Greetings from SlackBuilds.org

David Spencer
pkgsrcCon 2017
About SBo

11 years old
conventional ports-inspired setup

- from source ftw
- shell script + metadata

~6500 packages
disjoint from core Slackware (~1400 packages)

lightweight project

one new server, one old server
About SBo

~250 maintainers active in last year
~12500 commits in last year

no bugtracker
no CI

- Infrastructure is a productivity killer
- Aggressively reductionist on dep management
- Vanilla from upstream, patch only when needed
- Don’t split packages
- git git baby
About SBo

submissions are open
ambition to submit ‘something’ is a thing
maintainers drop in and drop out

review must be sympathetic
volunteers are a pipeline not a funnel
don’t crush people’s dreams

maintainer is expert on the software
reviewer (admin) is expert on good packaging
no room for style variations
About SBo

Education needs to be a thing
No time in review
feedback hurts, doesn’t scale
Currently done on mailing list & forum
  • CI as education
Listening

systemd refugees

rolling release

• stable versus current

out of date / security / unmaintained upstream
disappearing SBo maintainers
disappearing sources and projects

• repology

• keeps mailing list active
Happy community

Users helping each other

Tools

Satellite projects

Package all the obscure things

- if it exists it will attract users
Unopened letter to the world

Need to educate upstreams
proper releases with proper tarballs
don’t move or delete old tarballs
learn to write a decent Makefile
no, I don’t want your stinking CFLAGS
don’t use -Werror

the world isn’t Ubuntu, plz /usr/lib64 & /usr/man
no, we won’t stick with gcc-4.9 just for you

Biggest sewer - development/
Signs of the apocalypse

npm cargo pip bower gem cpan composer cabal
xmkmf qmake autotools cmake ninja scons jam
waf meson flatpak snappy bubblewrap docker
casync

- How many of these are you going to learn?
- Will they interwork?
- Where’s your support going to come from?
- Are you really going to keep them updated?

‘A new version is available!’ meme
What exactly do we do again?

Packaging is a filter, in the classic UNIX sense

- **Input**: arbitrary upstream software
- **Processing**: apply curated standards
  - interoperability
  - predictability
- **Output**: useful, usable and in use

None of the new paradigms do this
Let’s get together

Share microfixes

- discoverable repositories (cgit or gtfo)

Educate upstream developers

- your project isn’t worthless
  your project isn’t special

Educate end users

- npm is not Cyber Jesus