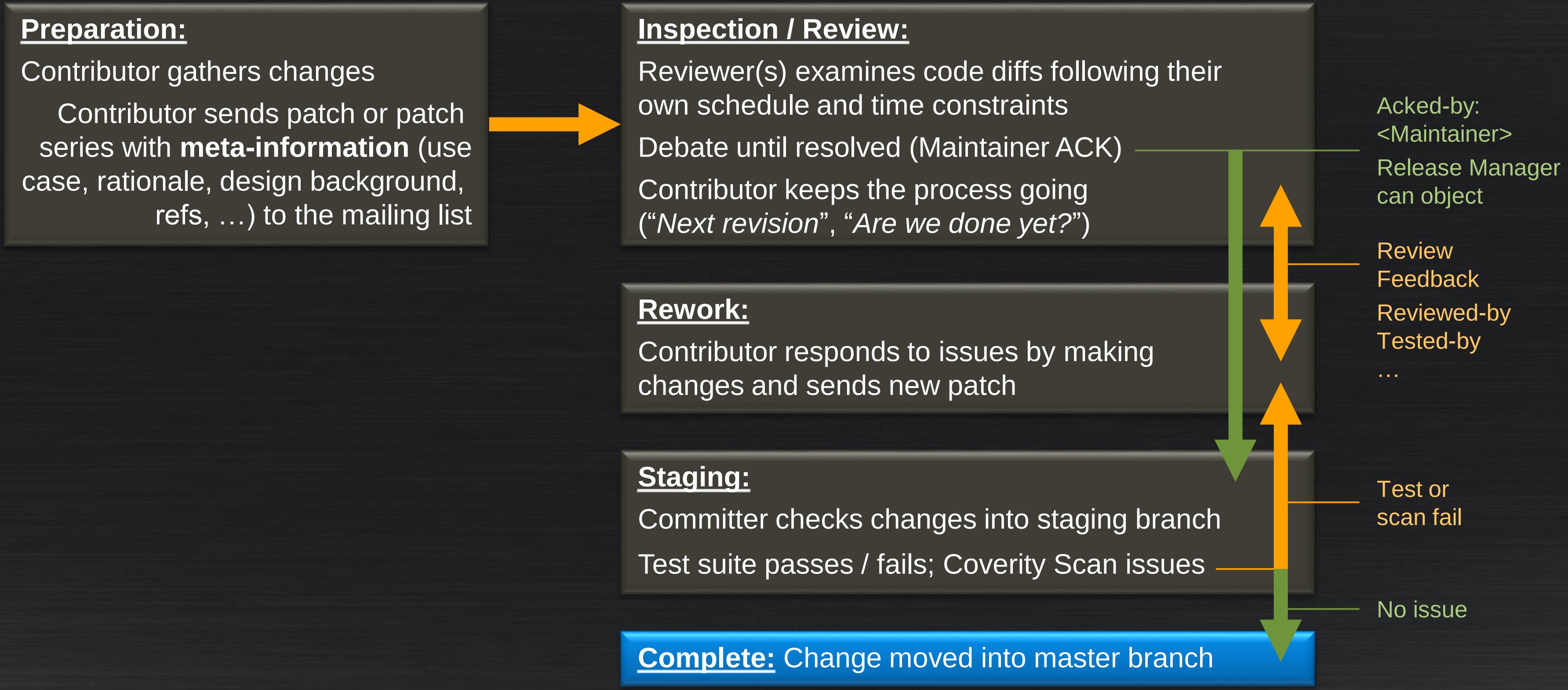
Contributions:

wiki.xenproject.org/wiki/Submitting_Xen_Project_Patches

www.xenproject.org/help/contribution-guidelines.html

E-mail based review Process



Preparation:

Contributor gathers changes

Contributor sends patch or patch series with **meta-information** (use case, rationale, design background, refs, …) to the mailing list

Inspection / Review:

Reviewer(s) examines code diffs following their own schedule and time constraints

Debate until resolved (Maintainer ACK)

Contributor keeps the process going (“*Next revision*”, “*Are we done yet?*”)

Rework:

Contributor responds to issues by making changes and sends new patch

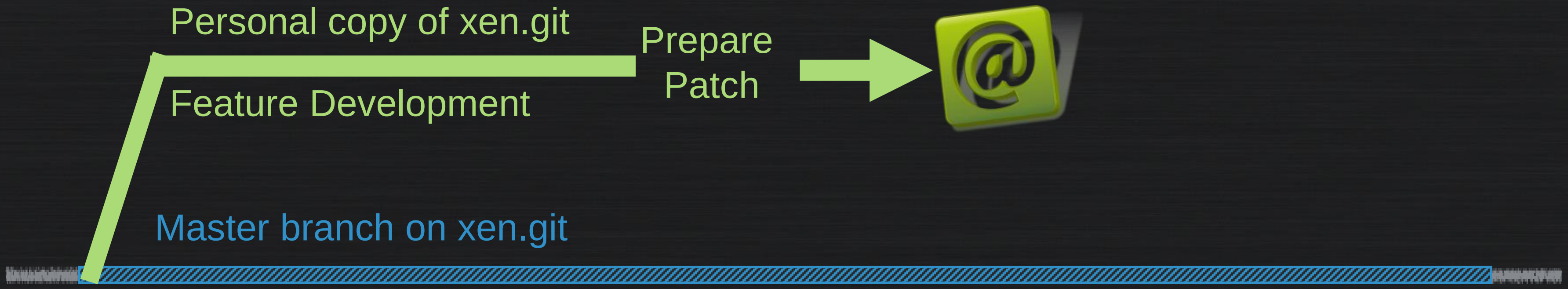
Staging:

