

Improving the Development / QA cycle

- looking at the upcoming 3.5.0 release

- ▼ Cor Nouws
- ▼ 2011-10-14
- ▼ VIP (*Volunteer In various Positions* 😊)

The reason

- ▼ 3.4.0 quality was not so great
- ▼ Doing better with 3.5.0 is important
- ▼ Of course some obvious reasons for problems with 3.4.0
- ▼ But what about 3.5.0 .. a lot happens, and can we know what we do not know ?

Improving requires understanding

- ▼ Input for this presentation taken from interesting discussions on dev-list
 - ▼ <http://lists.freedesktop.org/archives/libreoffice/2011-June/014201.html>
 - ▼ <http://lists.freedesktop.org/archives/libreoffice/2011-September/017650.html>
- ▼ We expect enough understanding and explanation (marketing) about the use of a point-zero release!

Quality means Balance



We talk about 'volume'

- ▼ Development volume
 - ▼ Development-To-Test-Output
- ▼ Quality Assurance volume
 - ▼ QA-Capacity

Development-To-Test-Output

- ▼ Is the result of more ..
 - ▼ code for new functions
 - ▼ changes for bugfixing
 - ▼ rework for cleaning, for building
 - ▼ etc etc

Number and severity of changes on code

- ▼ How many difficult, deep hidden stuff is touched in these months?
- ▼ For sure (many) nasty hidden older bugs will surface
- ▼ Surfacing and repairing of hidden problems will cause new bugs
(which of course is more or less the same as 'older bugs')

Do, or can, we know the unexpected ?

- ▼ Some things that caused problems with 3.4.0 will not happen
- ▼ But now, since months heavy rework on the code and build system is in progress
- ▼ Much needed and appreciated... however:

Obviously, we don't know the unexpected – we are in general (too) optimistic :-)

- ▼ Positive by nature
- ▼ We know what has been improved, may not know or see the new probs
- ▼ Developers may recognise this ;-)

Bugs handling by developers – a side note

- ▼ Currently devs pay more time at fixing bugs then at new features
- ▼ How much time can be consumed by one annoying bug? Two days, two weeks, ...?
 - ▼ We know that it can be very time consuming
- ▼ Fine growth off the attention for the (former) weaker spots (as Base and Windows)
- ▼ Increase in quality and use of automatic testing
- ▼ Are there other releases/tasks that need attention during the QA-preriod for major/minor a release ?
- ▼ Are people really available ? (Christmas, new year, vacations)

