

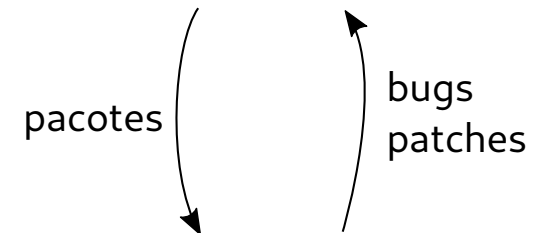
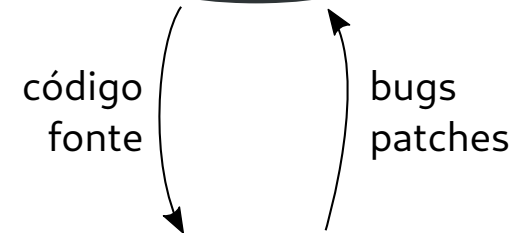
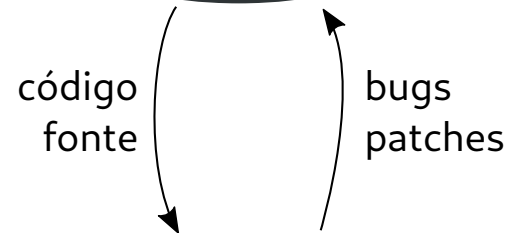
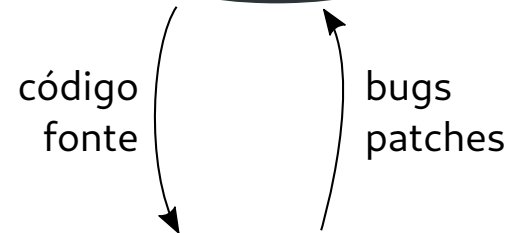
Why and how to (not) get your project  
packaged by distributions

**Antonio Terceiro**

terceiro@debian.org

This talk is based on my experience as upstream developer and Debian package maintainer, and on documentation maintained by Debian contributors

How a distribution works



`$ apt install foo`

Why would you want to have your project packaged?

<https://wiki.debian.org/AdvantagesForUpstream>

Easier for individual users to install

Easier for corporate environments

Trusted repository



Distro QA

Integration with the wider ecosystem

Free (as in beer) review

# Derivatives

Dependency tree is maintained

Multiple architectures

When NOT being packaged makes sense

Server-side applications (maybe)



Immature/fast-moving ecosystem

Constant addition of dependencies

How to make it easier for your  
project to be packaged

(a.k.a. How to be a good upstream)

<https://wiki.debian.org/UpstreamGuide>

Provide source distributions

Do not include convenience copies  
of third party code

zlib  
pcre  
sqlite  
poppler

Clearly specify a License

# Releases and Versions



Documentation

Provide Manual pages

Do not hardcode paths

Handle user home directories  
correctly (i.e. support XDG)

Mind your dependencies

Include a test suite

Have a method to handle security issues

Use a public Bug tracking system



Provide a public Version Control System  
(git)

Use a standard Build System

Reproducible Builds™

What about using containers?  
(docker, flatpak, snap)

## When using containers:

- You still need to use a base distribution for your application
- You can bundle any dependencies together with the application, but then you assume the work to keep them up to date

## Managing application dependencies **using a language-specific dependency management** (bundler, pip, etc)

- get new features at your convenience
- lock all versions on all packages, and miss relevant upgrades; or
- don't lock versions and live with upgrades at arbitrary points in time

## Managing application dependencies **using distribution packages**

- develop against a well-known platform with upgrades at well-defined points in time
- get security support for free
- live without new features until the next distribution upgrade





Thanks!

Contact:

[terceiro@debian.org](mailto:terceiro@debian.org)