Committer checks changes into staging branch

Test suite passes / fails; Coverity Scan issues

Complete: Change moved into master branch

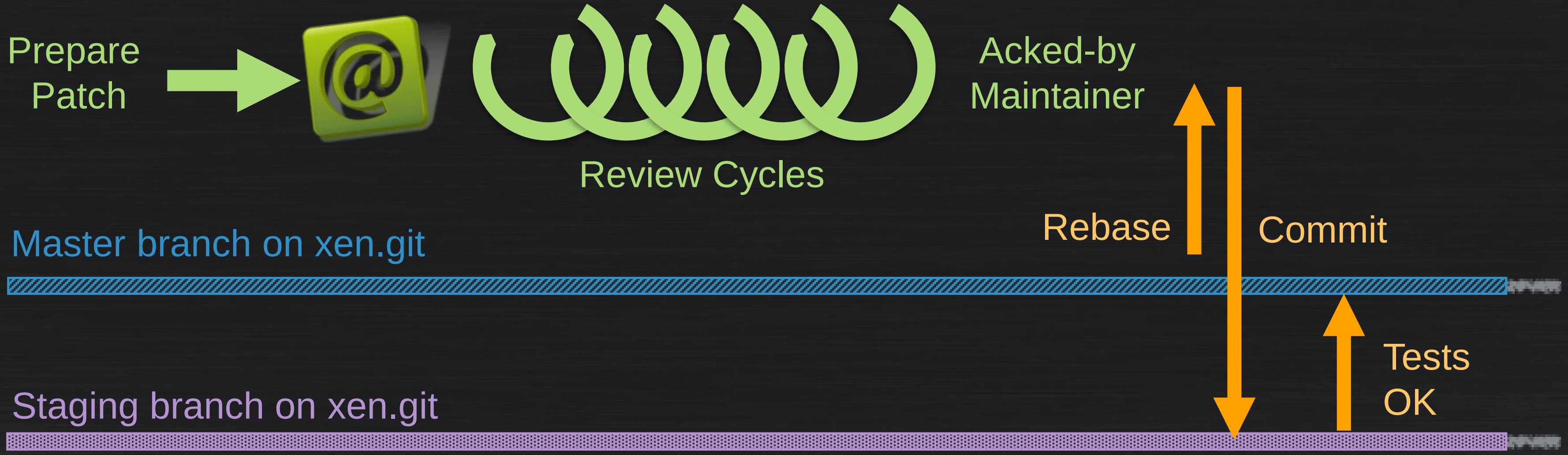
Review Process : Codeline View



Note: that you can request a personal development git tree to be hosted on xenbits.xenproject.org by sending an email to xen-devel@

These are listed on http://xenbits.xen.org/gitweb/ under people/*

Review Process : Codeline View



Prepare Patch

Review Cycles

Acked-by Maintainer

Rebase

Commit

Master branch on xen.git

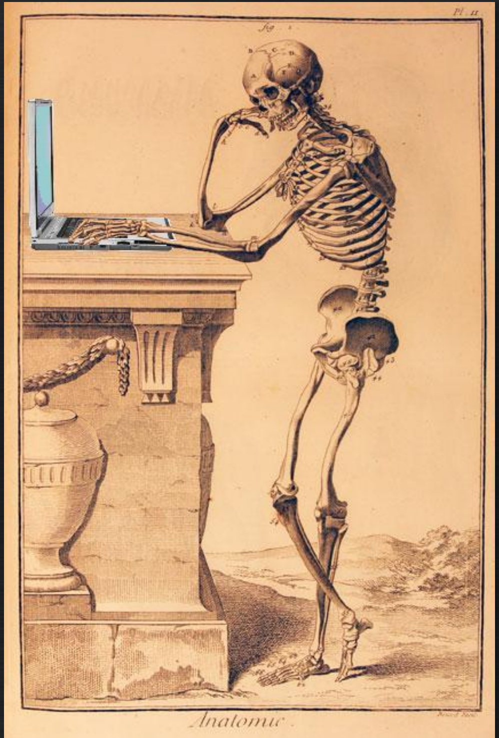
Tests OK

Staging branch on xen.git

Contains a limited number of “pending” patches

Identifies patches that lead to test failures in OSSTest via bisection.

