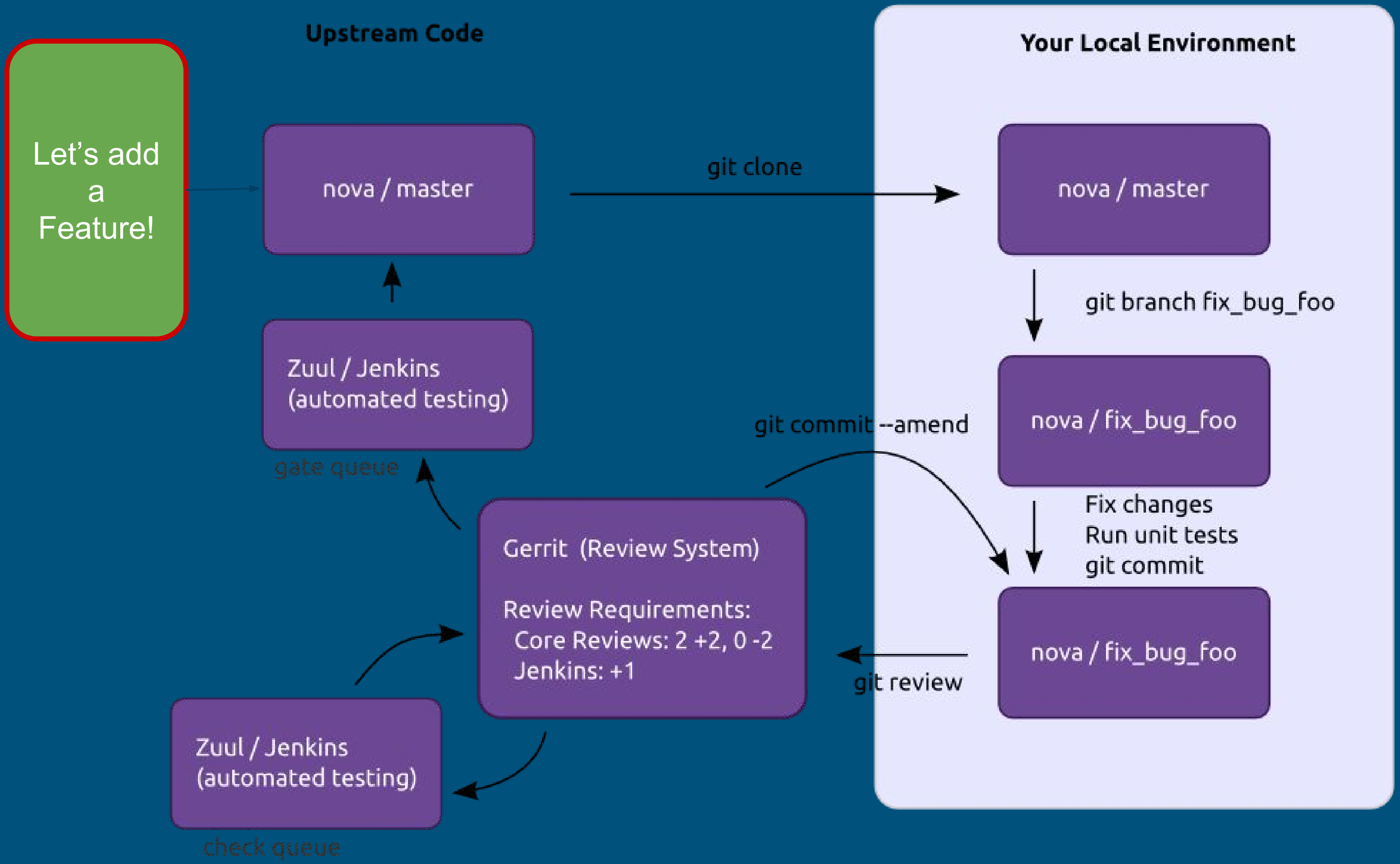
The Real Workflow of OpenStack: Managing Expectations

Kendall Nelson IRC: diablo_rojo twitter: @knelson92
Patrick East IRC: patrickeast twitter: @peast907

What’s all this about?

- Getting a new feature into OpenStack is a lengthy process and a difficult one for people that haven’t ever done it before
- Open Collaboration is good, but it can slow down the process and cause unexpected delays

Let’s Add a Feature!



Let's add a Feature!

