

POZNAŃSKA IMPREZA
W O L N E G O
OPROGRAMOWANIA

Package lifecycle

in the Fedora Linux distribution

Artur Iwicki
2018-04-28



slides available under the CC-BY-SA 4.0 licence

What are these "packages" about?

A very popular idea in Linux distributions is the concept of the package manager.

Instead of, as in other operating systems, having to look for software on shady sites on the internet, the distribution contains its own set of software that we can install programs from.

Software is distributed in the form of so-called packages.

Packaging is done not by the program authors, but by members of the distro's community.

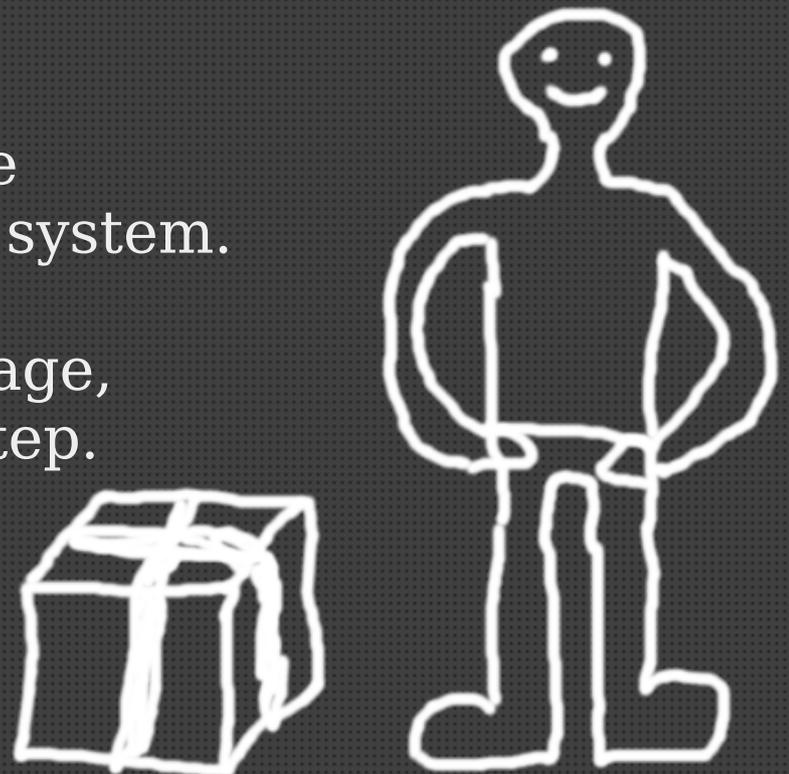


Step 1: packaging

The first step on the road to getting a package into the official repository is, obviously, packaging the software.

The person interested tries to package the software and build the package on their own system.

Once they have a working package, they can continue to the next step.



Step 2: package review

The next step is to file a ticket in Bugzilla, requesting a package review.

Reviews are a community process, which means there aren't any people strictly responsible for reviews; they are done by volunteers.

The author of the package should fix any issues noticed by the reviewers.



Step 2: package review

A review usually ends in one of the four following ways:



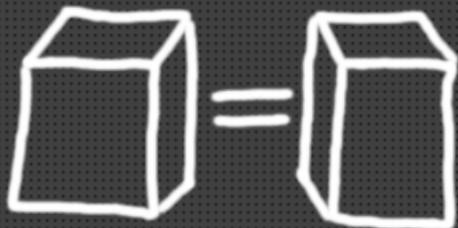
Approval:

the package has been approved by the reviewer.



Rejection: the package was deemed inappropriate.

Usually this is caused by legal issues, such as licensing, software patents, or trademark / IP issues.



Duplicate: this software has already been packaged and submitted for review by someone else.



Abandonment: the person who opened the review request stopped responding to comments.

Step 2.5: looking for a sponsor

Our package has received a green light!
Now it's time to use our *Package Maintainer* powers...
unless we haven't been made one yet.

New packagers must also look for a sponsor
who'll be willing to back up the person's request
to be made a package maintainer.

Being a *Sponsor* is a separate role
in the packager group;
as such, this is not about being
backed up by any other packager.



Step 2.5: looking for a sponsor

To become sponsored, we should demonstrate a good knowledge of the packaging guidelines and the community code of conduct.

This is usually done by reviewing a couple other packages from the Review Request list.

