Introduction to Partner Engineering

John Feeney, et al.
Kernel Platform Enablement, Red Hat

Legal terminology

- Do we need to find a legal signoff for this?
  - Doubt it but check with peterm and jfeeney
- Remove “No further distribution” after we finalize v1

Overview of Presentation

- Welcome to Red Hat's Partner Engineer Program
- Benefits
- Red Hat Expectations
- Partner Engineer Responsibilities
  - Status Reports, Bugzilla (Bugs and Feature Requests)
  - Schedules and Bugzilla Priorities
- What can/cannot be shared?

Welcome!

Welcome to Red Hat's Partner Engineer Program

- Historically, no way for a partner to easily get code into Red Hat products
  - Scheduling conflicts
  - Engineering conflicts (code reviews)
  - Hardware conflicts
- In 2004, Red Hat introduced Onsite Engineer Program
- Now called “Partner Engineer” Program

Welcome to Red Hat's Partner Engineer Program

- Partner Engineer
  - Schedules
  - Code review lists
  - Unreleased builds and composes
  - Internal IRC and communication
- Partner Company
  - Influences RHEL
  - Dedicated person resolves problems!

Benefits

Benefits: Schedule Information

- Internal schedule deadlines
  - Hardware
  - Feature Requests
  - Patches
- Your opinion on bugzillas and status
- Deadlines for various releases (alpha, beta, etc.)

Benefits: Code and Communication

- rhkernel-list@redhat.com
  - Internal LKML-like mailing list where we submit patches for RHEL
  - Other development lists as necessary
- Internal IRC
  - #kernel, #partnereng
  - Other channels as necessary
- Onsite
  - Better access to everyone!
  - More holidays!

Benefits: Red Hat

- Red Hat Benefits
  - YOU :)
  - Better understanding of your company's concerns (bugs)
  - Better knowledge of your company's products

Red Hat Expectations

Partner Engineer Expectations

- Like a Red Hat Engineer
  - Not adversarial
- “Red Hat wants you to succeed”
- Help with patch submissions
- HW/SW debug
  - Internal process
  - If we can ... upstream

Red Hat Expectations

Depending upon the situation, the partner engineer may have to:

- Help acquire and support hardware in RH
- Create feature request bugzillas
- Modify/Backport code in upstream Linux
- Backport fixes to RH product(s) according to the RH process in a manner that adheres to the specific release schedule
- Test RH releases for parent company's hardware
- Be aware and conform to the release schedules

Partner Engineer Responsibilities

Partner Engineer Responsibilities

- Feature Requests
- Bugzilla (real bugs)
- Weekly Testing
- Schedule
- Status Reports

Feature Requests

- All new features require feature requests
- Feature requests allow the partner to request what should be included in RHEL, new or updated drivers and/or packages.
- Filed in Bugzilla by Feature Request deadline, cc your Partner Manager
- Note that all feature requests are reviewed by RH Product Marketing before they are accepted.
- For details on Feature Requests with examples, refer to RHProcess1.odt on people.redhat.com/jfeeney/.docs

Feature Request Template

- 1. Feature Overview:
- a) Name of feature:
- b) Feature Description:
- 2. Feature Details:
- a) Architectures:
- b) Bugzilla Dependencies:
- c) Drivers or hardware dependencies:
- d) Upstream acceptance information:
- e) External links:
- f) Severity (U,H,M,L):
- g) Target Release Date:
- 3. Business Justification:
- a) Why is this feature needed?
- b) What hardware does this enable?
- c) Forecast, impact on revenue?
- d) Any configuration info?
- e) Are there other dependencies (business drivers).
- 4. Primary contact at Red Hat, email, phone (chat)
- 5. Primary contact at Partner, email, phone

Bugs

- Cc'd or assigned to partner engineer
- Cc partner manager
- Enter in bugs at https://bugzilla.redhat.com/
- For details on the handling of bugs, refer to RHResolutionProcess1.odt on people.redhat.com/jfeeney/.docs

Bugzilla States

NEW*	- nobody working on it
ASSIGNED*	- an engineer is working on it
POST*	- patch is posted internally for review
MODIFIED	- patch is officially committed to tree
ON_QA	- kernel with patch is released for testing
VERIFIED	- QE has verified patch
RELEASED	- kernel with patch is released publicly

*Partner engineer can control these states

Bugzilla Prioritization

- Priority List
    - Per release (7.1, 6.7, etc.)
- Input from Partner Engineer and Company
- Red Hat customer input
- Reporter has idea of status

Schedule

- Default naming scheme is X.Y.Z
- Every major release has minor releases.
  - Each minor release has a schedule associated with it.
- Every minor release has update releases, or “z-stream” releases
  - Each z-stream has a schedule associated with it
- RHEL7.4, RHEL6.8, RHEL7.3.z, etc.

Schedule

Each release has dates defined that the partner engineer needs to be aware of.

- Partner Feature Request Deadline
- Partner Hardware Sent to RH Deadline
- Provide Feedback (yes, no, maybe) on Specific Bzs Deadline
- Partner Patch Submission Deadline: Features done
- Internal Patch Submission Deadline
- Deadlines for Beta, Snapshots, and RC: Bug fixes only

Minor Release Schedule Example

– Feature Request Deadline:	January 15
– Hardware Sent to RH Deadline:	January 30
– Provide Feedback:	February 15
– Partner Patch Submission:	March 20
– Internal Patch Submission:	April 1
– Beta:	April 20
– Snapshot 1:	April 27
– Snapshot 2:	May 4
– Snapshot 3	May 11
– Snapshot 4	May 18
– Snapshot 5	May 25
– GA	June 15

Schedule and Bugzilla

- Release X.Y Priority List
- Partner Engineer needs to provide feedback on the status of a BZ
    - devel_ack+, “yes”
    - devel_ack-, “no”
    - conditional_nack, “no, but if there is hardware/patch/reproducer then maybe” or
        - defer to the next release
        - Close it

What can and cannot be shared?

Sharing information

- Consider everything Restricted Information
- “Err on the side of caution”
- If you want to share anything ask your mentor or partner manager

What cannot be shared with my company?

- Do not send back RH internal schedules, source code, binaries, rpms, specifications, emails, or anything RH confidential without permission from a RH manager.
- If there is a rpm that needs to be sent to be tested, the partner engineer <u>must</u> ask permission of RH manager before doing so.
  - Why? Because the code might be under security embargo and/or certain partner's products might be confidential until a specific time.

END

Links/TODO

- http://bugzilla.redhat.com
- Check against jfeeney's presentations
- Partner Engineering Confluence?
- Go through PE checklist individually to get an idea of what PE is capable of?
-