Upstream Code

nova / master

Zuul / Jenkins
(automated testing)

gate queue

Gerrit (Review System)

Review Requirements:
Core Reviews: 2 +2, 0 -2
Jenkins: +1

Zuul / Jenkins
(automated testing)

check queue

git clone

git commit --amend

git review

Your Local Environment

nova / master

git branch fix_bug_foo

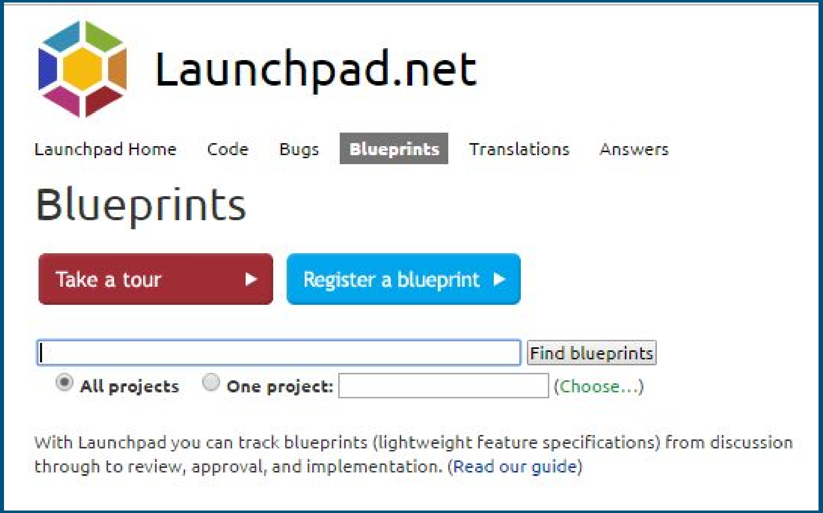
nova / fix_bug_foo

Fix changes
Run unit tests
git commit

nova / fix_bug_foo

What is a Blueprint?

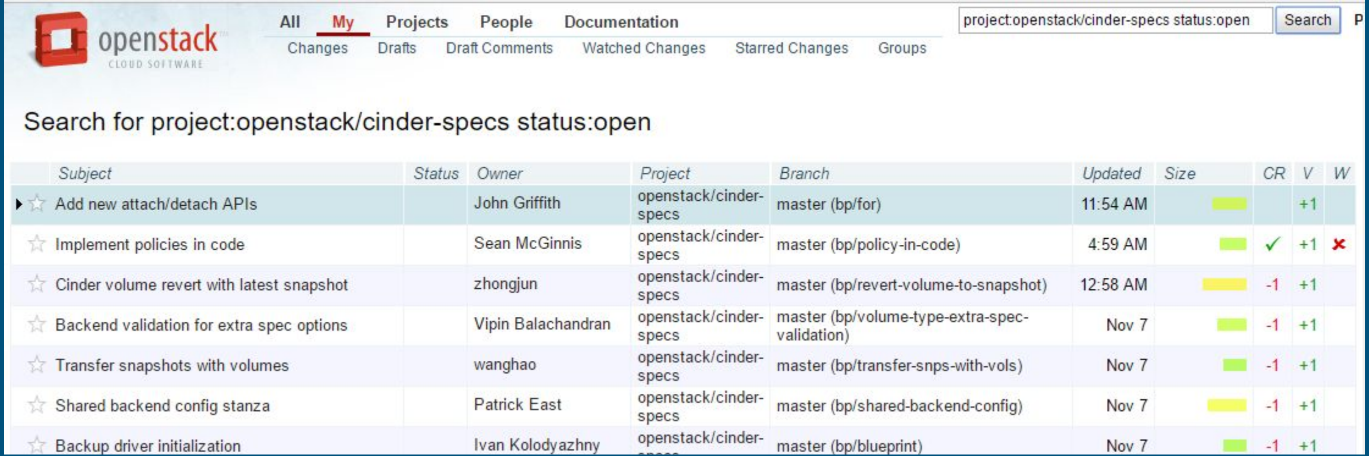
- https://wiki.openstack.org/wiki/Blueprints
- Do I actually need a blueprint?
  - Depends… check with your project



With Launchpad you can track blueprints (lightweight feature specifications) from discussion through to review, approval, and implementation. (Read our guide)

Spec? What? Where?

- Do I actually need a spec?
  - Depends… check with your project
- http://specs.openstack.org/
- Gerrit review process

