KDE Releases That Just Work(TM)

J.A. van Meeuwen

June 30, 2012
# Areas of Interest

- Stability of core components of Kolab product
  - KDE PIM, Kontact client,
  - Cyrus IMAP
- Happy Kolab Systems support staff
- Happy Kolab Systems customers
- Good relationship with happy developers
Jeroen van Meeuwen
Systems Architect, Kolab Systems AG

Areas of Interest

- Release early, release often
- Freezes halt (other) progress
- Increase ease for (...)

Areas of Interest

- Release early, release often
- Freezes halt (other) progress
- Increase ease for (...)

KDE Releases
J.A. van Meeuwen

Off We Go
Introduction
Ye Problem
Bugzilla
Target(s)
Challenge(s)

What To Do?
Tasks & Teams
QA
Cleanup
Versioning

KolabSystems
Current Bugzilla Status

- Many thousands of tickets with version “unspecified”.
- Many thousands of tickets with status “UNCONFIRMED”.
- Thousands of (open) tickets against legacy products.
- Thousands of tickets with no activity for more than 2 years.
- Generic (mailing list) assignment of tickets.
- Selected canconfirm and editbugs permissions.
- No one throat to choke.
- No use of CLOSED status.
- No DEFERRED resolution.
- Little to no automated maintenance (of Bugzilla).
- Very fast/frequent entry of new tickets.
Examples of Tickets that Linger Around

- "Not Our Problem"
  - #82679 - "uw-imap disconnect w/ second session"
  - lack of relevance, can’t be fixed, not ours to fix, ...

- "Not Going to Happen (Any Time Soon)"
  - #145645 - "do not wrap signature"
  - unless reporter puts up...

- “Closed, deferred”
  - #87331 - “Recurrence $xth and $yth DoW,oM”
  - lack of input, lack of feedback
Symptoms

- Some developers barely look at ticket(s) / queue(s)
Symptoms

- Some developers barely look at ticket(s) / queue(s)
  - Tickets do not get attention they need/deserve
  - Other tools are used to facilitate processes
    - Developers look at Bugzilla even less
    - Reporter feels like nothing happens
    - Less transparency
    - Prospect contributors turned off

- Mass-updates prohibited
- No ticket type (—, bug, task, enhancement)
- Priority / Severity (ab)used
- Up for improvement
- No or inconsistent use of milestones
Symptoms

- Some developers barely look at ticket(s) / queue(s)
  - Tickets do not get attention they need/deserve
  - Other tools are used to facilitate processes
    - Developers look at Bugzilla even less
    - Reporter feels like nothing happens
    - Less transparency
    - Prospect contributors turned off

- Mass-updates prohibited
Symptoms

- Some developers barely look at ticket(s) / queue(s)
  - Tickets do not get attention they need/deserve
  - Other tools are used to facilitate processes
    - Developers look at Bugzilla even less
    - Reporter feels like nothing happens
    - Less transparency
    - Prospect contributors turned off

- Mass-updates prohibited
- No ticket type (—, bug, task, enhancement)
  - Priority / Severity (ab)used
  - Up for improvement
- No or inconsistent use of milestones
Products, Components, Versions, Milestones

**Symptom #1: kmail**

- "kmail" no longer supported,
- still in Bugzilla, alongside preferred "kmail2" successor,
- Open for bug submission still,
- 2595 unresolved tickets,
- New tickets are logged frequently.

**Suggested Treatment**

- Temporarily disable Bugzilla change notifications,
- Resolve all unresolved kmail tickets,
- Move all "kmail2" bugs over to "kmail" product,
- Remove "kmail2" product.
High-level Target(s)

- Bugzilla cleanup
- Formulate and propose standardized use of ticket parameters
- Align documented process and establish responsibilities
- Align use of versions/milestones with source code management
- Mass-updating tickets through sysadmin requests
- One throat to choke
Challenges Ahead

- I’m not the only one to think he’s an expert
- I’m not the only expert
Challenges Ahead

- I’m not the only one to think he’s an expert
- I’m not the only expert
- New to KDE
Challenges Ahead

- I’m not the only one to think he’s an expert
- I’m not the only expert
- New to KDE
- Consensus…
Many Good Ideas

- Separate release cycles for (core) components
  - “Frameworks?”
- Concern for (stable) distribution support
- Suggestions for branch standardization (current/next/development)
- More people to test, must be easy to test
- Standardize / Document clearly, release / branch / versioning policies
- Better CI integration, mandatory commit review, blaming
Tasks & Teams

- Release Engineering
- Quality Assurance
- Triaging
- Development Teams
Quality Assurance

“New“ Ticket(s) Workflow

- Group(s) of people that focus on a particular product,
- Attempt to qualify and enhance quality of tickets,
- Attempt to determine relevance of "old" tickets.

Release Workflow

- Group of people that test alpha/beta/rc,
- Mark their tickets as high quality,
- Developers search for these tickets,
- Hopefully they get resolved (in time for GA).

- No QA contact Bugzilla field used.
- Issues resolved do not become test-cases.
Spring Fever

Grand Bugzilla Clean-up

- Close new ticket entry against unsupported products,
- Close off all tickets with the last change earlier than $x$ years ago,
- Close off all tickets against unsupported products / versions\(^a\),
- Make this a regular process but with earlier than $y$ months.

\(^a\text{This requires a walk-through first}\)
Define Bugzilla field values

Example: Version, Milestone

- Standardize version/milestone references, for example:
  - '4.9-next' for 4.9 branch HEAD,
  - '4.9-stable' for latest (stable) release in 4.9 series,
  - '4.9.$z' for specific (stable) 4.9 series release,
  - '5.0-next' for master branch HEAD,
  - not “Git (master)” (several), “KDE 4”, “MS Windows” (kaddressbook), “CVS”, “SVN Trunk”, “enterprise” (Kontact)

- Agree on & Standardize such guidelines
Define Bugzilla field values (2)

Example: Priority, Severity, Importance

- Priority is “order in which to resolve”,
- Severity is “impact” (critical, trivial), or “cannot release with/without” (blocker),
- Importance is combined Priority and Severity (no score)
- Ticket type (bug, task, enhancement) has not been added / is not used.
(Ab)use a “future” milestone, says...

- Inclusion not yet planned for any other existing target milestone,
- No due date.

The “4.9-next” milestone, says...

- To be included in the next teeny 4.9 series release.
- Act on “4.9-next” when “4.9.$z” release is planned.
- Tickets with “4.9.$z” milestone now form “To Do Before Release” queue, nice for developers.
Versioning Policies

Example Versioning Bump Policy

- $micro$ bumps for bugfixes only (patchlevel),
- $teeny$ bumps for small feature enhancements
- $minor$ bumps for larger feature enhancements (upgrade possible),
  - may include new/changed run-time dependencies,
  - may include new/changed build dependencies,
- $major$ bumps for great feature enhancements
  - rewrite, refactor, significant changes to package structure, dependency changes

- $x.y.z$ could correspond with SCM branches,
- $x.y.z$ with Bugzilla versions & milestones,
Release early, release often

- dev, alpha, beta, rc