

└ Fallout



- promote collaboration and co-development between FOSS projects
- traditional feedback cycle for how changes, and in particular, regressions become visible to consumers
- each of these steps takes days (frequent releases with rawhide) days to months (rare releases, LTS)
- demotivating and inefficient: when you get bug report, all knowledge and context are gone
- (fallout) cockpit project: top of the dependency tree, find lots of regressions in all OSes
- elaborate system to track these; oldest from 2018, 10 have been open since 2021 or earlier
- long cycle doesn't work: the damage is done, nuisance

└ Approach

Approach



```
consumer packit.yaml:  
- job: copr_build  
  trigger: commit  
  owner: "Upstream"  
  projects: "main-builder"
```



- tighten feedback to “immediate”, “shift left” in fancy parlance
- (2.) catch regressions *before* they land and affect users
- consumer’s tests run in PR, result in minutes/hours, current state in mind,
- (3.) consumer sees change, no bisecting distro, land w/ confidence
- (4.) build better relationship instead of yelling
- (Approach) build with public infra: copr, TF, packit
- consumer: (some) tests run through tmt (Fedora/RHEL gating)
- consumer: build COPR from main branch (cf. PR packit COPRs)
- your project: tell packit.yaml to run consumer test plan (bit technical, not shown here)
- see the result in PRs (local: your own, cockpit: consumer)

Upstream cross-project testing

└ Experience

Experience

- mostly social effort, not technical
- first [podman regression](#) @19888
- Cockpit tests run in podman, udisk2, and selinux-policy



- podman used to be one of our main regression sources, first enabled
- only after a few days: caught our first regression; collaborated in PR
- process works reasonably well, better working relationship
- since then we enabled it in some more projects
- be aware: opt-in, discuss, work together, and establish rules (response time)
- automatic notification of consumer proj devs in PR
- doesn't scale well, but tremendously helps with key dependencies