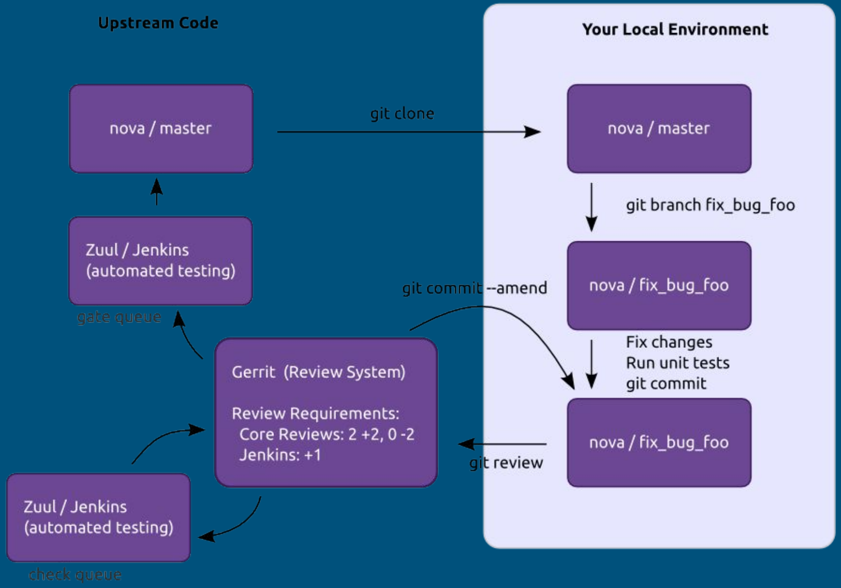


Search for project:openstack/cinder-specs status:open

Subject	Status	Owner	Project	Branch	Updated	Size	CR	V	W
Add new attach/detach APIs		John Griffith	openstack/cinder-specs	master (bp/for)	11:54 AM	🟩		+1	
Implement policies in code		Sean McGinnis	openstack/cinder-specs	master (bp/policy-in-code)	4:59 AM	🟩	✓	+1	✗
Cinder volume revert with latest snapshot		zhongjun	openstack/cinder-specs	master (bp/revert-volume-to-snapshot)	12:58 AM	🟨	-1	+1	
Backend validation for extra spec options		Vipin Balachandran	openstack/cinder-specs	master (bp/volume-type-extra-spec-validation)	Nov 7	🟩	-1	+1	
Transfer snapshots with volumes		wanghao	openstack/cinder-specs	master (bp/transfer-snps-with-vols)	Nov 7	🟩	-1	+1	
Shared backend config stanza		Patrick East	openstack/cinder-specs	master (bp/shared-backend-config)	Nov 7	🟨	-1	+1	
Backup driver initialization		Ivan Kolodyazhny	openstack/cinder-	master (bp/blueprint)	Nov 7	🟩	-1	+1	

Remember that nice flow chart?

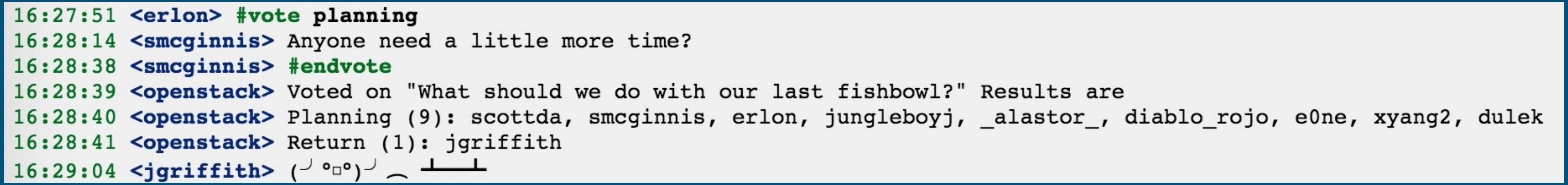
- So... gonna need to go through this workflow for a spec before you get your code in



X 2

The Meeting

- Somewhere along the way there will probably be at least one meeting to decide the fate of your feature
- What? Really?
  - Depends… some projects tend to use meetings to discuss features and design, others don’t
  - Sometimes less serious than others...



