Rethinking Debian release

18th November 2017

Hideki Yamane <henrich@debian.org>

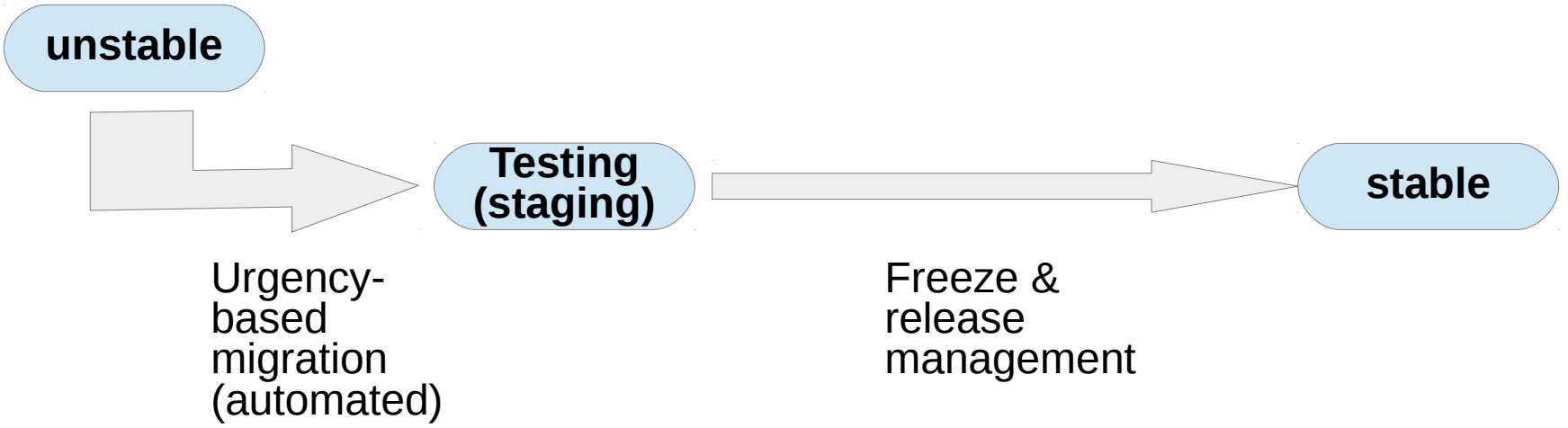
Why rethink release?

- “Debian is old”
- “Debian (stable) is old”
- “Debian (stable) is old (for our usage)”

- “Some packages are still old and buggy, but no update”

Current Release Migration

- Unstable → Testing
  - Based on urgency (high,medium,low)
    - Blocked by Release Critical bug
- Testing → Stable
  - (Looong) Freeze and release



Stable release management = many-legged-race

- Push release management causes “many-legged-race”

http://web-japan.org/kidsweb/archives/life/action/06-02/act0602.html

Its result...

- Like this...
  https://www.youtube.com/watch?v=Rwqvx99Gz2U

Another Problem:
Is that package tested, **really?**

- Who tests it?
  - Sometimes "Passive test" doesn't work well
- Code **never** matures
  - Code != Wine / Whiskey
  - But time makes features to rot...

Worst scenario

- Upload to unstable
  → no one cares it
  → no bugs filed
  → migrate to testing
  → release stable
  → found bugs in stable, but leave it...
  (since put not tiny changes to stable is not easy task...)
  → bad user experience
  → bad reputation
  → less user
  → less developer...

Answer (1): + “Active” migration

- Same as other distros
  - Gentoo: mask (package flag)
  - Fedora: bohdi (voting system)
  - openSUSE: openQA (automated test)
- "pull" migration system via vote by users & maintainers
  - “Package quality” is guaranteed by safety harness (pipeline)
- It ensure "it works" by someone, at least

Pull is better than push

“Push” Testing to stable migration

- Thousands changes in one time
- Handled by few release managers
  = capacity overflow
  → burnout...

vs

“Pull” migration

- Several changes in one time
- Handled by hundreds advanced users & maintainers

Answer (2): New distribution

Why we need "new distribution"?

