How to push your testing upstream

Sam Thursfield

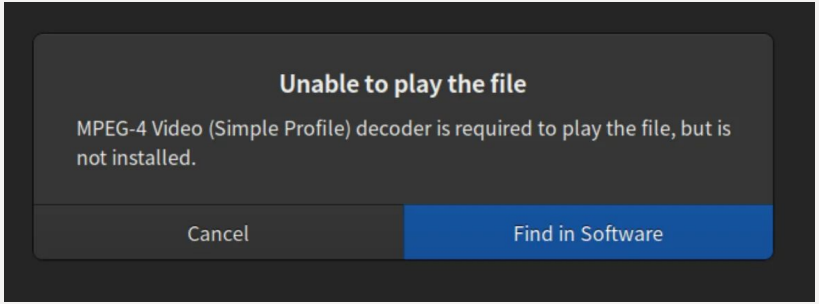
Ladies and gentlemen...

I have a **vision of the future.**

We have come a long way since 1998

End to end testing

- **Test the entire system**: kernel, systemd, userspace, graphics stack.
- **Navigate like a user**: keyboard & mouse events, visual scanning.



Quiz

- Which distros are doing this today?

Downstream testing with openQA

So far we know about:

- Fedora https://openqa.fedoraproject.org/ ,
- AlmaLinux http://openqa.almalinux.org/,
- Debian https://openqa.debian.net/ ,
- https://openqa.qubes-os.org/ ,
- https://openqa.endlessm.com/ ,
- the GNOME project https://openqa.gnome.org/, https://www.codethink.co.uk/articles/2021/automated-linux-kernel-testing/, https://en.euro-linux.com/blog/openqa-or-how-we-test-eurolinux/,
- openSUSE KDE contributors (with their own workflows, https://openqa.opensuse.org/group_overview/23),
- openSUSE GNOME contributors (https://openqa.opensuse.org/group_overview/35),
- OBS developers (https://openqa.opensuse.org/parent_group_overview/7#grouped_by_build) ,
- wicked developers (https://gitlab.suse.de/wicked-maintainers/wicked-ci#openqa),
- and of course our team itself for "openQA-in-openQA Tests" :) https://openqa.opensuse.org/group_overview/24 .
- Also see https://en.opensuse.org/openSUSE:OpenQA/Partners

Quiz

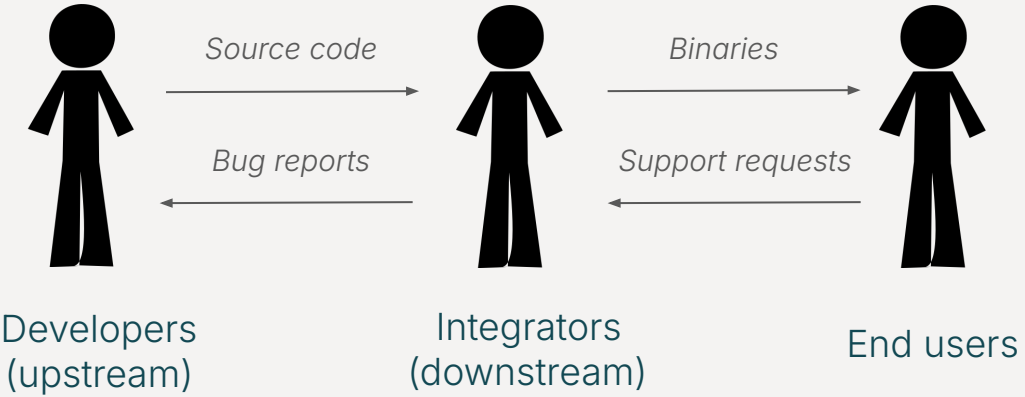
- Which distros are doing this today?

