Cooperating with upstream projects

Packaging tips and tricks

Philippe Coval
Tizen engineer

<philippe.coval@open.eurogiciel.org>
Who am I ?

- **FLOSS enthusiast**
  - Communities: MeeGo/harmattan, debian, Qt, gnome, meamo, jolla ...

- **Tizen packager, maintainer and developer**
  - Co maintenance :
    - Domains: Automotive, Graphics & UI frameworks, Multimedia...
  - Focus on core distro, graphics, toolkits (EFL, Qt ... )
  - Hacked webapps on Tizen RDPQ, ia32 mobile, ARM

- **Works for Eurogiciel Opensource Dept**
  - Intel OTC contractor for 2 year located in France (Brittany)
Agenda

• Cooperation matters
  • Upstream opensource projects
  • Tizen integrate them

• A bug life
  • Patches, trackers and git flow

• Communities

• Ask for Tips
  • Tools: git, gbs, rpm
Cooperation matters
Cooperation matters: Why?

- **Goals**: Philosophy + Pragmatism
  - **Maximize**: result / (efforts * time)
    - Best skills on worst issues
    - Better quality (Several POV More tests = More feedback)

- **Benefit for tizen (system)**
  - Focus on **Integration** not development
  - Long term **maintenance**

- **Benefit for upstream (softwares)**
  - Focus on **Development** not Integration
  - Users base / testing real use case

- **Tizen:Common : Bleeding edge**
  - Wayland, connman, gfx stack (intel drivers)
Cooperation matters: How?

- **Communicate**
  - In the open several places: connect tizen to upstream
  - Be nice, patient and proactive
- **Use efficient tools:**
  - git, gbs
- **Track changes**
  - Avoid loosing contexts
  - Ease up co maintenance
- **Community: + not vs**
  - Downstream + upstream
  - Inside project + Outside project
A bug's life
Communicate

- Is the bug or feature **tracked**?
  - No: Open bug [http://bugs.tizen.org](http://bugs.tizen.org), (TC if not profile specific)
  - Yes: let know your plans

- Is the bug **specific** to tizen?
  - Yes: fix, commit mention “Bug-Tizen: TC-42”

- Is this and **upstream** bug?
  - Yes: Open an item on upstream tracker or mailing list
  - Link it on tizen tracker ASAP

- Benefits:
  - Avoid duplication of efforts / Long term maintenance
Package for Tizen

- All packages are in VC :-)  
  - check repo manifest
- Get the sources: create **upstream branch**  
  - git clone -b upstream $git_upstreamurl  
  - gbs import ../$package-$version.tar.gz
- **RPM** packaging  
  - create “packaging/$package.spec”
- Build package along tizen rpm repos  
  - gbs build
- Ask me for tips:  
  - Avoid packaging mistake, git submodules, community repo ...
Upstream & tizen git branches

```
{master}@upstream
...

{master}@upstream
<0.4.1>

{upstream}@tizen
<upstream/0.4.1>

{tizen}@tizen

packaging: Initial packaging on 0.4.1 for Tizen
Bug-Tizen: TC-41
```

git checkout -b upstream $tag
git checkout -b tizen
git add packaging/*.spec
gbs build
Develop on Tizen

• Use Tizen:Common if possible
  • Minimal config, then will land in other profiles (IVI)
  • Ability to hack, build and run on a target (desktop)

• **Track** changes : using DEP3 conventions
  Bug-Tizen: TC-42
  Bug: 2014

• **Verify** you changes, check packaging, test

• Ask for Tips :
  • Sharing code in sandboxes
  • Avoid spec files mistakes
Local change on tizen branch

{tizen}@tizen
packaging: Initial packaging on 0.4.1 for Tizen
Bug-Tizen: TC-41

{tizen}@local
Fix bug Y by adding feature X
Bug-Tizen: TC-42

{master}@upstream

$ git rebase remotes/origin/devel
$ git format-patch -o ../patches/ HEAD~1
Show for feedback

- Use upstream bug report at **entry** point
- Rebase on upstream's devel branch
- Check your changes are clean
  - Is tracked (using Bug:)
  - Keep the changes **minimal** : (tizen packaging free)
- Publish patch
  - and ask authors kindly for reviewing code
- Take care of it
  - Take feedbacks into account, **improve** to reach consensus
  - Make it evolve until agreement
  - Be patient, constructive and never give up :)
Change rebased on upstream dev branch

Fix bug Y by adding feature X
Bug: 2014

packaging: Initial packaging on 0.4.1 for Tizen
Bug-Tizen: TC-41
Get the change merged upstream

- is it reviewed?
  - positive feedback? none negative?