- Average users **never** use unstable or testing, they use "released" one (= stable)

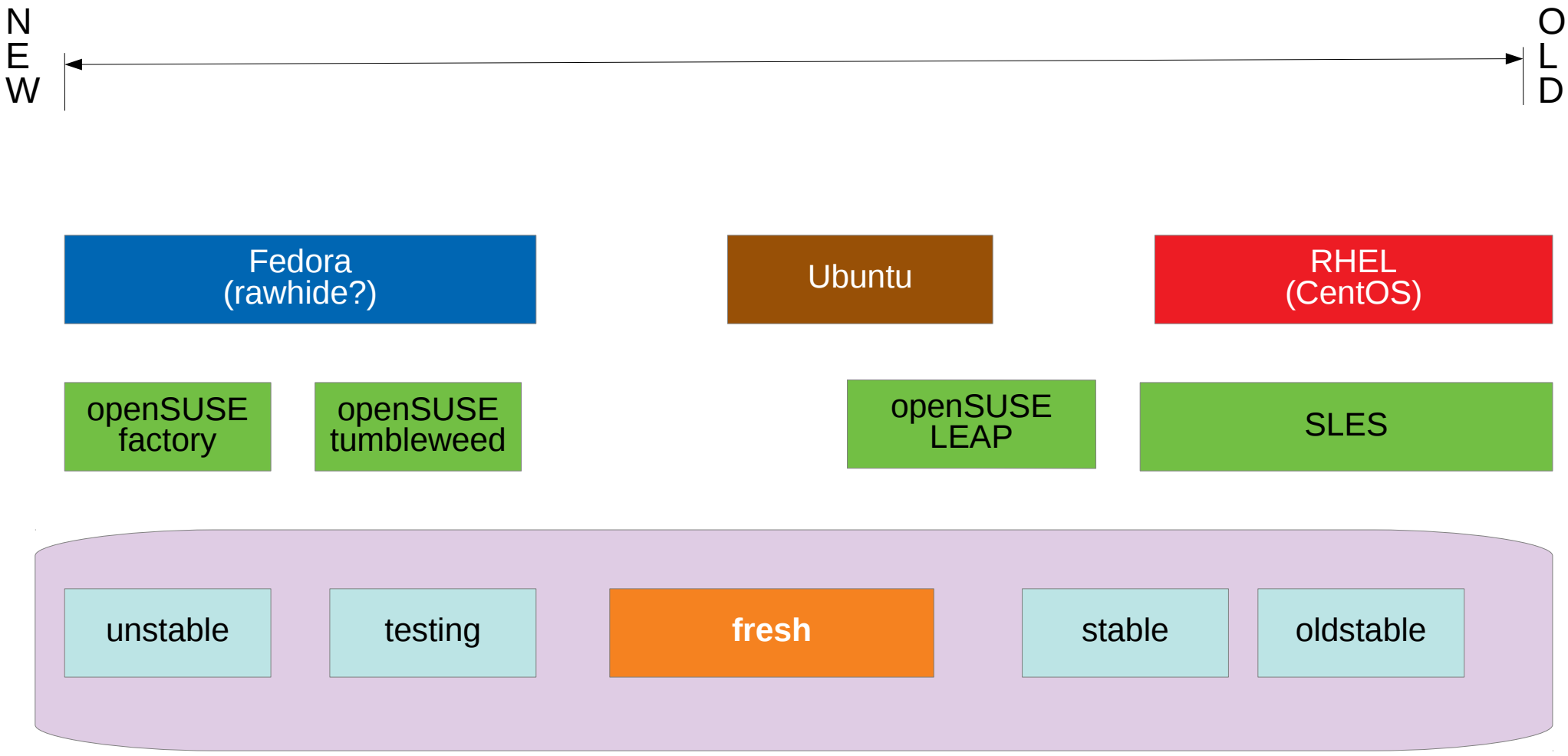
- “Innovators theory” (by Everett M. Rogers)
  - Innovators : 2.5 % (unstable)
  - Early Adopters : 13.5 % (testing)
  - Early Majority : 34.0 %
  - Late Majority : 34.0 % (stable)
  - Laggards : 16.0 % (oldstable)

“**Fresh**” distribution

Innovators	:	2.5 %	(unstable)
Early Adopters	:	13.5 %	(testing)
Early Majority	:	34.0 %	**“Fresh”**
Late Majority	:	34.0 %	(stable)
Laggards	:	16.0 %	(oldstable)

We can get more users! (100 / 66 = 150%)

Positioning



Fresh?