Pushes patches that do not lead to regressions automatically to master



Anatomy of a Good Patch Series:

Or what makes a good patch (series)

Mike Licht @ Flickr

Key parts of a Patch / Patch Series

- Description of patch series : provides
  - Information about its use, use-case and users (if not obvious)
  - Information about the design, architecture, assumptions, etc. (if not obvious)
  - References to relevant context information - e.g. specs, etc. – (if relevant)
  - Short description of elements of the patch series and how they relate to each other (if necessary)
- Each patch contains
  - Description of the change and what it is for
  - Ideally also contains an analysis of the problem solved and why it is solve this way
  - The patch itself (ideally less than 200 LOC, reviewable in < 1 hour)
  - API documentation and test cases (if appropriate)
  - Sign-off-by : you state that you abide by the Developer Certificate of Origin

Issue: Unclear Description

marc.info/?l=xen-devel&m=141407695210809

This driver uses hwdom to change frequencies on CPUs

marc.info/?l=xen-devel&m=141407702410864

Xen changes frequencies on CPUs using this high-level cpufreq driver.

marc.info/?l=xen-devel&m=141492507021019&w=2

This patch series are only the Qemu part to enable Xen stubdom vTPM for HVM virtual machine. it will work w/ Xen patch series and seaBios patch series.

Build it with --enable-tpm and --enable-xen options and link with Xen, or change QEMU_STUBDOM_VTPM compile option from 'n' to 'y' in <Xen>/Config.mk, when the Qemu/ SeaBios patch series are merged.

Run Xen virtual machine with below QEMU command line options: "-tpmdev xenstubdoms,id=xenvtpm0 -device tpm-tis,tpmdev=xenvtpm0" or Xen xl tool adds this options when virtual machine cfg includes: **vtpm=["backend=vtpmN"]**

qemu-system-* --tpmdev help

Supported TPM types (choose only one): passthrough - Passthrough TPM backend driver xenstubdoms; Xenstubdoms -TPM backend driver

Example of a better description

lists.xenproject.org/archives/html/xen-devel/2014-10/msg00993.html

This patch series breaks multiboot (v1) protocol dependency and adds multiboot2 support. It lays down the foundation for EFI + GRUB2 + Xen development. Detailed description of ideas and thoughts you will find in commit message for every patch. If something is not obvious please drop me a line.

Patch #13 reveals a bug which probably was introduced between commit 3e2331d271cc0882e4013c8f20398c46c35f90a1 (VT-d: suppress UR signaling for further desktop chipsets) and 61fdda7acf3de11f3d50d50e5b4f4ecfac7e0d04 (x86/HVM: properly bound x2APIC MSR range). Xen crashes at video_endboot() because earlier scrub process wipes vga_console_info data (sic!). So, it means that at least page containing this structure was freed mistakenly somewhere. Interestingly this issue appears on legacy BIOS machines only. EFI platforms work as usual. It is possible to workaround this bug by passing no-bootscrub to xen.gz.

I was not able to spot anything obvious just looking briefly at commit history. I am going to narrow down and fix this issue in next release.

ARM build has not been tested yet. Most of the requested things are fixed but there are still some minor outstanding issues (multiboot2 tags generation, excessive amount of casts in xen/arch/x86/boot/reloc.c, etc.; please check commit messages for more details). If something is not fixed yet it means that I do not have good idea how to do that. In case you spot something which was mentioned during previous reviews and still think that your comment is valid in particular case please notify me.

Example of a good description of an Individual Patch

lists.xenproject.org/archives/html/xen-devel/2014-11/msg00525.html

```
When using pvgrub in graphical mode with vnc, the grub
timeout doesn't work: the countdown doesn't even start.
With a serial terminal the problem doesn't occur and the
countdown works as expected.
```

Problem Description

```
It turns out that the problem is that when using a graphical
terminal, checkkey () returns 0 instead of -1 when there
is no activity on the mouse or keyboard. As a consequence
grub thinks that the user typed something and interrupts the
count down.
```

Analysis of why it occurs

```
To fix the issue simply ignore keystrokes returning 0, that
is the NUL character anyway. Add a patch to grub.patches to
do that.
```

How the patch fixes it

Example of a good Patch Series description

lists.xenproject.org/archives/html/xen-devel/2014-08/msg02369.html

The x86 architecture offers via the PAT (Page Attribute Table) a way to specify different caching modes in page table entries. The PAT MSR contains 8 entries each specifying one of 6 possible cache modes. A pte references one of those entries via 3 bits: _PAGE_PAT, _PAGE_PWT and _PAGE_PCD.

The Linux kernel currently supports only 4 different cache modes. The PAT MSR is set up in a way that the setting of _PAGE_PAT in a pte doesn't matter: the top 4 entries in the PAT MSR are the same as the 4 lower entries.

This results in the kernel not supporting e.g. write-through mode. Especially this cache mode would speed up drivers of video cards which now have to use uncached accesses.