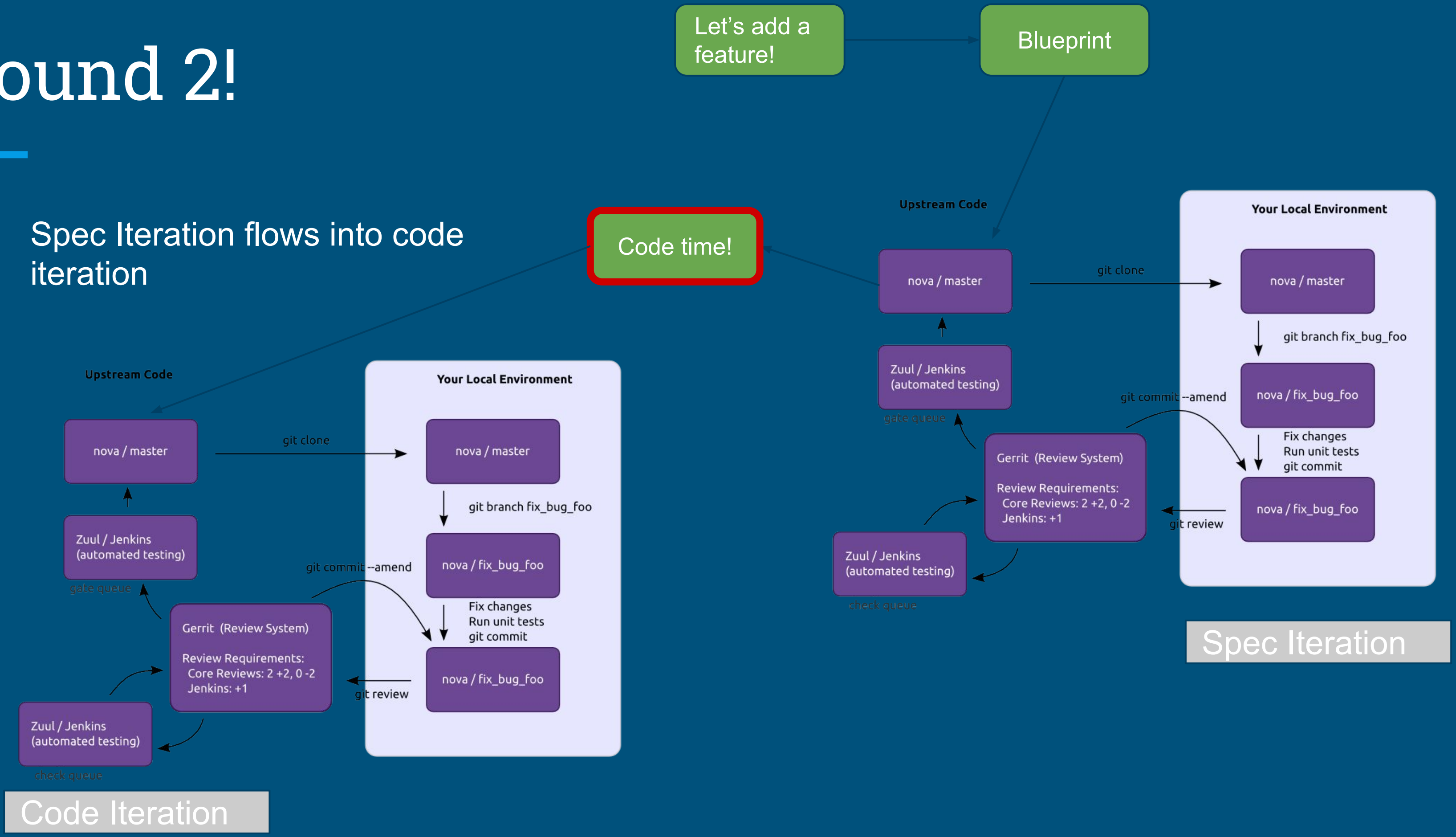
```
16:27:51 <erlon> #vote planning
16:28:14 <smcginnis> Anyone need a little more time?
16:28:38 <smcginnis> #endvote
16:28:39 <openstack> Voted on "What should we do with our last fishbowl?" Results are
16:28:40 <openstack> Planning (9): scottda, smcginnis, erlon, jungleboyj, _alastor_, diablo_rojo, e0ne, xyang2, dulek
16:28:41 <openstack> Return (1): jgriffith
16:29:04 <jgriffith> (╯°□°)╯︵ ┻━┻
```

Spec has been Merged, Now What?

- Spec & Blueprint Approved
- Code time!

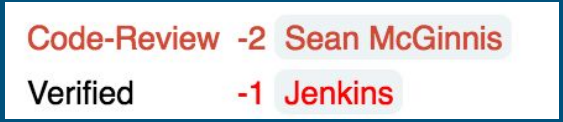
Round 2!

- Spec Iteration flows into code iteration

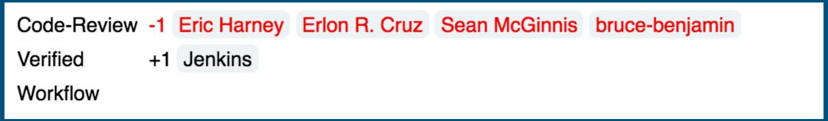


Implemented the Code...and Everyone Has an Opinion about it...

- -1's? -2? What?!
- Don't sweat it...
- Time to increment the patchset



Code-Review -2 Sean McGinnis
Verified -1 Jenkins



Code-Review -1 Eric Harney Erlon R. Cruz Sean McGinnis bruce-benjamin
Verified +1 Jenkins
Workflow

Feature Freeze!

- It has a way of sneaking up on you…
- Don’t forget there are different freeze dates for different parts of the process
  - https://releases.openstack.org/
- Not all hope is lost! Look for the Feature Freeze Exception emails!