Github: linux-qa/linux-qa/issues/1

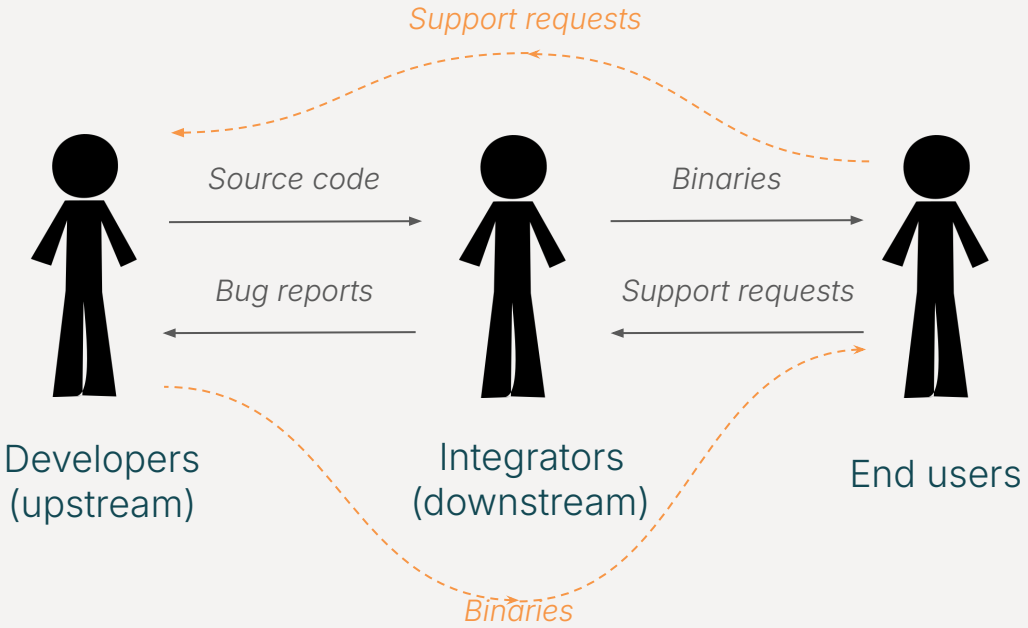
openSUSE wiki: openSUSE:OpenQA/Partners#GNOME_&_KDE

Another quiz

Has a desktop or app developer ever you helped you write and maintain **end-to-end tests** for their software?



Developers want fast feedback on changes



Automated QA upstream, for GNOME

Set up by a Codethink R&D project in 2021.

Extended during Outreachy internship in 2023.

Maintained by volunteers... at times.

Challenges

- GNOME is **distro-independent**.

  Paid contributors from: Canonical, Endless, openSUSE, Red Hat, ...

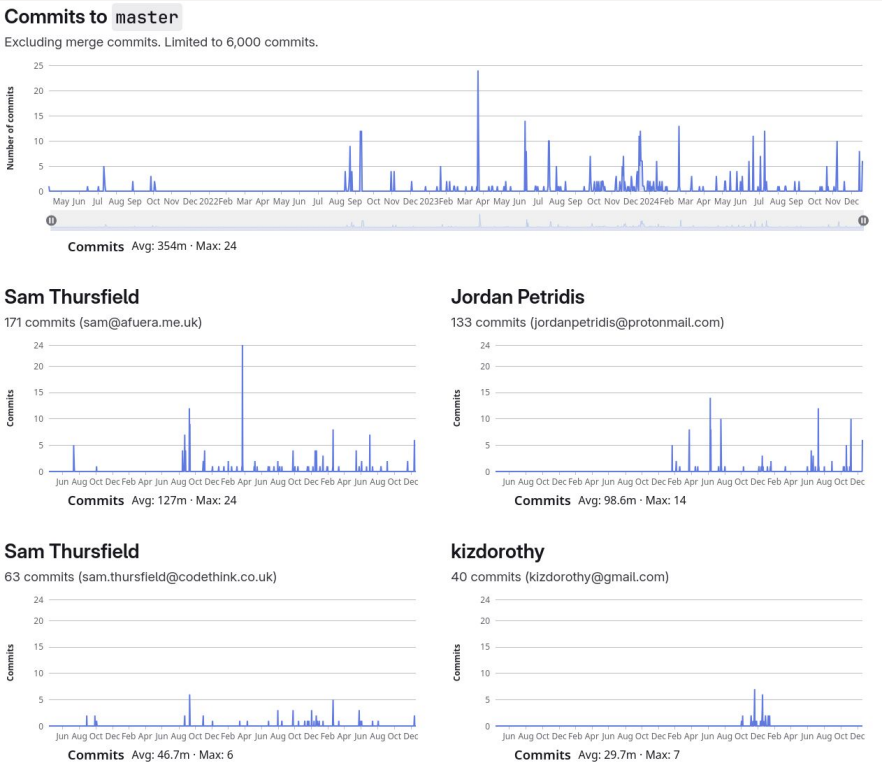
- Limited funding for **infrastructure.**

- **OS images** are big (around 4GB).

