

LTSI – Developer Experiences

Simon Horman <simon@horms.net>
Horms Solutions Ltd.

30th May 2013



History: Part I: Before the Beginning

■ Internal Project for Platform Team:

- 2.6.35.7 target (gingerbread)
- 2.6.38 upstream (initially)
- Improve quality of Drivers for Production Team
- Do so in a re-usable way
- Experiment with reproducible back-porting methods



History: Part II: The Beginning

- LTSI-3.0
 - 3.0.31 target
 - 3.4 upstream
 - Improve quality of Drivers for Production Team
 - Bring code re-use to a wider audience



History: Part III: Now

- LTSI-3.4
 - 3.4.25 target
 - v3.5, v3.6, v3.7 upstream



Target Selection Process

- SoC, Driver, etc... selection based on customer needs

Patch Pool for each Target

- Target files isolated in conjunction with domain expert
- Short-list of patches obtained using git
`git log --oneline v3.4..v3.7 -- x/y/z. [ch]`



Scope

- Full backport of target drivers, board, SoC, etc...
 - Reproducible
- Partial backport of dependencies
 - Limits scope creep



Observations: Many Components

- Requests for many components to be back-ported
- A sign of success?

Observations: Merges

- When back-porting is useful to use different git branches for different components.
 - Allows testing on individual components
 - Allows components to act as discrete dependencies for other components.
 - Components can be combined using merges.



Observations: Merges

- When back-porting is useful to use different git branches for different components.
 - Allows testing on individual components
 - Allows components to act as discrete dependencies for other components.
 - Components can be combined using merges.
- In my experience consumers of backports want single branch; a flat patch series



Observations: Merges

- When back-porting it is useful to use different git branches for different components.
 - Allows testing on individual components
 - Allows components to act as discrete dependencies for other components.
 - Components can be combined using merges.
- In my experience consumers of backports want single branch; a flat patch series
- Git cherry-pick and rebase in conjunction with rerere are rather handy to create single branch



Questions?