- Is it merged upstream?
  - git cherry-pick, reword commit message using DEP3 tags:
    Bug-Tizen: TC-42
    Origin: $upstream_url
  - git snapshot

- Is change in new released version?
  - Rebase on new version:
    git commit -sam \\
    'packaging: Bump to 1.4.2\nBug-Tizen: TC-42'
6/ Update version will bring the change

{master}@upstream

Fix bug Y by adding feature X
Bug: 2014

<0.4.1>

{tizen}@tizen:
packaging: Bump to 0.4.2
Bug-Tizen: TC-42

packaging: Initial packaging on 0.4.1 for Tizen
Bug-Tizen: TC-41

<0.4.2>

git checkout upstream
git rebase -i $new
git checkout tizen
git rebase -i upstream
git commit
But in reality?

- Upstream integration will happened when ready
- but it was needed for yesterday
  - **time out** expire (TBD ~1 week)
  - If no negative feedback
  - Just commit and push it to tizen reviewing branch
    Tags: “Bug-Tizen:” “Bug:” “Origin:“ ...
  - Risky ? Then try to make the changes conditional
    - Packaging flag or env variable
    - Don’t mix upstream changes and tizen/packaging ones
- Half done job
  - Keep an eye for next version
  - **maintain** change downstream (and maybe also upstream)
Downstream change on tizen branch

{tizen}@tizen
Fix bug Y by adding feature X
Bug: 2014
Bug-Tizen: TC-41
Origin: $url_of_patch

{tizen}@tizen
packaging: Initial packaging on 0.4.1 for Tizen
Bug-Tizen: TC-41

{master}@upstream
Workflow Summary

- Report, assign, comment bugs

- Develop on tizen system : Tizen:Common
  - Rebase on upstream

- Forward patch to upstream project for reviewing
  - Adapt if needed

- Merge it to tizen
  - reviewing tizen side too
Downstream patches examples

- **Downstream change**
  - Upstream: reviewed about to merge
    URL: https://codereview.qt-project.org/#change,84222
    Task-number: QTBUG-38633/part/2of2
  - Tizen: merged
    URL: https://review.tizen.org/gerrit/#/c/21158/
    Task-number: QTBUG-38633/part/2of2
    Bug-Tizen: TIVI-3113
    Origin: https://codereview.qt-project.org/#change,84222

- **Upstreamed changes**:
  - Connman: https://01.org/jira/browse/CM-655
Community
Community contribs

- Projects :
  - QtForTizen, tizen-sunxi, MonoTizen, yours?
- Platform :
  - Start packaging, file bug, look for mentors, share source
- Applications :
  - Let us know, share .wgt or code
  - https://wiki.tizen.org/wiki/Applications
- Community repo :
  - Binaries : for native packages and/or applications etc
  - Sources : shared project for pre-integration, maintenance facilities : http://gitorious.org/tizen
Brotherhood

- Don't reinvent the wheel: **Give and Take**
  - RPM distro / spec files
  - Meego: the root of Tizen
  - Mer: used as upstream for Qt5
  - OpenSuse / Fedora etc

- **Compare to other systems**:
  - Connectivity: connman. Bluez: Mer
  - Systemd: ArchLinux, Debian?
  - Xwalk, webkit / blink: android?
  - Efl, Wayland?
Extra tips and tricks
Ask me for tips

• Packaging
  • Initial packaging
  • git modules
  • gbs : upstream git or tarball
  • gbs : use upstream tags
  • rpm : macros
  • rpm : multi configuration

• Publishing
  • Git Sandboxes

• Commit changes
  • Split packaging / downstream
  • Bump version
Initial Packaging: Create upstream branch

- Git vs tarballs git (gbs import)
  - Git: Upstream will feel home and can cherry pick patches
  - Git: easier rebase on version bump
  - Tar: bring generated files
  - Other SCM: git-svn . Git-hg
- Share the source
  - Request creation of git repo on tizen.org
  - or share your own/community
Gbs : git modules

- Git in git (~ svn externals)
  - used for common code among projects
- Example
  - Source: platform/upstream/gstreamer-vaapi.git
  - Usage:
    ```
    git rebase upstream
    sh packaging/gitmodules.sh
    git commit -sam 'packaging: gitmodules refresh'
    packaging/*.tar.bz2
    ```
Packaging : RPM : Use macros

• Use macros :
  • Ie: not arch dependents path :
    /usr/lib != /usr/lib64 != %{_libdir}
  • Ie: systemd :
    - /usr/lib/systemd/user/*.service
    + %{_userunitdir}/*.service

• Tips :
  rpm --eval %{_sysconfdir} : /etc
  rpm -ql rpm | grep macros : /usr/lib/rpm/tizen/macros ...

