Getting VOLTHA to Production

The Boring Stuff

ONF Connect – San Jose, California, United States
September 10-13, 2019 2:45p PDT

What Does Production Ready Mean?

Value in production

- Is Usable
- Does something useful
- Perhaps not everything we want it to eventually do

Predictable [behavior and resource usage] over [extended periods of] time

- It works

Supportable

- Documentation on how to operate
- Ability to troubleshoot and correct with restarting world

Is a Production Quality VOLTHA Needed?

"It is necessity and not pleasure that compels us"

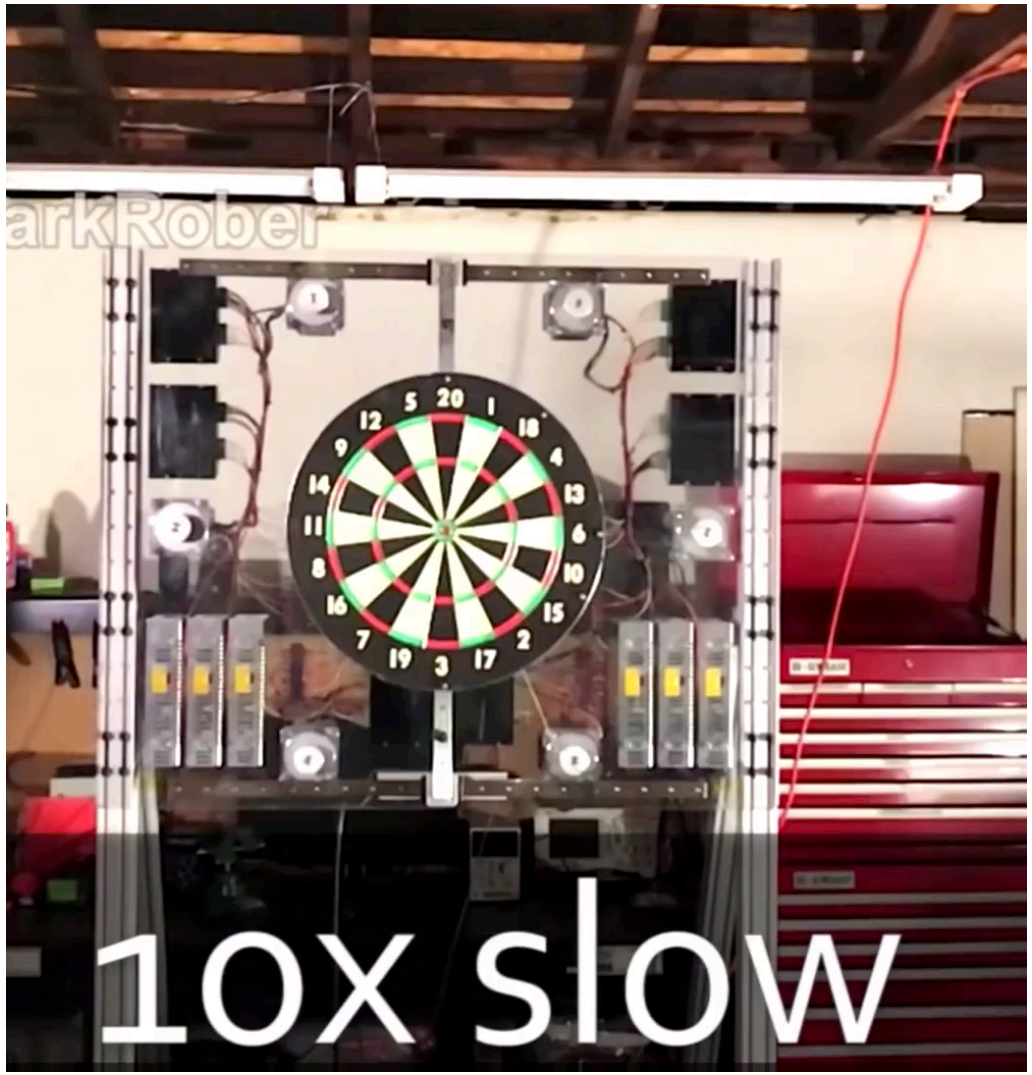
Dante Alighieri, Inferno

"If necessity is the mother of invention, then dissatisfaction must be its father."