Once we find a *sponsor* willing to back us up, all that's left to do is just wait for them to file our application and hope it gets accepted.



Step 3: creating and importing into a repository

After getting our package approved, it's time to send a request to create a *dist-git* repository for it.

Holding the package specifications inside *git* repositories allows several co-maintainers to easily collaborate, and also provides a mechanism to use different specs for different distro releases (via *git* branches).

To save space and bandwidth, package (upstream) sources are not stored inside the *git* repositories, but rather using a helper service.

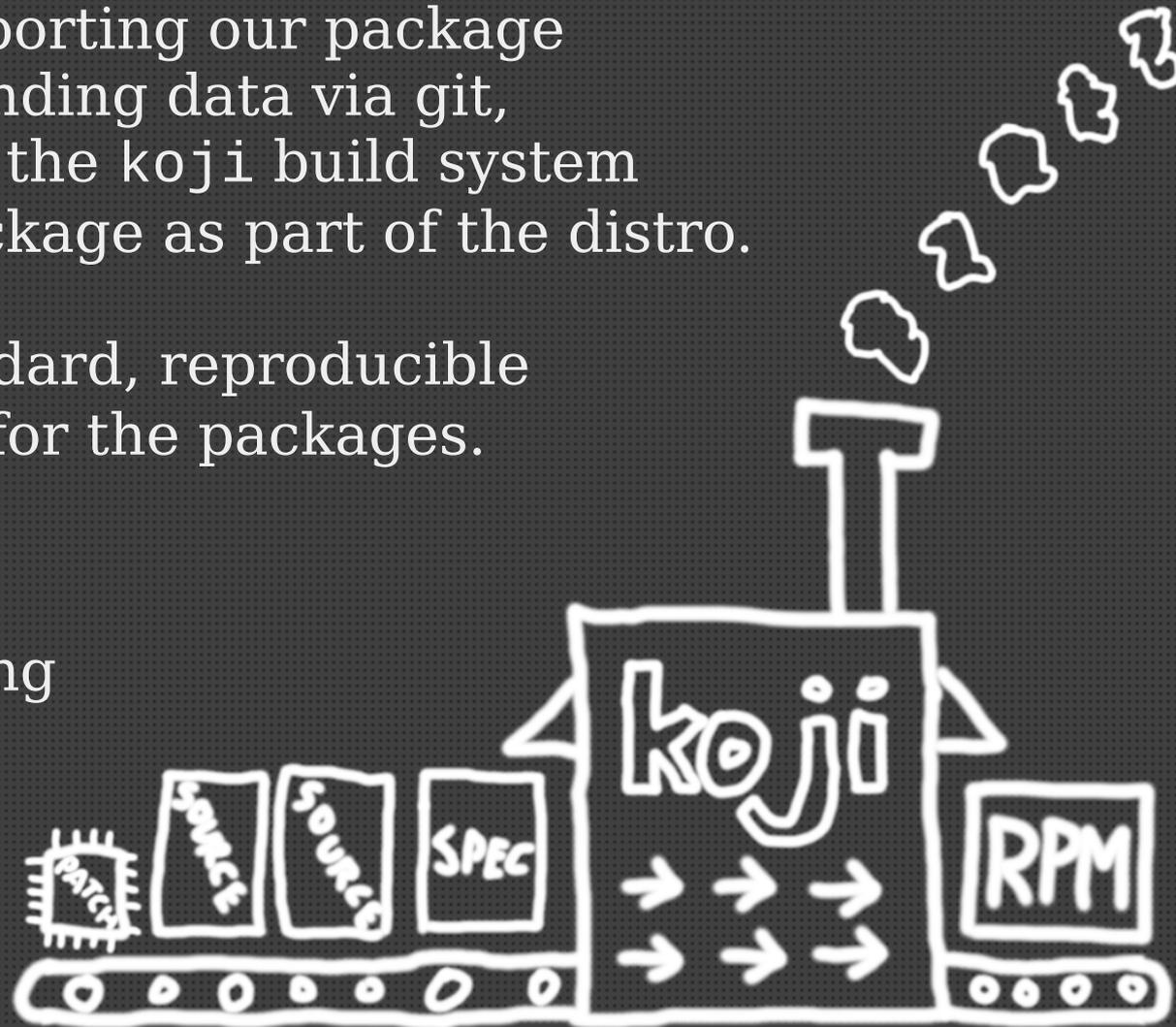


Step 4: building the package

After importing our package and sending data via git, we can use the koji build system to build the package as part of the distro.

koji provides a standard, reproducible build environment for the packages.