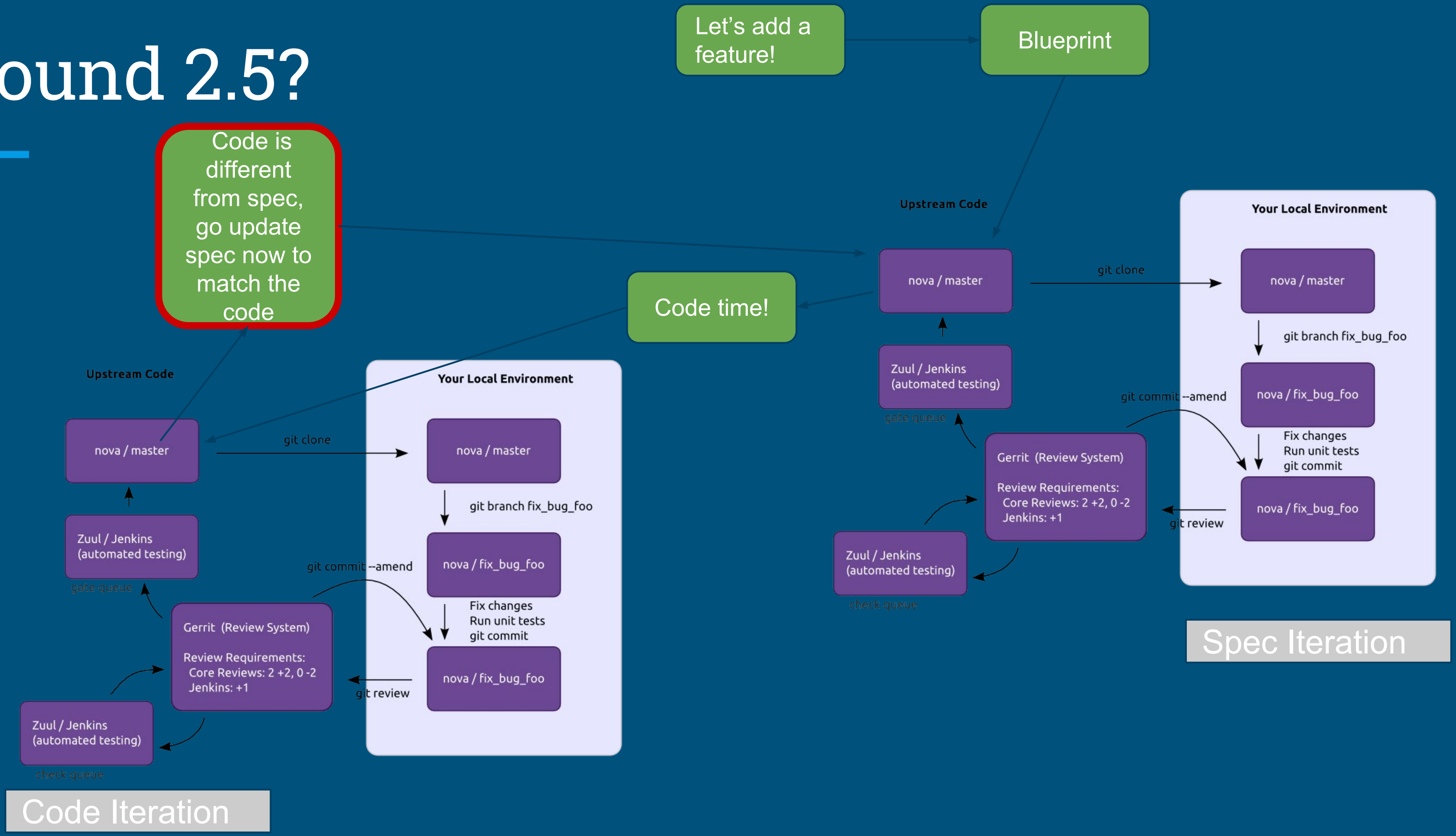
Working for the Common Good

- Design by committee
- Everyone has a use-case and an opinion
- Features typically cater to the lowest common denominator
- Edge cases for atypical deployments, hardware, and use-cases need to be handled

Update Spec

- Update the spec to match the actual code implementation
- Get it reviewed and agreed upon
    - ...again
- Move it to the right release directory!

Round 2.5?



Let's add a feature!

Blueprint

Spec Iteration

Upstream Code

nova / master

Zuul / Jenkins (automated testing)

gate queue

Gerrit (Review System)

Review Requirements:
Core Reviews: 2 +2, 0 -2
Jenkins: +1

Zuul / Jenkins (automated testing)

check queue

Your Local Environment

git clone

nova / master

git branch fix_bug_foo

nova / fix_bug_foo

Fix changes
Run unit tests
git commit

git commit --amend

nova / fix_bug_foo

git review

Code time!

Code Iteration

Upstream Code

nova / master

Zuul / Jenkins (automated testing)

gate queue

Gerrit (Review System)

Review Requirements:
Core Reviews: 2 +2, 0 -2
Jenkins: +1

Zuul / Jenkins (automated testing)