Automated QA upstream, for GNOME

- **openQA** web UI and database set up at https://openqa.gnome.org.
- Five **test suites** defined in GNOME/openqa-tests
- Glue code to integrate openQA testing into **Gitlab CI** pipelines.
- Every commit to **GNOME/gnome-build-meta** `master` branch runs the testsuite.

What happens next will astonish you



So far, openQA is largely invisible in GNOME.

Module maintainers tell me:

> *"I want to see the QA test results on my project's open merge requests, when I'm doing code review".*

We need to meet upstream maintainers where *they* are.

Codethink

Let’s put tests in module repos, then.

Prototype: openQA tests for GNOME Shell

Part of STF + Codethink funded project during 2024, focused on GNOME OS + systemd-sysupdate.

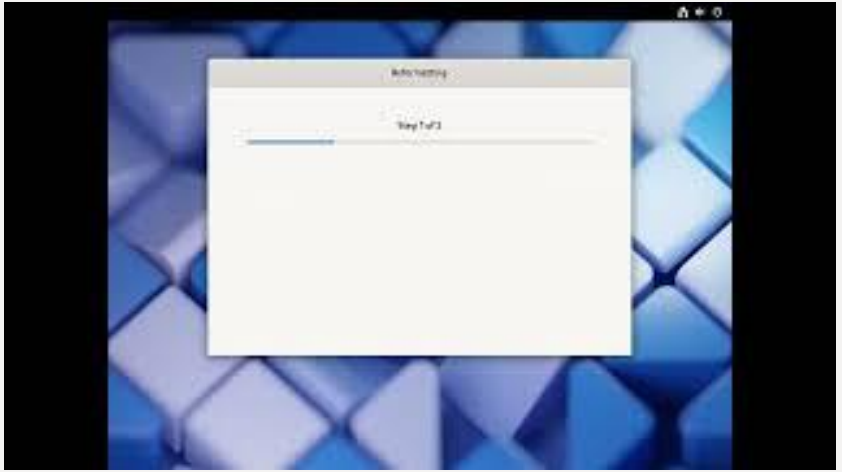
GNOME Shell is **one of the most difficult GNOME modules to develop and test** – core apps should be easier.

Challenges

- Infrastructure is still limited.
- Building a full OS image is **slow** and **expensive.**
- OS install takes **several minutes** per test.
- Lots of **moving parts** to coordinate
- Creating openQA tests is a **new skill**

Prototype: openQA tests for GNOME Shell

Installing GNOME OS into a VM from the ISO looks like this...



1. Build the image
   45min
2. Download the image onto the runner
   10min
3. Run the installer in a VM, then reboot
   2min
4. Run GNOME Initial Setup to create a user
   20 sec

Now you can actually test applications!

Prototype: openQA tests for GNOME Shell

Installing GNOME OS into a VM from the ISO looks like this...

1. ~~Build the image~~

   Build a **systemd-sysext binary** and overlay it on an existing image.

2. Download the image onto the runner
   10min
3. ~~Run the installer in a VM, then reboot~~

   Use a preinstalled **hard disk image**.