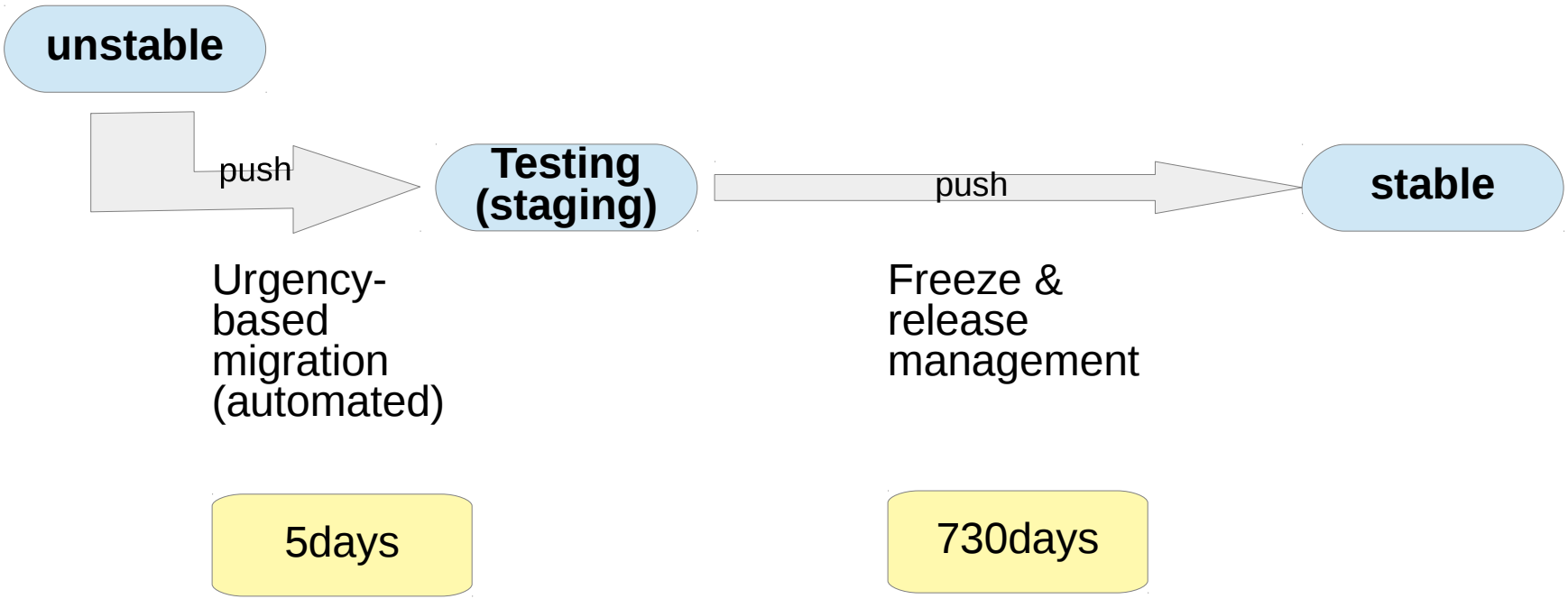
- Borrow name from LibO :)
- Target: Average users (Early Majority)
- **Release** every one or two week
  - Rolling release
  - Predictable scheduled release
- Pull change sets
  - Sustainable deploy
  - Ensure changes, not break anything

Not push changes into stable directly

- Why new “fresh” distribution?
  - Users expect stable as stable (≒ not changed so much)
  - We afraid to break stable release