check queue

Your Local Environment

git clone

nova / master

git branch fix_bug_foo

nova / fix_bug_foo

Fix changes
Run unit tests
git commit

git commit --amend

nova / fix_bug_foo

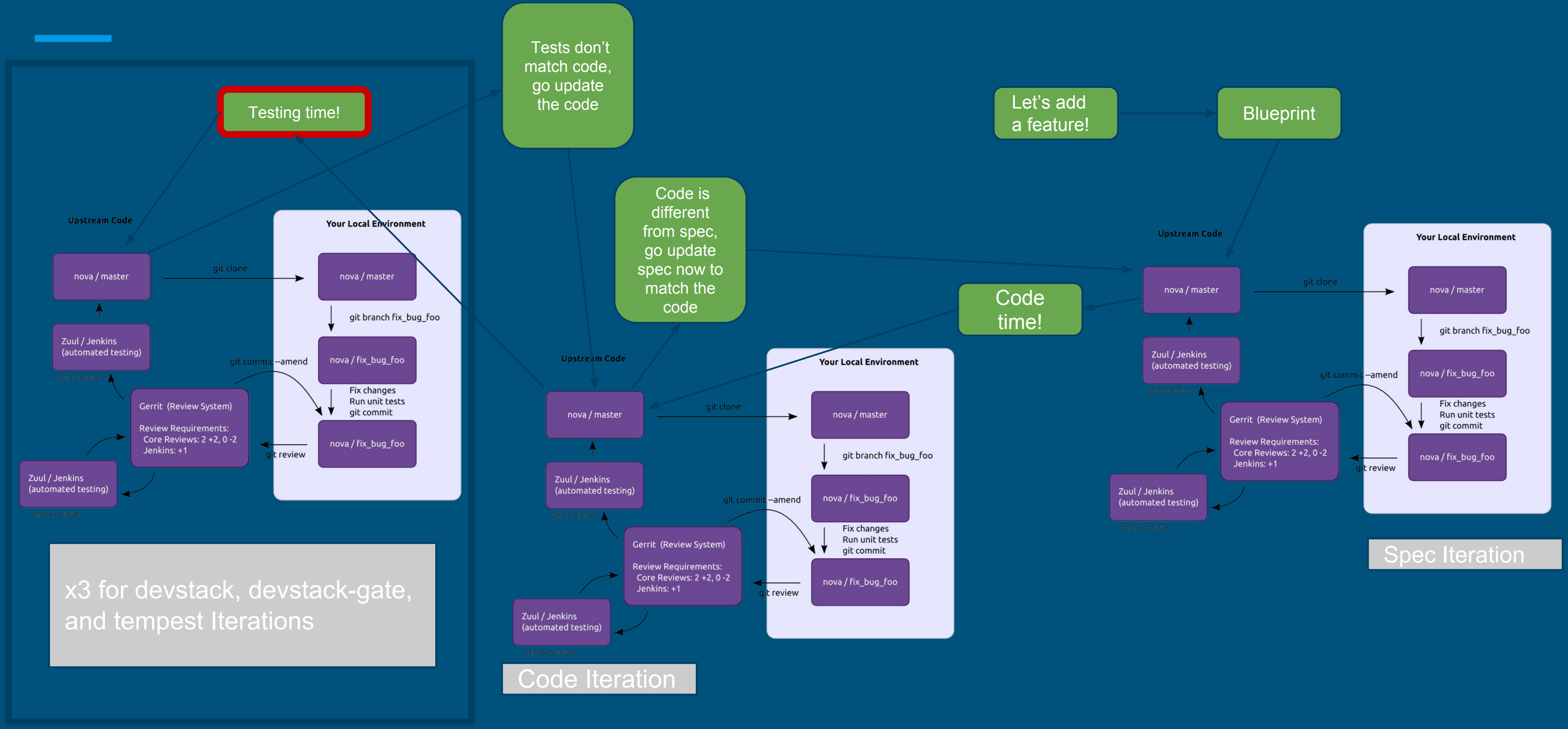
git review

Code is different from spec, go update spec now to match the code

Test the Code

- Need more than just CI and Jenkins to pass
- Functional tests? Unit tests? QA?
- Ask others to test on their hardware

Round.. 5 to x ?



Testing time!

Tests don't match code, go update the code

Code is different from spec, go update spec now to match the code

Code time!

Let's add a feature!

Blueprint

Upstream Code

nova / master

Zuul / Jenkins (automated testing)

gate queue

Gerrit (Review System)

Review Requirements:
Core Reviews: 2 +2, 0 -2
Jenkins: +1

Zuul / Jenkins (automated testing)