OTOH some old processors (Pentium) don't support PAT correctly and the Xen hypervisor has been using a different PAT MSR configuration for some time now and can't change that as this setting is part of the ABI.

This patch set abstracts the cache mode from the pte and introduces tables to translate between cache mode and pte bits (the default cache mode "write back" is hard-wired to PAT entry 0). The tables are statically initialized with values being compatible to old processors and current usage. As soon as the PAT MSR is changed (or - in case of Xen - is read at boot time) the tables are changed accordingly. Requests of mappings with special cache modes are always possible now, in case they are not supported there will be a fallback to a compatible but slower mode.

Summing it up, this patch set adds the following features: - capability to support WT and WP cache modes on processors with full PAT support - processors with no or uncorrect PAT support are still working as today, even if WT or WP cache mode are selected by drivers for some pages - reduction of Xen special handling regarding cache mode

Patch series with Design Information

marc.info/?l=xen-devel&m=123003693614292

This set of patches introduces a set of mechanisms and interfaces to implement populate-on-demand memory. The purpose of populate-on-demand memory is to allow non-paravirtualized guests (such as Windows or Linux HVM) boot in a ballooned state.

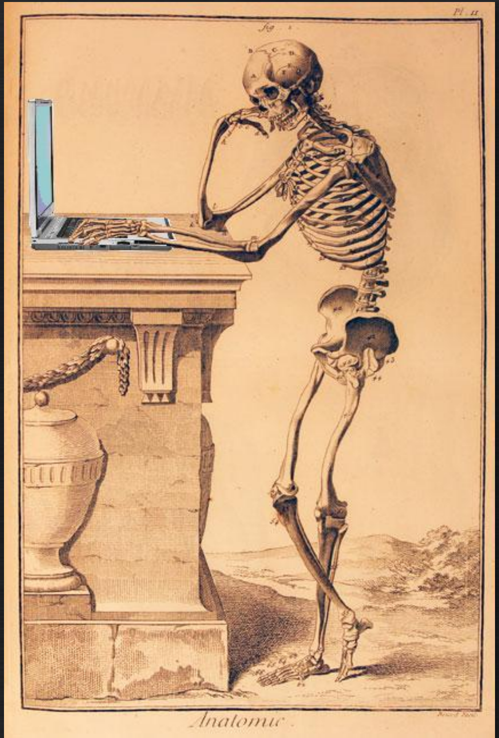
BACKGROUND

When non-PV domains boots, they typically read the e820 maps to determine how much memory they have, and then assume that much memory thereafter. Memory requirements can be reduced using a balloon driver, but it cannot be increased past this initial value. Currently, this means that a non-PV domain must be booted with the maximum amount of memory you want that VM every to be able to use.

Populate-on-demand allows us to "boot ballooned", in the following manner:

- Mark the entire range of memory (memory_static_max aka maxmem) with a new p2m type, populate_on_demand, reporting memory_static_max in th e820 map. No memory is allocated at this stage.
- Allocate the "memory_dynamic_max" (aka "target") amount of memory for a "PoD cache". This memory is kept on a separate list in the domain struct.
- Boot the guest.
- Populate the p2m table on-demand as it's accessed with pages from the PoD cache.
- When the balloon driver loads, it inflates the balloon size to (maxmem - target), giving the memory back to Xen. When this is accomplished, the "populate-on-demand" portion of boot is effectively finished.

…



Anatomy of a Good Patch Revision:

Addressing Feedback!
Common mistakes to avoid!

Mike Licht @ Flickr

Patch Revisions

Each new patch revision contains

- Change history (what has changed between each iteration)
- Tags (by maintainers, reviewers, testers, etc.)
  - Acked-by: <Maintainer>
  - Reviewed-by: <Reviewer>
  - Tested-by: <Tester>
  - Note that there is disagreement about use of Acked-by: <non-maintainer> which is being discussed amongst the community and not yet resolved (see http://lists.xen.org/archives/html/xen-devel/2014-06/msg01419.html)

Issue: Changes that were not asked for

marc.info/?l=xen-devel&m=140259327023935

```
> Add support for GIC v3 specification System register access(SRE)
> is enabled to access cpu and virtual interface regiseters based
> on kernel GICv3 driver.
>
> This patch adds only basic v3 support.
> Does not support Interrupt Translation support (ITS)

I see that a lot of things changed in this version of the patch. It w  ld be nice if you
kept a log of the incremental changes.

In particular in this version of the patch we basically added isb() everywhere. I would
like to see a written comment about it.

When Julien commented "isb?", I think he was just asking whether they are needed, not
requesting you to add them. The only one I was sure about was the one at the end of
gicv3_eoi_irq.

But it is good to see that you addressed all my comments.
```

Makes it **hard** for the reviewer to only re-review the portions of the patch that need to be looked at

Seems there was also a misunderstanding between reviewer and submitter

Issue: Incomplete Resend

marc.info/?l=xen-devel&m=140285195123082

You didn't address the comments I made on V4 for this patch. See a copy of them inline...

marc.info/?l=xen-devel&m=140422021120851

You continue to ignore my comments on this patch... Please explain why you don't want to make those changes. Sending 3 times the same patch without any change is a waste of time for both of us.