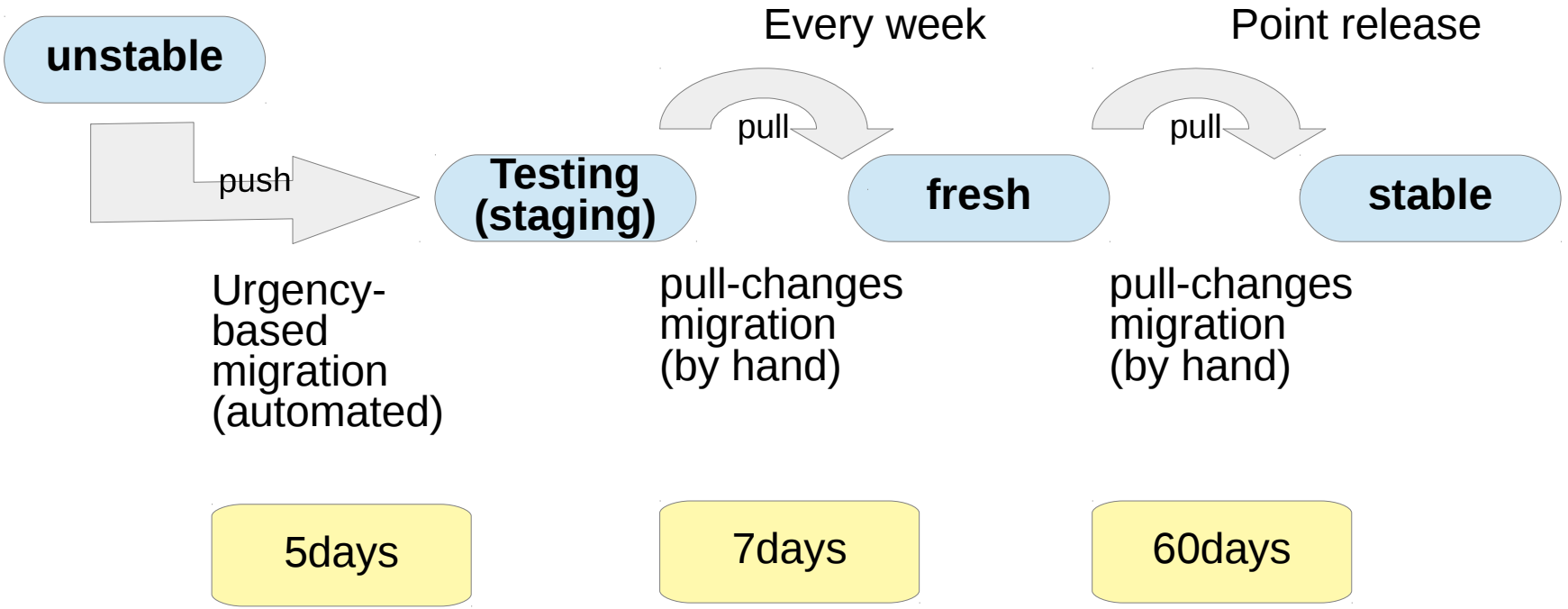
Migration cycle time

Traditional migration



Migration cycle time

Add “Fresh” distribution



735days → 12days (5+7) = **60times faster!**

Shorten cycle time

- Before : 730 days (minimum 180days)
- After : 12 days

  15-60 times faster delivery!

- Maximize added-value

Change the rule!

- There was a reason to make rules
  - Unstable – Testing – Stable
  - Long freeze term and release
- But situation has changed, then rules should be changed, too. Because its rule becomes bottleneck

Faster release introduce more bugs?

- Q: It may introduce more bugs!
  - A: “test early and fail fast” on fresh stage, but less bugs in stable since more test users watch it.
  - Testers
    - Previous : 2.5 + 13.5 = 16.0
    - Fresh : 2.5 + 13.5 + 34.0 = 50.0 → **300%**

“Fresh”: Pros & Cons

- Pros)
  - 150% users, 300% testers
  - 60 times faster release
  - Same cadence, its release date & changes are predictable
  - Changes in each release are small, users can bite it (No Big Bang release)
    - Less freeze term for next release
    - Not need to hassle to make huge release note
    - Moe "real acceptance test" by real users for next stable release
- Cons)
  - It just costs
    - Infrastructure change
    - Docs & website update
    - More release manager & publicity work
    - Prepare security fix (but delta with unstable is small, right?)
      - maybe it reduce backport effort in stable

Metrics?

- More testers
  - BTS number
  - RC in stable / bugs in stable
- More users
  - Download number