Jeffrey Fry

Proverb - “You can not fix a moving train”

Test Test Test Test



Develop How We Deploy

- Self-test in HA environment before commit

Comprehensive Unit Tests

- No Test / No Commit
- Positive / Negative Use Cases
- Significant Code Coverage
- Regression

Integration Tests

- Spin up all components
- Positive / Negative Tests
- Regression

Cause Chaos

- Integrate “Chaos Monkey” into CI/CD
- Test [unexpected] Failure Modes
- Regression

Stop The Train (well, “slow” it anyway)

The Suggestion

Canary

- Permissive
- Fast evolving for new features
- Experimental development or changes to existing features
- Unstable
- Useful for “forward looking” conference demonstrations
- Frequent “releases” (1 month to 3 months)

Production (LTS)

- Restrictive
- Emphasize stability over features
- Very minimal feature set
- Complete documentation
- Long “release” cycle (6 month or 1 year)

Simplify

Focus on Key Architecture Requirements

Usability, Stability, Supportability, Scale, and Performance

- VOLTHA Core 2.0
- HSIA
- Scale Target [Minimal required for target production deployment]
- Performance Target [Minimal required for target production deployment]
- Static Technology Profile
- No Multicast
- GRPC API only
- *Technical Profile*
- *Event format normalization*

Migration

LTS changes should be periodically merged to Canary

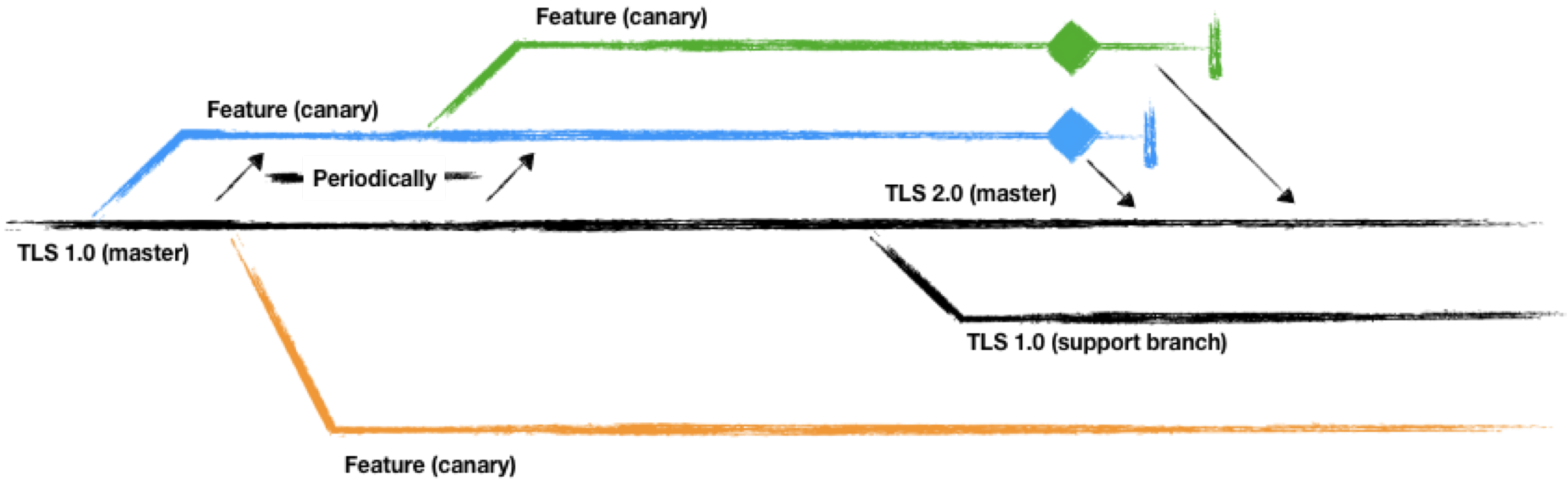
Possibly significant differences between to LTS and Canary at time of merge

Long term schedule / plan for migrate features to LTS

Stringently vet features before the migrate to LTS

- Developers comfortable with stability of implementation
- Have been thoroughly tested
- Rate of change in code near zero
- Documentation required
- Beware of complex feature dependency graphs
- Default to “no” until proven “required”

Migration



Aren’t We Doing This Already?