When you don’t understand understand feedback,
feel free to ask for clarification

Adds extra unnecessary iterations and wastes reviewer’s time.

Example of a good changelog

```
v3 - suggestions/fixes:
   - rename some variables
     (suggested by Andrew Cooper),
   - remove unneeded initialization
     (suggested by Andrew Cooper),
   - improve comments
     (suggested by Andrew Cooper),
   - further patch split rearrangement
     (suggested by Andrew Cooper and Jan Beulich).

v2 - suggestions/fixes:
   - improve inline assembly
     (suggested by Andrew Cooper and Jan Beulich),
   - use __used attribute
     (suggested by Andrew Cooper),
   - patch split rearrangement
     (suggested by Andrew Cooper and Jan Beulich).
```

Coding Styles:

Hypervisor Coding Style

xenbits.xen.org/gitweb/?p=xen.git;a=blob;f=CODING_STYLE

- Indentation
- White Spaces
- Line Length
- Bracing
- Comments
- Emacs Local Variables

Libxl Coding Style

xenbits.xen.org/gitweb/?p=xen.git;a=blob;f=tools/libxl/CODING_STYLE

Similar to QEMU and Linux with a few exceptions. Different from Hypervisor coding style.

- Whitespaces
- Line width
- Variable Naming
- Statements
- Block Structures

Tools to check coding styles

There are a number of uncommitted scripts, which automatically check coding style. These have not yet been fully reviewed and may not work in all cases, but may partly be usable. See

- lists.xen.org/archives/html/xen-devel/2014-09/msg01543.html
- lists.xen.org/archives/html/xen-devel/2014-09/msg01171.html

Other useful scripts:

- scripts/get_maintainer.pl : looks at the files you have modified in the patch and matches it up with the information in the MAINTAINERS file ➔ generates a list of people + email addresses who need to be CC'ed
  ➔ Some issues: see
  lists.xenproject.org/archives/html/xen-devel/2014-11/msg00060.html

Updated

Release Process:

blog.xenproject.org/2014/05/08/how-the-hypervisor-team-manages-releases/

Updated

Roles: Release Manager

- A Maintainer that is elected by the community to manage a specific release
  - Nominated by community members (but needs to have agreed to want to do the job and have the time to do so)
  - Elected using a formal vote by committers of the Xen Project Hypervisor team
  - For one or several releases
  - (Past) Release Managers: Ian Campbell (Citrix), George Dunlap (Citrix), Korad R Wilk (Oracle)
  - Xen 4.7 Release Manager: Wei Liu
  - Driving the Roadmap process
  - Deciding the Release Timetable
  - Making the Final call on what goes into the Release at each Freeze Point
  - ~~Handling requests for "freeze" Exceptions : Cost vs. Benefit analysis~~
  - Coordinates with other committers in making Release Candidates and Releases
- Other responsibilities
  - Helps with PR related to the release
  - Can also act as spokesperson for the project (if the Release Manager is allowed to by employer)

Updated

Release Process Goals

A bug-free release

An awesome release

An on-time release

A predictable release cadence

The Xen Project aims to do releases on a 6 month cadence (historically this was 9+ months). We aim to make a release

- At the beginning of each June
- At the beginning of each December

Notable changes to 4.7 Release Cycle

New

Reduce release cycle length to 6 months:

- 4 months development
- 2 months freeze, with earlier creation of release branch based on risk assessment
- Xen 4.7 will be slightly longer to support future June / December releases