We can also use koji for testing, by performing *scratch builds*.



Step 5: proposing an update and testing

Once our package has been successfully built, we can use the bodhi system to propose updating the package in the repository.

Updates sent to bodhi first go into the testing repository - updates-testing.

Distro users can download test updates and send their comments via bodhi.

The testing period lasts one week.

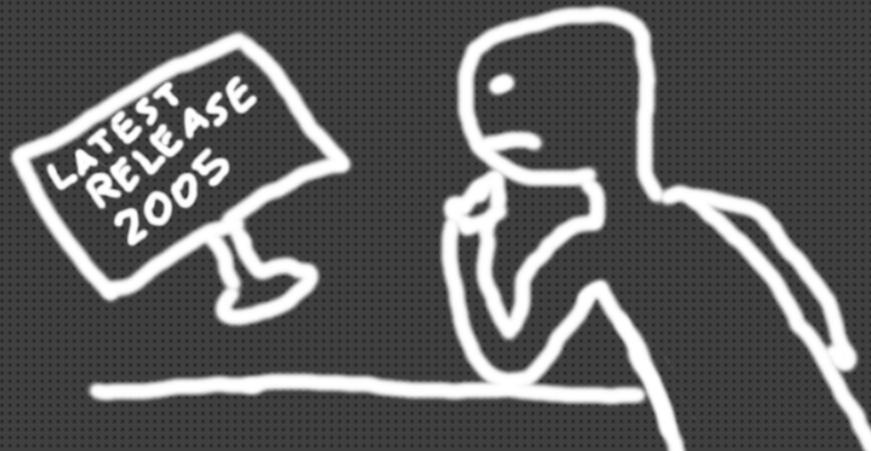
A package can be pushed to the main repository earlier if it receives enough "plus one" votes.



Step 6: waiting for some news

Our package made it into the repository. *Yay!*

Now all that's left is to visit the project's website every once in a while to check for new releases.



Step 7: orphaning

A sad fate that many packages will eventually face is orphaning – being abandoned by their current maintainer.

A package can be orphaned “right away” – most often because the packager decided they no longer have the will or the time required to maintain it.

It can also be taken away from the maintainer, e.g. when they become non-responsive.



Step 8: adoption

Being orphaned is not the end of the world – a package that has been orphaned can be adopted.

To do this, it's usually enough to open a ticket with *Release Engineering* and express your intent on the mailing list.



Step 9: retirement

Sometimes packages don't have so much luck and nobody wants to maintain them after they're orphaned.

Packages orphaned for six weeks are retired.

A package can also be orphaned "right away" by its maintainer.

This usually happens to software which is no longer supported by its authors and contains serious bugs affecting its usefulness, stability, or security.



Step 10: coming back from retirement

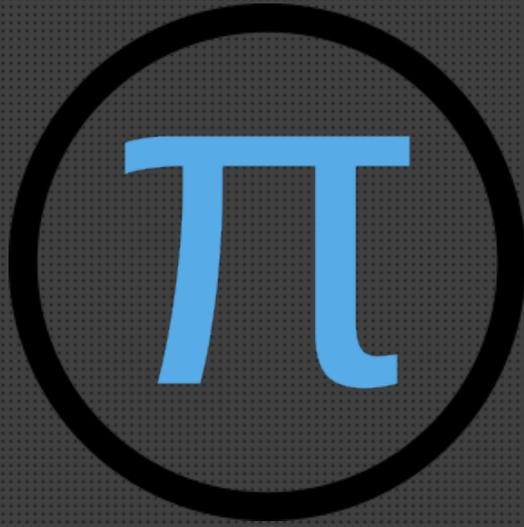
It is not dead which can eternal lie...

"Call of Cthulhu", H.P. Lovecraft

There is a possibility to return a package from retirement. The process is similar to adopting a package – one must ask *Release Engineering* to restore the package and be given admin rights, and announce this on the mailing list.

Packages that have been retired for over two weeks must once again go through the review process.





POZNAŃSKA IMPREZA
W O L N E G O
OPROGRAMOWANIA

Thanks for listening

Questions?

blog: blog.svgames.pl
e-mail: iwicki.artur@svgames.pl
twitter: [@suvepl](https://twitter.com/suvepl)



slides available under the CC-BY-SA 4.0 licence