4. ~~Run GNOME Initial Setup to create a user~~

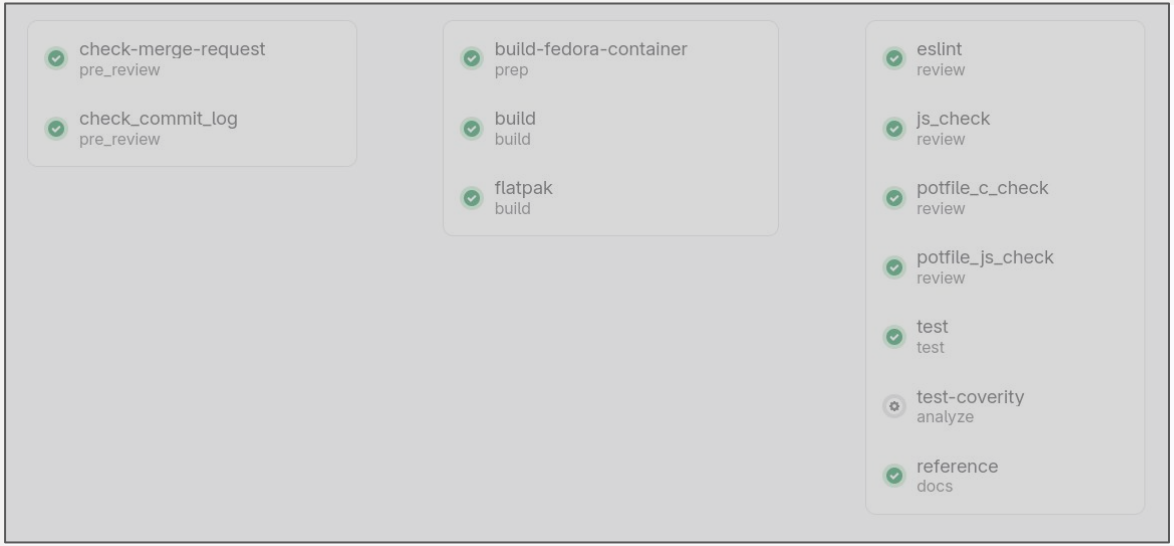
   Configure systemd via SMBIOS to **pre-create a user** and bypass initial setup.

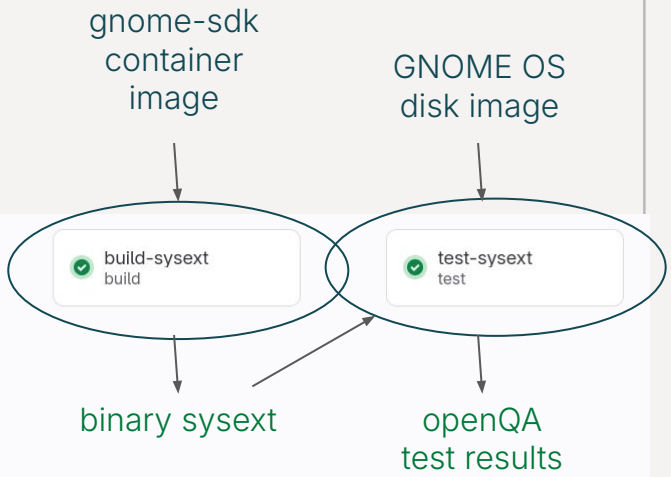
   (Patch: https://github.com/systemd/systemd/pull/33811)

Prototype: openQA tests for GNOME Shell

Status:

- Building sysexts: landed in gnome-shell
  GNOME/gnome-shell !3390 “ci: Build and publish system extensions”
- Testing the sysext in GNOME OS: not ready to land.
  GNOME/gnome-shell !3419 “Run minimal openQA tests with the system extension”





Prototype: openQA tests for GNOME Shell

Great results... for a prototype

- Tests are **fairly fast!**
  - Less than 10 minutes
- Tests are **kind of reproducible!**
  - We control entire software stack.
  - VM performance varies due to infrastructure limitations.
- About 100 lines of **scaffolding code.**

Is it ready for prime time?

- Still runs *gnome-initial-setup*, need to finish systemd dropin user records PR.
- Image downloads are slow.
- Gitlab runners are overloaded, and unpredictable.
- Upstream maintainers don't know how to write openQA tests.

Where do we go from here?

Where do we go from here?

Investment needed

- Finish the systemd dropin user records patch
- Build a shared base test for Linux +systemd
- Dedicated QA infrastructure for upstreams

Thought needed

- Make openQA more accessible for newcomers
- Sharing tests between openQA instances

Shared base for Linux QA testing

The os-autoinst basetest is **completely generic** – it can be used for Linux, Windows, Mac OS and any other OS.