Fixed and predictable Release dates

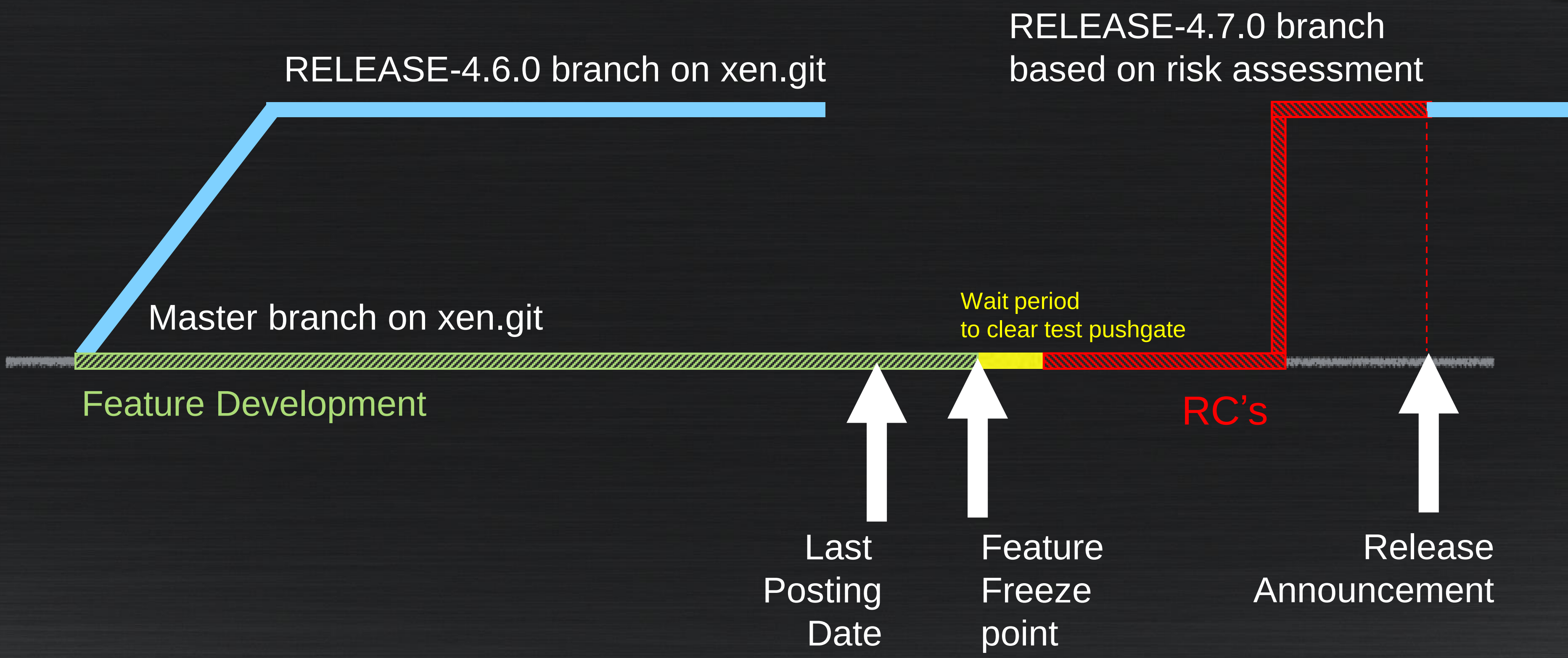
- Beginning of June and December of each year
- Eat into next cycle if we don't release on time

Fixed cut-off date:

- Freeze dates: last day of March and September of each year
- No more freeze exceptions, but heads-up mails about freeze will be sent a few weeks beforehand

