# Test Driven Development in Debian — BoF

Ian Jackson Tom Marble Stefano Zacchiroli

Debian

26 July 2011
DebConf11
Banja Luka, Republic of Srpska
Bosnia and Herzegovina

# On gobby

```
# apt-get install gobby-infinote
$ gobby-0.5 -c gobby.debian.net
+ gobby.debian.net
  + debconf11
    + dc11-TDDD
```

# Disclaimer

1. more testing in Debian
   - per-package testing
   - distro-wide testing

2. test driven development in Debian

# Testing in Debian — state of the art

- building (FTBFS)
- policy compliance (lintian, piuparts, automated REJECTs)
- build time testing (upstream test suite, sometimes. . . )
- integration testing (`http://edos.debian.net`)
- [ automated code analysis (e.g. DACA) ]
- . . .

# Do we need more?

# Do we need more?

Yes.

- quality!
- reduce packaging contribution barrier
  - esp. when working on packages of others (e.g. NMU)
- ease testing of archive-wide changes (e.g. in conjunction with PPA)

# Doing more — system testing

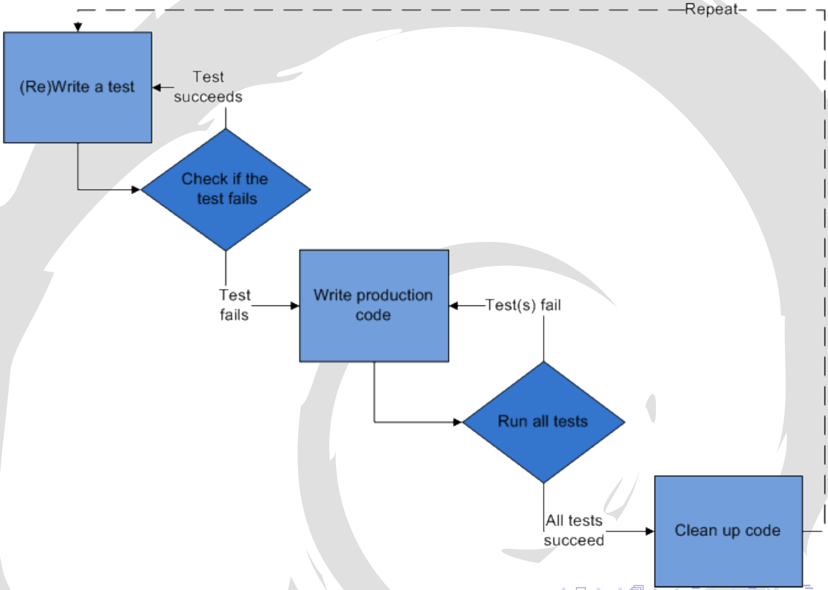
Goal: making sure a base Debian will always work

- Build an image (e.g. vmdebootstrap + tasksel + custom configs)
- Boot image (kvm / virsh)
- Run tests relevant to the scenario (systest)
- Possibly upgrade system (stable to testing, then reboot and re-run tests)
- Report results
- Take action (publish images, access to live images (VNC?), allow developers/upstream to replicate bug and then confirm the fix)

# More testing — Test Driven Development

- TDD for Debian (TDDD) is about applying the TDD principles in the context of keeping a running Debian system functionally correct.
- We need the courage and confidence to refactor code and make significant system changes by being sure that our changes do not break anything.
- This can help Debian evolve *faster*

# More testing — Test Driven Development (cont.)



# Doing more — autopkgtest

- DEP8: autopkgtest — automatic as-installed package testing
- establish a standard interface to define and run “as-installed” tests of packages, i.e. the testing of packages in a context as close as possible to a Debian system where the packages subject to testing are properly installed

- `http://dep.debian.net/deps/dep8/`

# autopkgtest — some details

- README.package-tests provides a standard format to declare per-package tests using the new debian/tests/control file. Tests come as executable files which will be run by the adt-run tool in a testbed where the package(s) to be tested is already installed
- README.virtualisation-sever describes the interface among the test runner and the testbed
- Ian brought it back to life!

# Questions

- what can we do to improve package testing in Debian?
- which archive-wide packaging testing initiatives are you aware of in Debian?
- how do you test your packages?
- do you have packages with (extensive) upstream test suites?
- if yes, do you run those test suites at package build-time?
- if not, why not?
  - e.g. they are not flexible enough to be run at build-time
- are you aware of testing-related initiatives that other distribution are doing and that Debian can adopt?
- how can we leverage Debian packaging testing to benefit upstreams?
  - e.g. we have way more architectures than avg. upstream
- how can we collect system tests collaboratively?

about the slides:

| | |
|---|---|
| available at | `https://gitorious.org/zacchiro/talks/trees/master/2011/201107-dc11-tddd` |
| copyright © 2010–2011 | Stefano Zacchiroli |
| license | CC BY-SA 3.0 — Creative Commons Attribution-ShareAlike 3.0 |