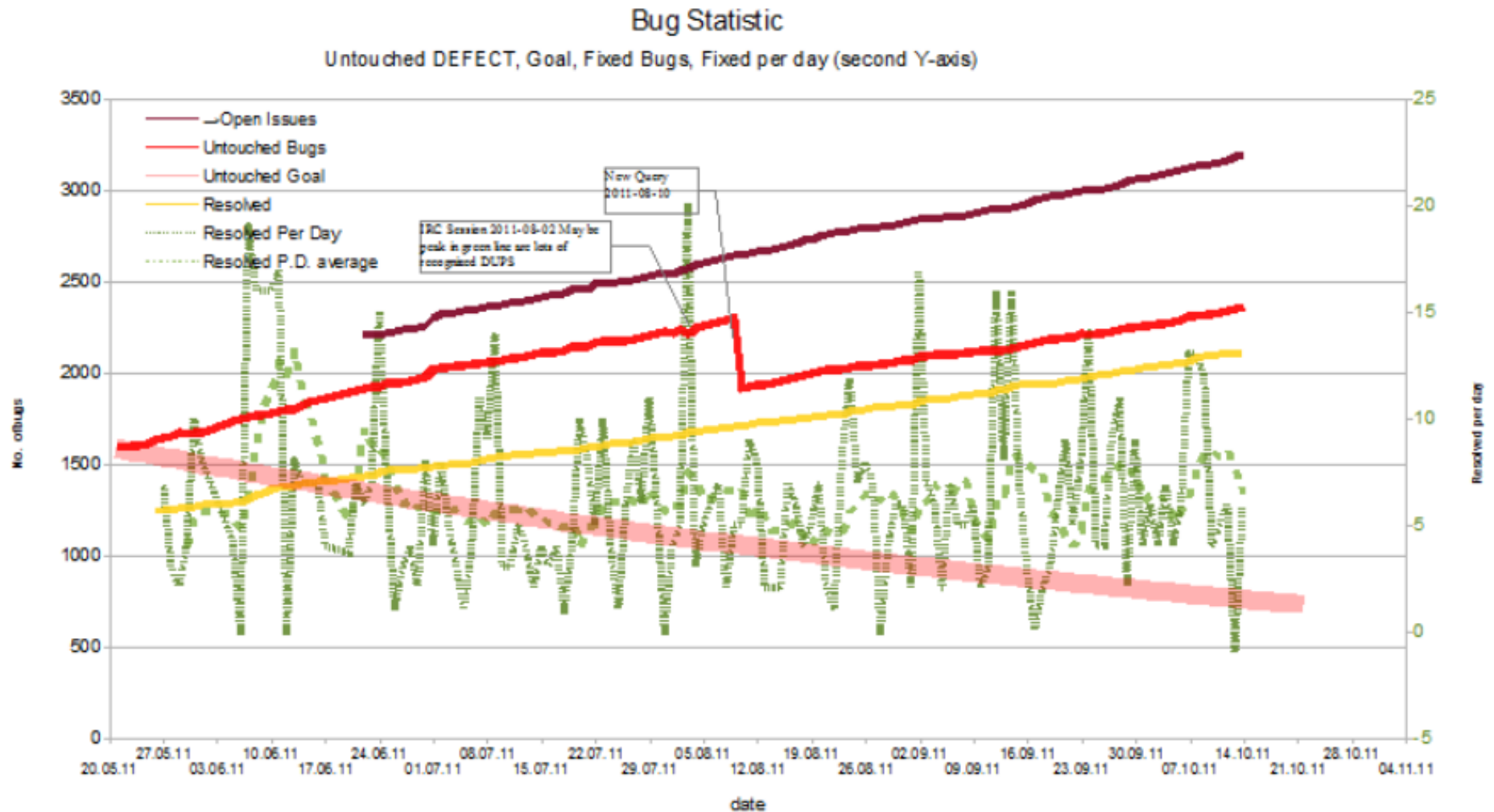
What do we know about bugs?

- ▼ How many issues do we see from QA volunteers?
- ▼ How fast are those solved?
- ▼ Statistics are not yet that good

For this discussion we are especially interested in bugs found in master / daily builds

- ▼ Some general numbers from Bugzilla in diagram on next slide
(*Some numbers in bugs against Master further on*)

Statistics from BugZilla



Link to the wiki http://wiki.documentfoundation.org/User:RBd/Workbench#TDF_QA_Activity

Testing is done on..

- ▼ Release candidates
- ▼ Beta releases
- ▼ Daily builds
 - ▼ From master, great for manual testing, available regularly

Listing is in reversed order of importance!
Testing early in the process is important

- ▼ Of course also bugs from released versions are submitted

QA-Capacity

- ▼ Number of people
 - ▼ Most of those QA volunteers
- ▼ Time they have available
- ▼ Tooling for their work

In this presentation no special emphasis on the tooling for e.g. release testing

- ▼ Info about that (and more) in the presentation of Yifan Jiang “Libreoffice QA - Regression, Feature and Smoke test”

What do we know about people involved in QA?

- ▼ How many people
 - ▼ submit new bugs regularly (say once a week or more) ?
 - ▼ work on bug triage regularly – say 3 times or more a week ?
 - ▼ work on bug triage now and then – once a week or less ?
 - ▼ join QA-parties now and then ?
- ▼ How do those numbers evolve ?
- ▼ Is the work well spread over Windows / Linux / MacOS ?