check queue

Your Local Environment

git clone

nova / master

git branch fix_bug_foo

nova / fix_bug_foo

Fix changes
Run unit tests
git commit

nova / fix_bug_foo

git commit --amend

git review

x3 for devstack, devstack-gate, and tempest Iterations

Code Iteration

Spec Iteration

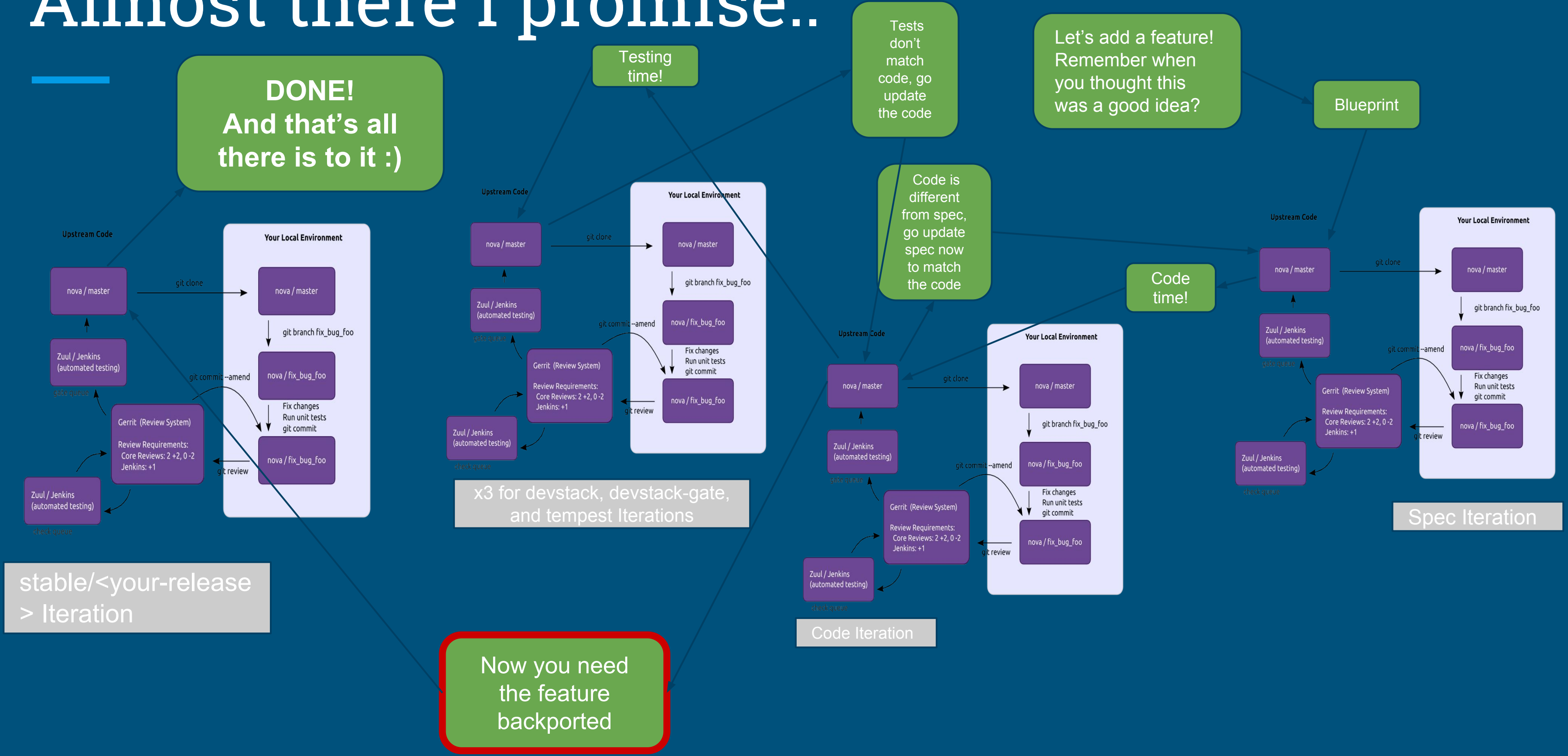
Hurdles to Merging the Code!

- Merge conflict?!
- Jenkins failure?
  - http://status.openstack.org/zuul/
  - http://status.openstack.org/openstack-health/#/
- Dependent patches?

It’s in Master, Now Where is my Feature?

- Chances are you aren’t running on master (maybe not even the most recent release) and need to to be backported
- Release outside of upstream maintenance?
- Ask <insert distro here> nicely to backport it to your current release (is possible)

Almost there I promise..



Let's add a feature! Remember when you thought this was a good idea?

Blueprint

Spec Iteration

Code time!

Code Iteration

Code is different from spec, go update spec now to match the code

Tests don't match code, go update the code

Testing time!

x3 for devstack, devstack-gate, and tempest Iterations

Now you need the feature backported

stable/<your-release > Iteration

DONE! And that's all there is to it :)

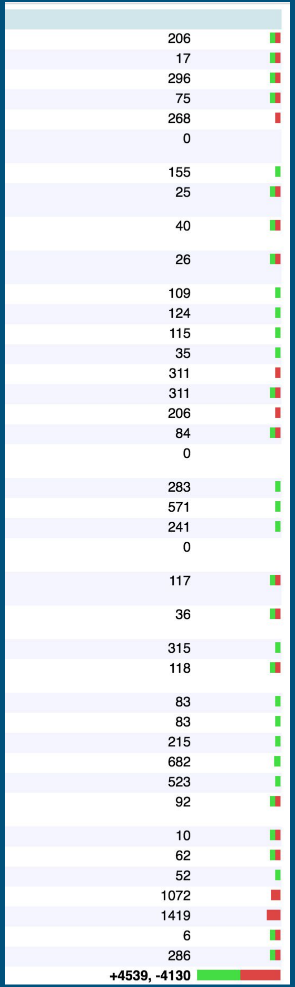
Why Did this Suck so Much?

