



How to Write Clear Development Requirements and Bug Reports

Chris Sharp, GPLS
Grace Dunbar, Equinox

Where do I start?

I want to pay for some development... what should I do?

- Remain calm
- Do you need to go to RFP? maybe...
- What is the human doing?
- Use prose, not perl
- Accommodating other users needs
- A picture says a thousand words

Contracts

- Released under the GPL
- Written to Evergreen master
- Engage developer community early and often
- Delivery dates
- Backporting to other versions?
- What about testing and bugs?
- End user docs?

Community

Don't forget to engage the community!

We all occasionally have different opinions about functionality (see: Acquisitions Workflows), but with a little open communication we can resolve them to everyone's benefit.

Anatomy of a good requirement

- Specificity
- Use cases (no edge cases)
- Workflow without “jargon”

What I am doing:

What happens now:

How I think it should be improved and why:

Use Cases

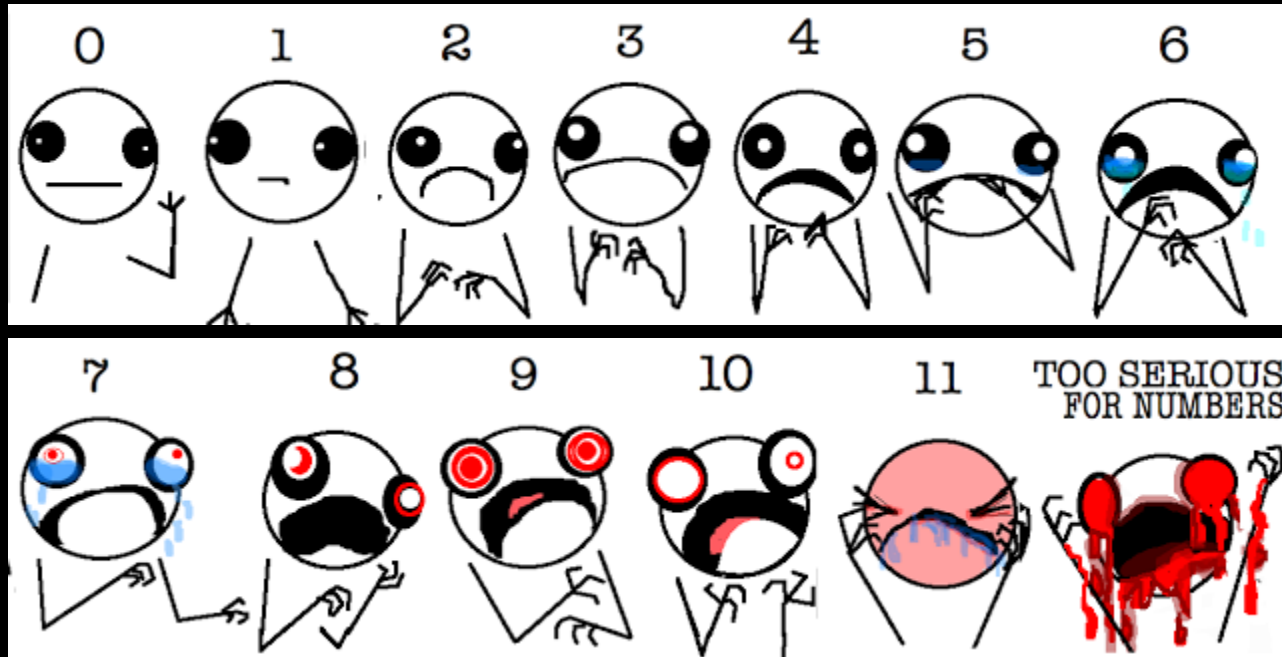
Tell me what you need to do.

Cindy Circulator needs to be able to remind patrons of local elections. There should be a mechanism by which the organization can record data about a patron's voting district. There also needs to be a mechanism for the organization to input local election dates tied to the voting districts and the patrons in those districts. Upon the "trigger" (staff mediated transaction at a workstation - i.e. not a renewal or computer reservation, etc.) the system should display a pop-up notice to staff notifying them to remind the patron of their upcoming election.

But it's obvious... right?



Draw me a picture...



Images courtesy of Hyperbole and a Half. <http://hyperboleandahalf.blogspot.com/2010/02/boyfriend-doesnt-have-ebola-probably.html>

Bugs. Scary, scary bugs.



What is a “bug”?

- a software defect that results in a reproducible error or otherwise unexpected* behavior

What causes bugs?

- typos, “thinkos” (errors in programming logic), or other implementation errors
- unexpected interaction between two apparently unrelated parts of a software program
- “bitrot” - lack of maintenance of certain parts of the program

You may **not** have a bug if...

- the behavior is the result of a suboptimal runtime environment (e.g., workstation problems/conflict with another software program, local networking problems/low bandwidth)

You may **not** have a bug if...

- you are unable to reproduce the issue with the same steps you used to elicit the behavior
- your description begins with the phrase “it would be nice if...”

PEBKAC

- “Problem Exists Between Keyboard And Chair”

Example: The PINES Way

- Help Desk Manager (filter 1)
- Subject Matter Experts (filter 2a)
- Technical Support (filter 2b)

Technical Support

- decide whether this is a “local” issue or a “system” issue
- if a system issue, decide whether it’s a configuration/environmental problem or a problem with the software
 - system logs
 - test server vs. production server

Research!

- have others reported this?
 - search Launchpad
 - Google the error message text
 - consult #evergreen IRC or the email lists

Report the Bug

- create a bug report via Launchpad
- if it **is** a duplicate, it can be marked as such in Launchpad

What Happens Then?

Two uses for the bug tracker in the Evergreen development community:

1. as a communication tool for development
2. as problem reports (similar to a Helpdesk)

Bugs as Development Tool

- developers create bug reports for found problem
- include proposed solution
- testers/other developers test the proposed solution and sign off
- the fix becomes part of the core code

Bugs as Problem Reports

- reported by those who do not have the resources to fix the issue themselves
- from the user's perspective, similar to an IT Helpdesk

Is this thing on?

- bug reports can linger...
- nearly all Evergreen Developers and Sys Admins subscribe to bugmail

But...

- you are relying on *volunteer* efforts on the part of community developers to fix your problem

Therefore...

- it **really** matters how you write the bug report if you want community interest in providing a solution

Anatomy of a Good Bug Report

- your software versions (EG, Postgres, OS, OpenSRF)
- detailed description, including error messages (copy/paste)
- detailed steps to re-create the problem

Phrases to Use

- “When I take the following steps...”
- “The following error message comes up...”
- “The end user expects...”

Things to Avoid

- writing in long paragraphs*
- telling your life's story
- critical, negative, or blaming language

EXAMPLE: Good or Bad?

We're not sure if this is a bug, or if we are just not using the system correctly. When we try to change and save the classification in holdings maintenance, it doesn't stay.

EXAMPLE: Good or Bad?

Why would an item indicate that there is a Google Preview, but not actually display one?

EXAMPLE: Good or Bad?

When an item is marked as “long overdue”, overdue fines are not refundable; accrued overdue fines that were voided at the time of payment will be reinstated and subtracted from the refunded amount.

EXAMPLE: Good or Bad?

Have the option to print labels to file.

EXAMPLE: Good or Bad?

When selecting Clear Shelf-Expired Holds from the Circulation menu or when selecting View Shelf-Expired Holds from the Browse Holds Shelf interface, the resulting list should only include items that expired before today.



Thank you!

Questions?