Find the QA volunteer !



Just saying that it is 'important' to have more people working on QA and that 'good' testing is needed, does not bring in more volunteers..

You can't order QA volunteers from the shelves ..!

Finding and handling bugs is time-consuming

- ▼ Most people do it beside other responsibilities, normal work, etc.
- ▼ Installing new builds, set preferences, etc takes time too
- ▼ Creating proper bug reports is often difficult too
 - ▼ Enough time to test thoroughly and report accordingly?
 - ▼ I do not always have that!
 - ▼ So some bug I find may not be reported ..
- ▼ So more people needed for testing – also good for diversity
 - ▼ do I recognise this somehow ;-)

So how is that with daily builds?

- ▼ Installing daily builds takes time
 - ▼ Be prepared for abnormal behaviour
- ▼ Do QA volunteers jump on new daily builds often?
 - ▼ Not reasonable to expect that, IMO
 - ▼ Happily, quality of daily builds reasonable- good to work with when you take care
 - ▼ Availability is alas not always OK
- ▼ It is known from e.g. Mozilla that releasing often, results in a firm workload for the QA community

What about bug reports against daily builds?

- ▼ Are reported in Bugzilla as for version 'LibO Master' or 'LibO 3.4 Daily'
- ▼ Is the amount of bugs being reported high (90 in 3,5 months)
 - ▼ Are there not many bugs?
 - ▼ Or too little testing or not enough complicated testing?
- ▼ See for some numbers from BugZilla > next slide

Some numbers from BugZilla

▼ Unique valid bugs against 'LibO 3.4 Daily' and 'LibOMaster'

| | Opened (unique, valid) | Fixed (unique) | Still open |
|----------------------------------|------------------------|----------------|------------|
| June 2011 | 3 | 1 | 2 |
| July | 22 | 5 | 17 |
| August | 34 | 16 | 18 |
| September | 19 | 2 | 17 |
| October | 13 | 4 | 9 |
| Totals | 91 | 28 | 63 |
| <i>data 2011-10-16 20:00 UTC</i> | | | |

Summary – Improving the Dev-QA cycle for 3.5.0 ?