This means, every Linux distro testsuite has **custom Perl code** to bring up a Linux machine:

- Connect to serial console
- Wait for boot and login over serial
- Switch between framebuffer consoles
- Upload journal on failure

... and **everybody does it differently**, of course.

Goal: ***a new "basetest-linux-systemd" module.***
https://github.com/linux-qa/linux-qa/issues/8

```
sub login {
    die 'Login expects two arguments' unless @_ == 2;
    my $user = shift;
    my $prompt = shift;
    my $escseq = qr/(\e [\(\[] [\d\w]{1,2})/x;

    # Eat stale buffer contents, otherwise the code below
    # after reboot and start typing the username before th
    # ready to accept it
    wait_serial(qr/login:\s*$/i, timeout => 3, quiet => 1)
    # newline nudges the guest to display the login prompt
    # changes then remove it
    send_key 'ret';
    die 'Failed to wait for login prompt' unless wait_seri
    enter_cmd("$user");

    my $re = qr/$user/i;
    if (!wait_serial($re, timeout => 3)) {
        record_info('RELOGIN', 'Need to retry login to wor
        enter_cmd("$user");
        die 'Failed to wait for password prompt' unless wa
    }
```

Dedicated infrastructure

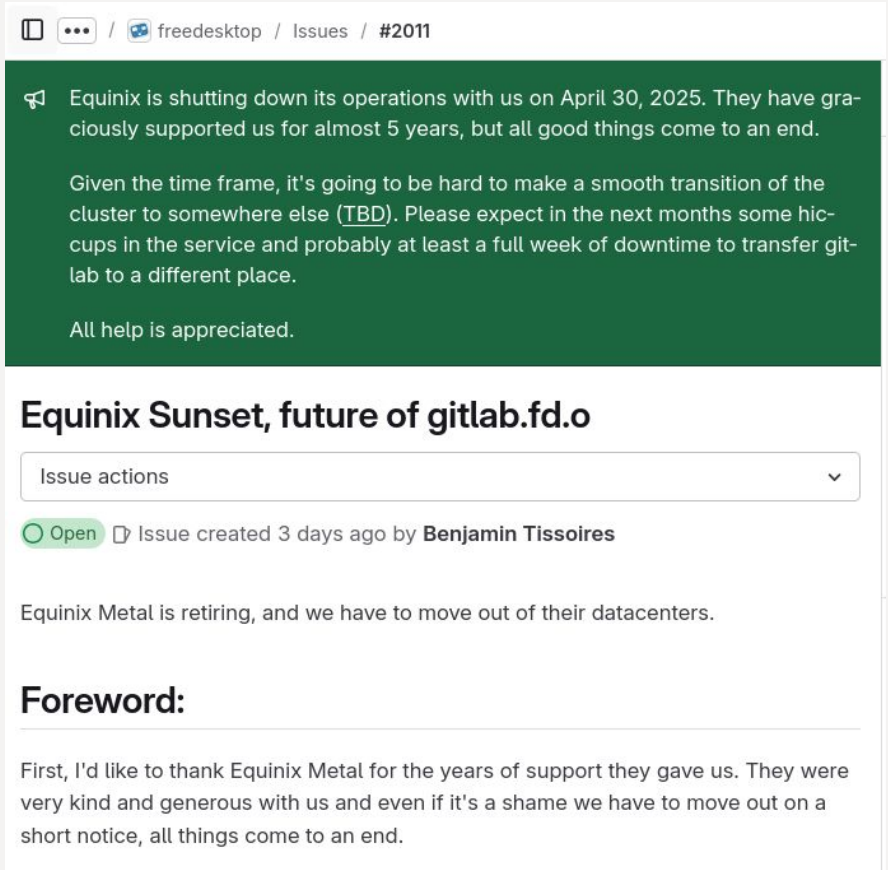
GNOME has a small infra team (2 × 0.5 people).