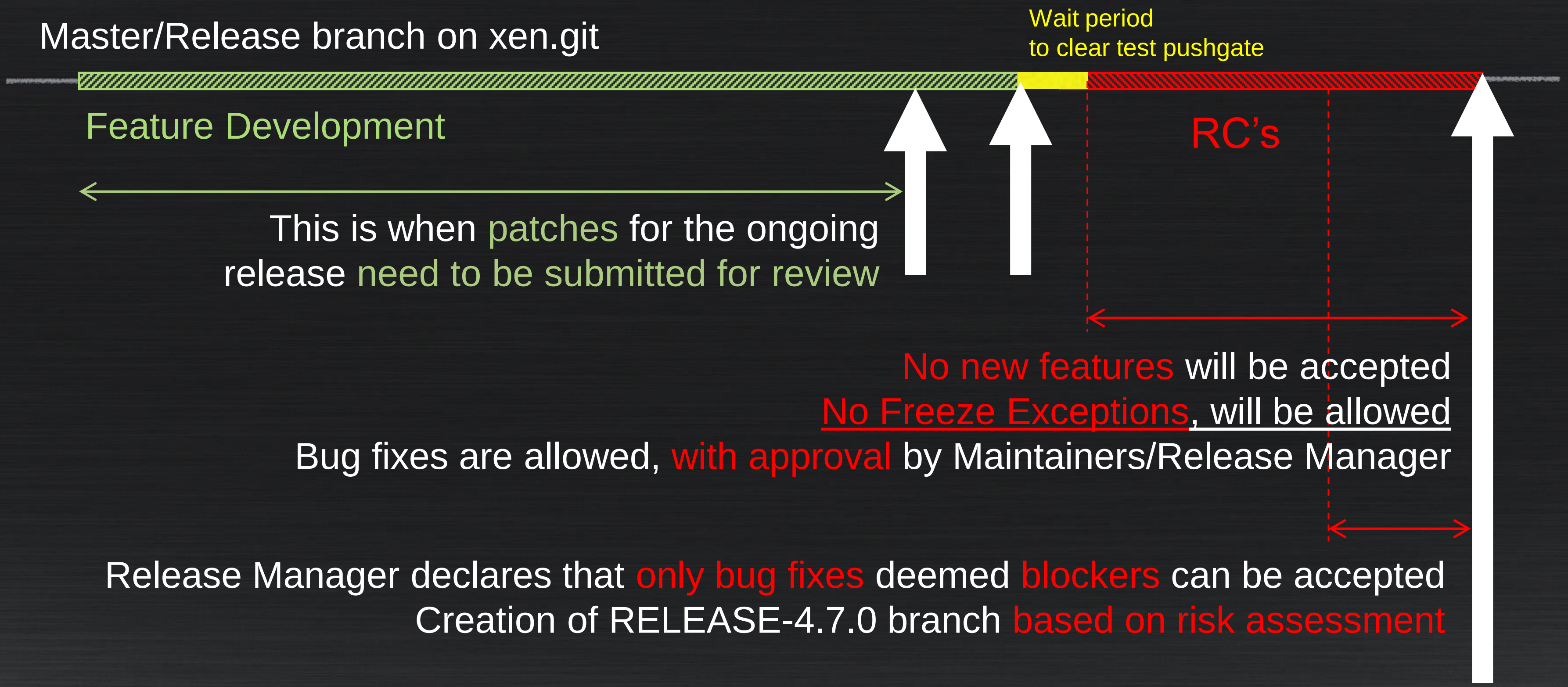
Maintenance support: no changes

Release Process : Stages



Updated

Stages & Gates in more detail



Master/Release branch on xen.git

Wait period
to clear test pushgate

Feature Development

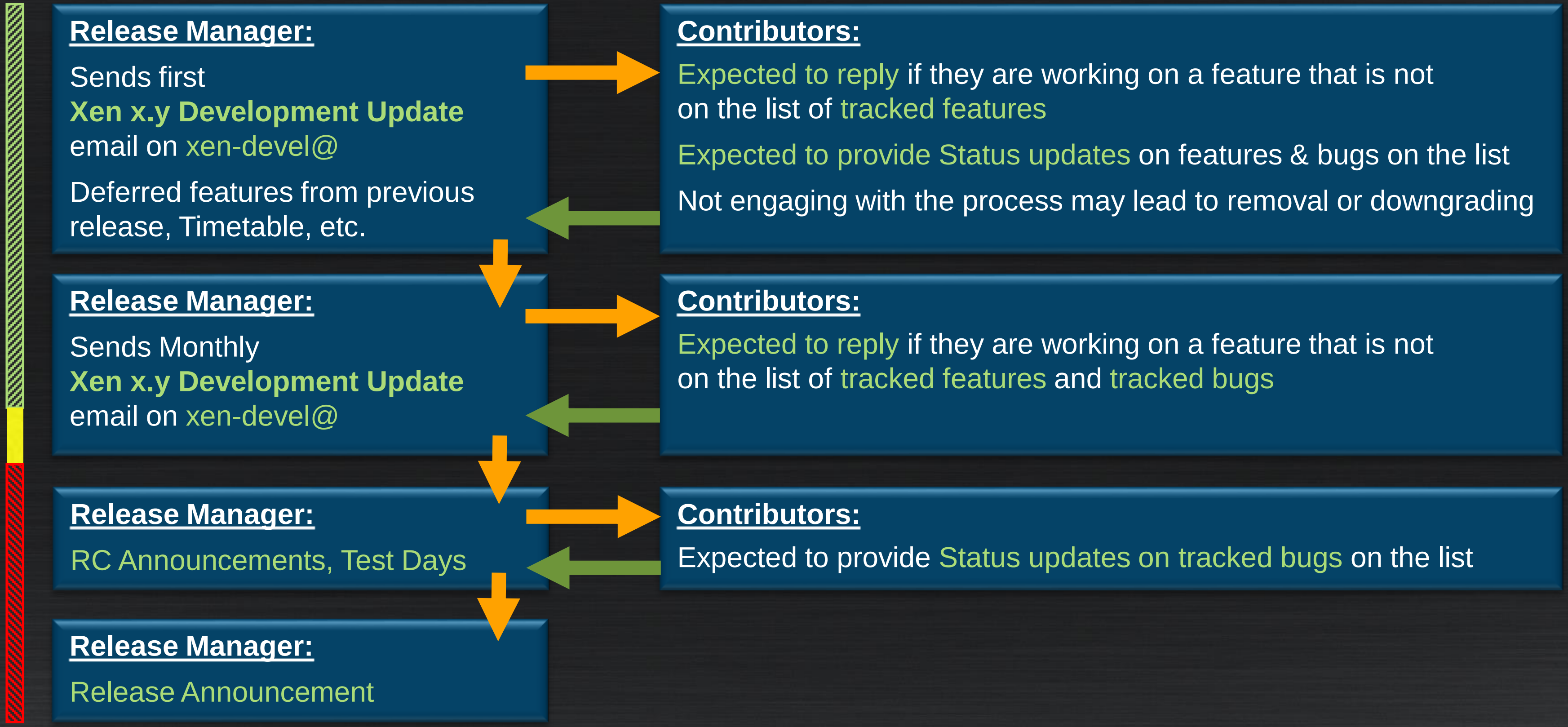
RC’s

This is when patches for the ongoing release need to be submitted for review

No new features will be accepted
No Freeze Exceptions, will be allowed
Bug fixes are allowed, with approval by Maintainers/Release Manager

Release Manager declares that only bug fixes deemed blockers can be accepted
Creation of RELEASE-4.7.0 branch based on risk assessment

Roadmaps : Tracking Contributions



Release Manager:

Sends first
Xen x.y Development Update
email on xen-devel@

Deferred features from previous release, Timetable, etc.