Can it be easier?

Baby Steps

A 10k Line Patch Isn’t A Good First Impression

- Start small
  - Learn the process
  - Meet the reviewers
- Big patches can be hard to get review attention on
- Generally need a lot of help to merge
- Rebase hell
- Probability of introducing bugs skyrockets
- Start with making friends not LOC to get involved



```
          206
           17
          296
           75
          268
            0

          155
           25

           40

           26

          109
          124
          115
           35
          311
          311
          206
           84
            0

          283
          571
          241
            0

          117

           36

          315
          118

           83
           83
          215
          682
          523
           92

           10
           62
           52
         1072
         1419
            6
          286
+4539, -4130
```

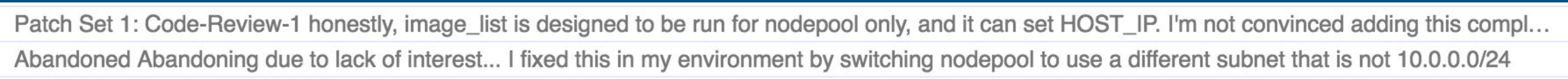
Contributors are individuals

- Don’t rely on your company’s to get code merged
- The community sees you as a single person
  - Need to build street cred and that may take time depending on the project

It’s Not Personal

-1’s Are Not The End of The World

- Don’t stress it
- Could mean anything, usually it’s a typo or formatting issue with your comments
  - Unless its jenkins…
- Doesn’t mean the code is bad or can’t ever merge



Patch Set 1: Code-Review-1 honestly, image_list is designed to be run for nodepool only, and it can set HOST_IP. I'm not convinced adding this compl…

Abandoned Abandoning due to lack of interest... I fixed this in my environment by switching nodepool to use a different subnet that is not 10.0.0.0/24

Someone actually reviewed your patch!

- Good news!
- Someone cares enough to look at your code!
- Negative reviews are progress too
- Any attention is usually good attention for a patch

-2's May Not Be Final…

- Sometimes… maybe…
- General indication that more discussion might be needed
  - Weekly-meeting
  - PTG
  - Summit
- Can be reversed with communication and/or compromise
  - Sometimes it's a misunderstanding

You Can Say No To Review Feedback

- Not all reviewers are right 100% of the time
  - Yes, even the project cores and PTL’s
- If you get negative feedback and disagree, it’s okay to push back
- Explain your side and ask for them to clarify their points
- Involve more people
  - Weekly meetings are good
  - Openstack-dev mailing list
  - IRC

Keep Your Cool

- Remember that you are dealing with other humans
- Keep things civil, even if they are in the wrong
- Consider other viewpoints
- (╯°□°)╯︵ ┻━┻

Working With Others

How to Make Friends

- Actively participate in the community (mid-cycles/ PTG, meetups, beer NDA's, etc)
- Give back to the Community
  - Answer other people's questions that post in IRC
  - Code Reviews
  - Test other's code
  - Triage bugs
  - Respond to ML threads
- You can't expect something if you give nothing

Changes Impact Everyone

- It is a huge community
- Affect multiple distro’s, vendors, deployments
- Requires feedback and buy-in from diverse community

Setting Expectations

Features & Bug Fixes Are Sometimes Slow

- Even small patches can take a very very long time
- Depends on..
  - Complexity of change
  - How controversial the change is
  - Project activity
  - Priority
  - Severity
- Some trivial changes can take months to merge

Making Promises

- Be careful about promising feature X will be in release milestone Y
    - Until it actually merges all bets are off..and even then..
    - Don’t rely on FFE
- Targeting features and bug fixes for a release milestone is a guide not a commitment from the project team

Can’t Plan Like In-House Projects

- Normal project planning with milestones doesn’t always work very well
  - Hard to predict landing features
  - If they slip a freeze date they are postponed ~6 months
- Resources you require aren’t usually all on your team
  - The larger features will require time and resources from other companies, they all have their own schedules and goals too

Better Way To Plan?

- Look at release schedule
  - More direction about when things need to be done by to make the cut
- Anticipate delays due to the process
- Be involved in conversations about release priorities
- Be involved in the community and review other code
  - Participating takes time, plan for it

Why are we this masochistic?

- It’s not always this bad
  - We pick on extreme examples...
- Features are better tested
- Bugs get fixed quickly because they affect everyone
- Your company doesn’t do ALL the work maintaining the core code
  - Shared responsibility
  - Get to benefit from the hard work of others

TL;DR:

- The process of getting a new feature is often a long and difficult one
- The community is a part of the equation, it is not just writing code
- This is open source, it is difficult to quickly force in new things
- Be wary of making promises to management... it’s not over until the code merges and ships downstream

Questions??

Thanks!