CI hardware is provided by a few sponsors:

- AWS open source credits
- Canonical
- David Heidelberg
- ~~Equinix~~
- GIMP
- OSU Open Source Lab
- Red Hat

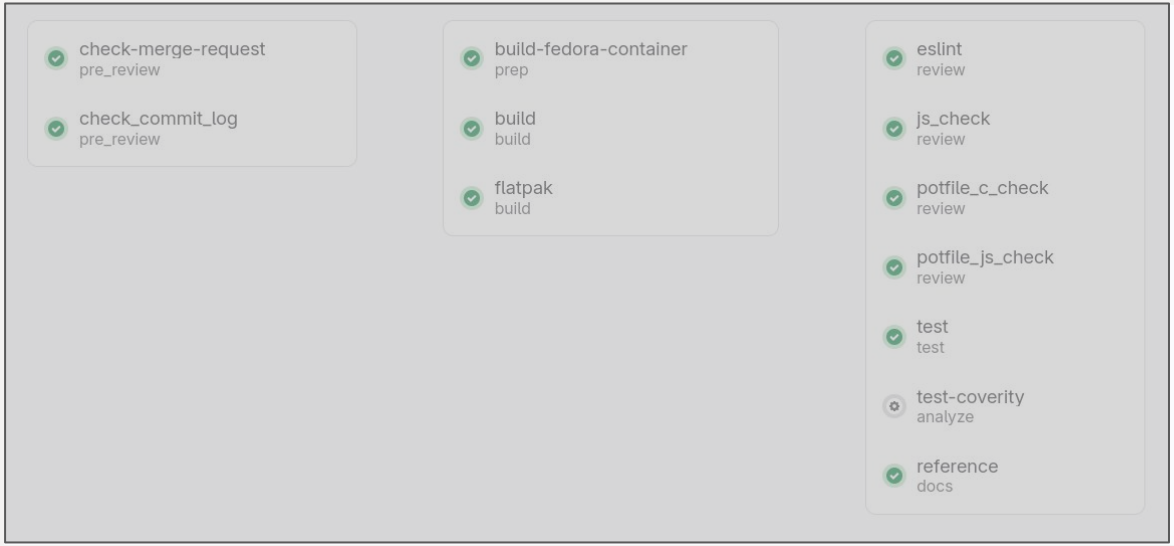
Goals:

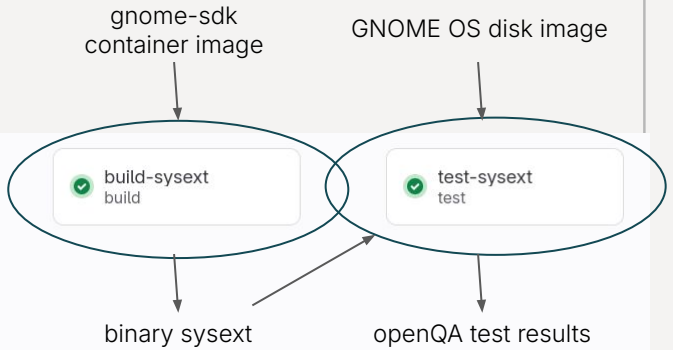
- ***Funded QA infrastructure for upstream projects***
- ***Hardware test labs.***



More hardware... more sysexts

With enough hardware, we could build sysexts for multiple OSes: GNOME OS, Debian unstable, Fedora rawhide, openSUSE Tumbleweed...