No

- No common base (1.x v 2.x)
- Evolving core and features at same time in multiple branches
- Lenient about what is a feature and what is a bug fix

To be fair

- We are at an inflection point
- But, we need to put serious effort on stability going forward

Why Won't Improved Testing on master Work?

Our track record isn't good

- Community has good intentions and good testing teams at ONF
- Community tends to develop to the "next conference demo"
- Community has a lot of "explorers" and it is fun to explore

This isn't a "one time" or "one sprint" effort

- Continual focus on a production release
- Emphasis on stability over features

Can’t a Vendor Do This Independently?

Yes, do you want a vendor specific fork?

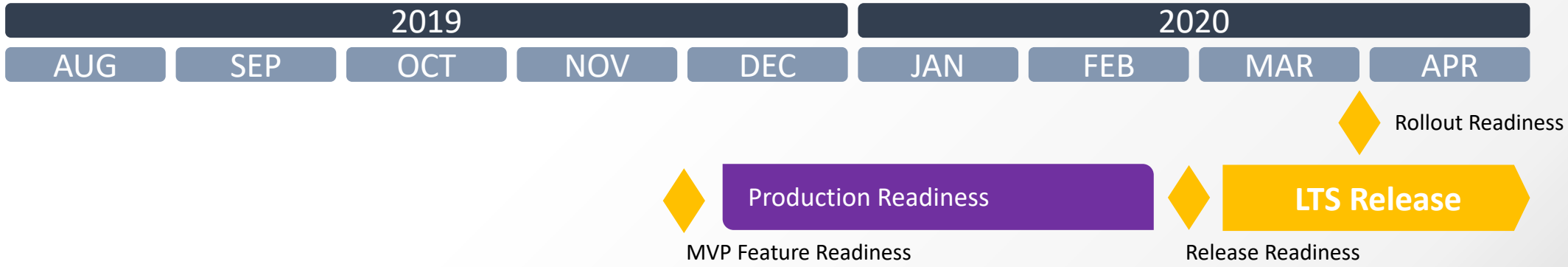
- Once vendor forks surface things may not get merged back

It might become many vendor specific forks

- Vendor competition may be good
- Vendors may make vendor specific “improvements”
- Choosing a single vendor fork is essentially vendor lock in

Migration (actual)

VOLTHA STABILIZATION EFFORT ROADMAP



Community

A4 Team

Stabilization Brigade

VOLTHA Master Branch

Focus on MVP Feature Development: FTTH / FTTB

Ongoing other Feature and Stabilization Development

Stabilization patches cherry-picked into Master branch

Stabilization Increment | Stabilization Increment | Stabilization Increment | Stabilization Increment | Stabilization Increment

VOLTHA Stabilization Branch

Focus Groups

Scope of Work for Stabilization - backlog compilation session - known issues list, any refactoring requirements, NFR benchmarks, et. al.

Automated Testing Framework – developing our testing approach, framework and tooling choices, definition of done et. al.

Code Commit Standards – defining target code commit practices and standards of operation

Status

The Good

- Stabilization Brigade Created
- Renewed focus on automated testing
- Create stabilization branch (voltha-2.1)

The Bad

- Bug fixes being checked in w/o tests
- Continued new feature development on master

The Ugly

- Cherry pick all bug fixes between master and voltha-2.1 branches

Takeaway

Einstein's Insanity Definition*

Everything is permissible, but not everything is beneficial

We need to stabilize VOLTHA if we ever expect to get to production

A simplified VOLTHA is a viable / workable solution

* I realize that Einstein may not actually be the origin of this quote / definition, but that isn't really important for this presentation.

Mèsi