• packaging/*.changes files vs git log ?
Packaging : RPM : Improve spec files

- Multiconfiguration packaging
  - Test features not profiles or versions
  - Ie: Graphics stack: (wayland vs X11 support)
  - Don't hardcode path uses variables (tz-platform-config)

- Release: Field in packaging/*.spec
  - Depends on versionned subpackages
    Requires: %{name} = %{version}-%{release}
  - 0 for gbs
  - Tip: increment on local build for repo:
    release?=$(shell date +%Y%m%d.%s)
    zypper update -r $USER
How developer can ease maintainers job

- Avoid to mix packaging changes and sources patches
  - benefits: upstream can pick your patch, and will skipped on rebase

- Identify packaging changes using prefix
  - Ie: git commit -sam 'packaging: Initial packaging on x.y.z for Tizen'
  - Easier maintenance (think git rebase -i)

- Keep source changes simplest
  - Ie: git commit -sam 'Install to $DESTDIR' Makefile
  - Forward upstream and track it
Maintainer updates to new version

- No need for upstream/x.y.z edit ".gbs.conf"
  ```
  [general]
  upstream_branch = upstream
  upstream_tag = libevdev-${upstreamversion}
  ```

- Need for snapshot versions?
  ```
  $ git describe : upstream/0.15.2+9+g40c3756
  ```

- Track dropped / changes
  ```
  $ git commit -sam 'packaging: Bump to x.y.z
  skip c7f2a1e... API: add support for buffer exports'
  ```

- Reset head after checks
  - gbs export : patch serie vs binary patch archive
  - git push -force
Developer need to share work in progress?

- sandbox/$USER/tizen
  - Based on /tizen branch (and then /upstream)
  - Purpose: push for review first and then share changes before merged
- sandbox/$USER/upstream
  - Upstream's master branch if needed to rebase (ahead upstream branch)
- sandbox/$USER/devel
  - Based on previous branch
- workaround for missing tags:
  ```
  $ cat $package/packaging/../.gbs.conf
  +upstream_branch = sandbox/$USER/upstream
  +upstream_tag = sandbox/$USER/upstream
  ```
Helper scripts and recipes

- tizen-helper
  - https://gitorious.org/tizen
  - Scripts and recipes to share
- Local Git cache
  - Locally:
    tizen-helper/bin/git.sh clone https://gitorious.org/tizen/tizen-helper.git
    alias git=git.sh
  - On a lan:
    git cclone https://github.com/kooltux/git-cache
References

- **Packaging / Workflow**
  - [https://wiki.tizen.org/wiki/Packaging/Guidelines](https://wiki.tizen.org/wiki/Packaging/Guidelines)
  - [http://dep.debian.net/deps/dep3/](http://dep.debian.net/deps/dep3/)

- **Community**
  - [https://gitorious.org/tizen](https://gitorious.org/tizen)
  - [https://wiki.tizen.org/wiki/Applications](https://wiki.tizen.org/wiki/Applications)

- **Me or Related WIP:**
  - [https://dockr.eurogiciel.fr/blogs/embedded/author/pcl/](https://dockr.eurogiciel.fr/blogs/embedded/author/pcl/)
Thanks

- **Tizen community** : for the spirit
  - Developers
  - Reviewers / Maintainers
  - Hackers
- **Sponsorship** : for helping to be there
  - Linux Foundation
  - Eurogiciel
- **Tizen Association** : for the project and TdcSf14 gifts
  - Intel
  - Samsung
- **You** : for caring of coop
Q & A
Ask me online
Open source development and integration:

Several Maintainers for tizen.org

Embedded systems for real-time multimedia:
- Widi/Miracast stack,
- Wayland/Weston,
- Webkit2 browser with HW acceleration,

Application: HTML5/CSS3, jquery, jqmobi, Cordova

Location: Brittany - France