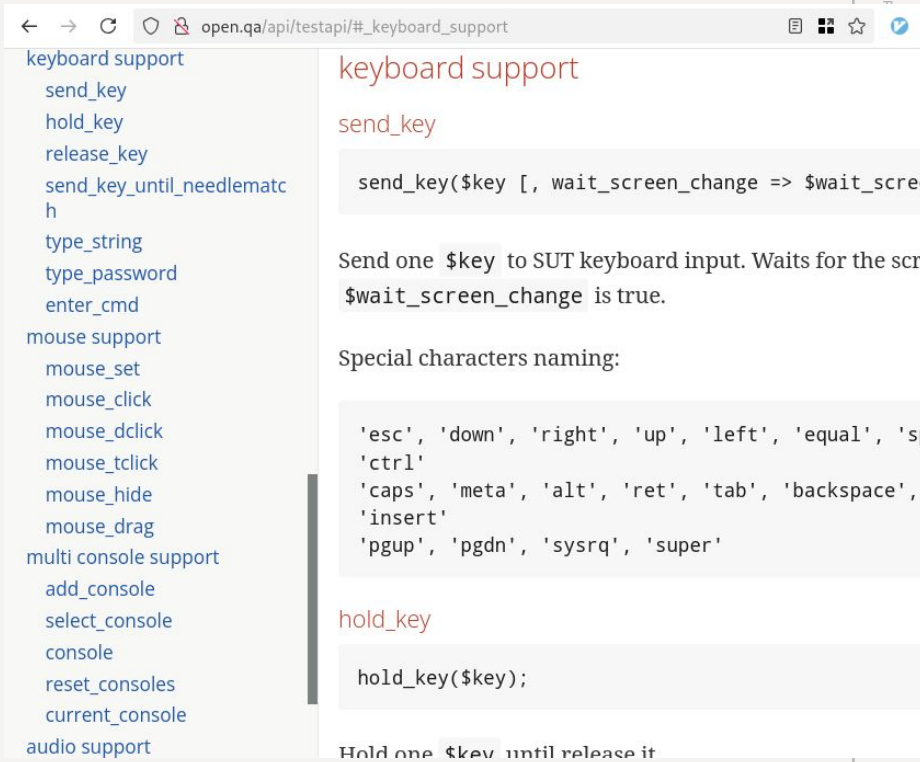


Making openQA accessible for newcomers

openQA's **testapi DSL** is ❤ fantastic ❤!

But:

- Tests are arbitrary Perl programs
- os-autoinst-distri-opensuse *lib/* dir has *42,000 SLOC*.
- The recommended way to develop tests is to deploy an openQA web UI on your laptop.

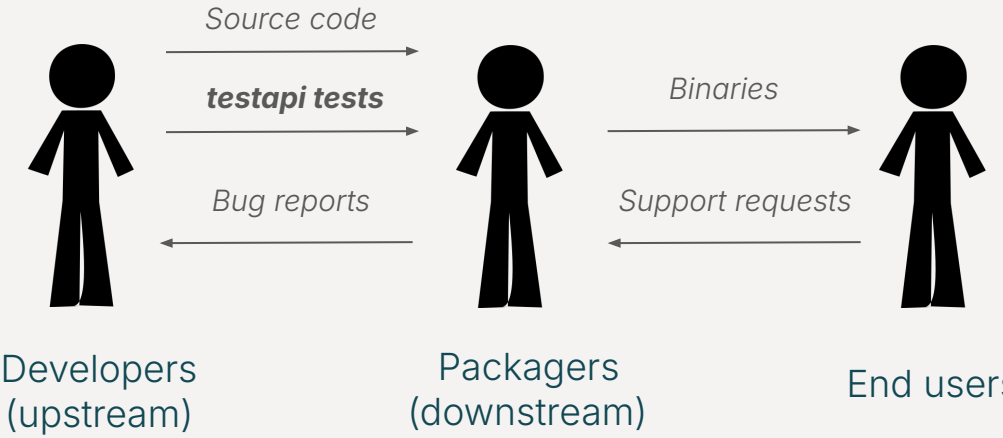


Sharing tests between instances

If we all use the *basetest-linux-systemd* class and we publish our helper libraries somewhere like CPAN...

Then:

- Upstream GNOME developers could maintain openQA tests which run against GNOME OS
- Downstreams could run *the same tests* against their own distros.



This is ambitious!

We aren’t going to get to this point in 2025.

But we can begin our journey and map the way there.

Let’s tackle the **hardest problem in software development**...

Getting different groups to **talk to each other** and agree on things.

Thanks for watching

Join our monthly call!

Second Thursday of every month.

- Odd number months: Europe morning, Asia evening
- Even number months: Europe evening, US morning.

Next call: **Thursday 6th Feb, 16:00 UTC.**

Details here: https://github.com/linux-qa/linux-qa

Contact me:
sthursfield@gnome.org
sam.thursfield@codethink.co.uk