Contributors:

Expected to reply if they are working on a feature that is not on the list of tracked features

Expected to provide Status updates on features & bugs on the list

Not engaging with the process may lead to removal or downgrading

Release Manager:

Sends Monthly
Xen x.y Development Update
email on xen-devel@

Contributors:

Expected to reply if they are working on a feature that is not on the list of tracked features and tracked bugs

Release Manager:

RC Announcements, Test Days

Contributors:

Expected to provide Status updates on tracked bugs on the list

Release Manager:

Release Announcement

Criteria and Considerations

The release manager’s decision on a patch going in or out / after feature freeze date is in huge part influenced by:

- Risk of regressions:
  - Does this patch introduce a regression that will affect a large set of use-cases
  - Does this patch touch common code areas (code used by more components is less likely to be accepted, as it is more risky)?
  - Does this code affect specific cases only (e.g. MMIO pass through only) and is thus less risky?
  - Is this code simple and easy to understand and thus less risky?
- Has the code been tested for failures and succeeded multiple times?
- Has the code been Acked or Reviewed by the maintainer?

New

Release Process : Freeze Exceptions

No Feature Freeze Exception

- Prior to Xen 4.7, the project allowed Feature Freeze exceptions
- This has been dropped!
- Features patches must be submitted before the Last Posting Date

A Bug Freeze Exception

- Contributors can request Freeze Exceptions by replying to the **Xen x.y Development Update** mail.
- Bug freeze exceptions are part of the normal release process and trigger a cost-benefit analysis by the Release Manager with input from the community.

Testing & Coverity Scan:

Coverity Scanning Service

The Xen Project is registered with the "Coverity Scan" service which applies Coverity's static analyser to the Open Source projects

Because "Coverity Scan" may discover security issues the full database of issues cannot simply be made public.

To get access, consult

- www.xenproject.org/help/contribution-guidelines.html under “Code Security Scanning”

Test Lab and OSSTEST

The Xen Project has an automated test infrastructure that is run on

- Xen.git master
- Xen.git staging (pushgate) – see earlier
- Overview:
  - www.xenproject.org/help/presentations-and-videos/video/xpds14-osstest.html
  - xenbits.xen.org/gitweb/?p=osstest.git;a=blob;f=README;hb=HEAD

Can be run in stand-alone mode

- blog.xenproject.org/2013/09/30/osstest-standalone-mode-step-by-step/

Earning Status:

Earning Maintainer & Committer Status

Maintainers

- Demonstrate good technical knowledge in an area and submitted good patch series over a prolonged time period (6-12 months)
- Demonstrates engagement and engagement skills in the community via reviews
- Self-nominates, then signed off by other Maintainers

Committers

- Demonstrates prolonged engagement with the community upholding technical and community values – in other words demonstrates that he/she can be trusted
- Needs to be one of the top contributors to the project for a prolonged time period (e.g. >200 patches submitted per year; no objective rules)
- Gets elected by other maintainers

Updated

Meetings:

Face-2-face meetings and other community activities

Updated

IRC: #xendevel

- Was Invite only to avoid user questions
  - Open access, without explicitly advertising it
- A lot of coordination, quick pings, reminders happen on #xendevel

F-2-F: Xen Project Hackathons

- Once per year, hosted and paid for by a vendor
- 2 days long, 25-45 people depending on location
  - 95% developers
- Not so much a Hackathon, but a series of structured 1-1.5h discussions to solve architectural, design, review, process and other issues
  - Typically 3-6 discussions happen in parallel
  - Sometimes presentations are used, but interaction, discussion and decision making are the key focus of Hackathons
  - Discussions are minuted and posted on mailing lists
- Also see: blog.xenproject.org/2013/05/28/event-report-xen-hackathon-2013/

F-2-F: Xen Project Developer Summits

- Once per year, organized by the Xen Project
  - Co-located with LinuxCon Europe or North America
- Paid for by sponsorship and tickets, and subsidized by Xen Project
- 2 days long, 90-120 people depending on location
  - 70-80% developers
- Mainly presentations and conference format
  - Some interactive elements (BoF discussions)

F-2-F: Xen Project Developer Meeting

- Once per year, organized by the Xen Project
- Co-located with Xen Project Developer Summit (the day before or after)
- Typically we also have Working Group meetings and Advisory Board meetings on the same day
- 3 hours long, 20-25 core developers, maintainers and committers
  - Plenary format
  - 6-10 topics covered

Updated

Ad-hoc Meetings

- Community members can propose ad-hoc conference calls (or other on-line meetings) on specific topics on development lists
  - Example: regular IRC meeting on #xendevel to discuss x-Splice
  - Proposal made on xen-devel@
- Can be a conference call, IRC meeting, etc.
  - Community Manager can help set up