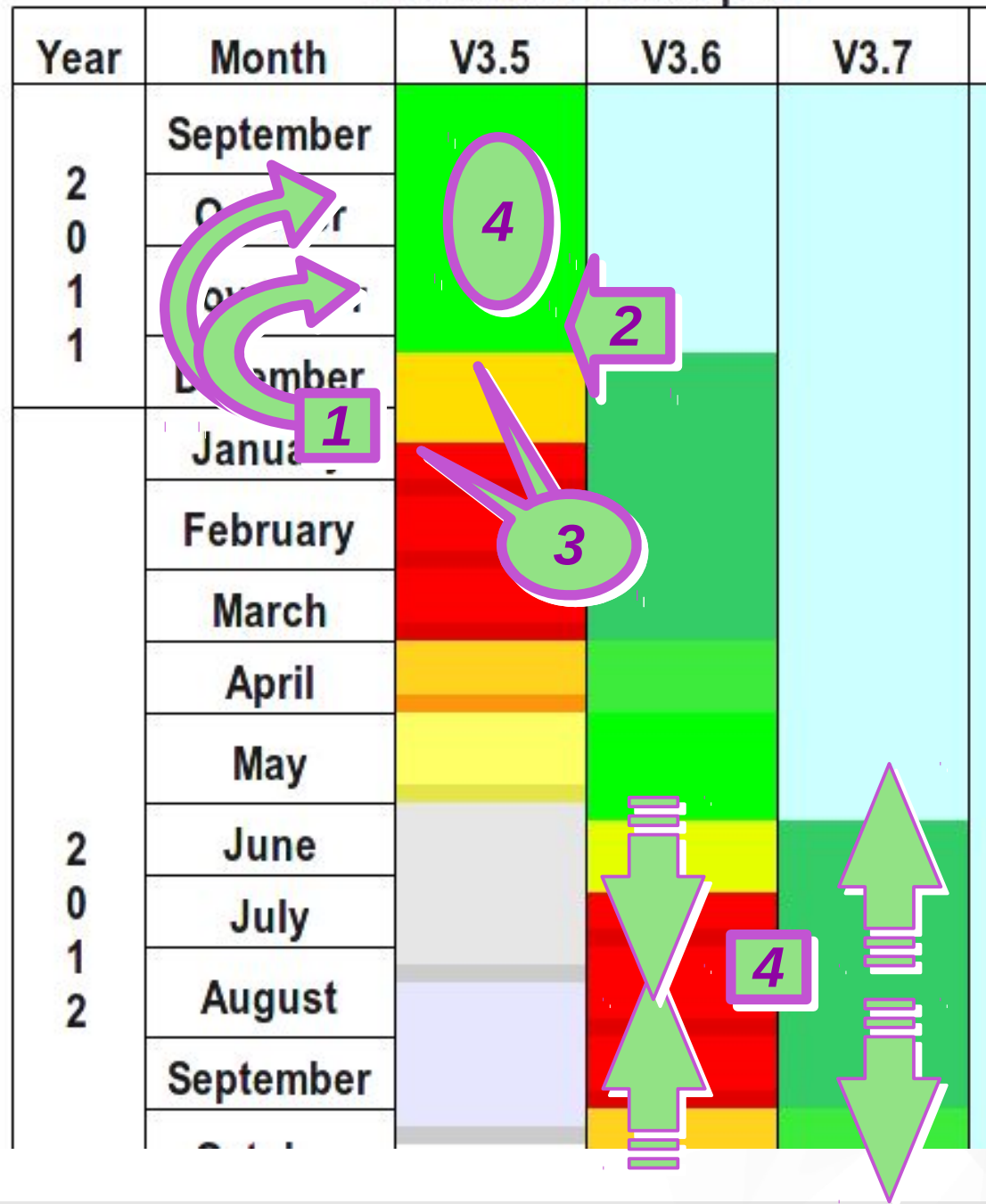
- ▼ Development-To-Test-Output
 - ▼ Quite some unknown stuff in it
- ▼ QA-Capacity
 - ▼ There is good activity, but currently not clear how many people, how much is covered
- ▼ Testing on daily builds
 - ▼ Does happen, but again, not exact picture of situation
- ▼ Bugs reported against master / daily builds
 - ▼ number not heigh? due to too little test or good quality?
 - ▼ number of fixes of those currently less then 50%

So ... what possibilities do we have now?

1. Try to get more testers doing the testing earlier
2. A 'quiet period' , pre-official freeze on master
3. Two weeks time frame after first beta and first RC
4. Try faster fixing of issues in master

(see the numbers on the next slide)

LibreOffice release plan



original diagram from
presentation 'LibreOffice
Release Process' by
Petr Mljackek

1 -Try to get more testers doing the testing earlier

- ▼ Interest more people in testing daily builds
 - ▼ marketing:
 - ▼ make features known, explain how it works, and why ..
- ▼ Try to get more people involved in first beta (right after feature freeze) instead of with first RC
 - ▼ E.g. with a competition
 - ▼ prize for the top-5 (clear, useful) issue submitting testers ?
 - ▼ Or an 48 hrs event
- ▼ Prepare good support for the testers
- ▼ Specific items / requests for the native-lang communities
 - ▼ but tests that do not require a too short timeframe

2 - A 'quiet period', pre-official freeze on master

- ▼ Can be done around an earlier release with new features
- ▼ A self-imposed elevated attention to committed breakages
- ▼ Emphasis on bug fixing
- ▼ Will result in better first beta, so jump start for good release quality

3 - Two weeks time frame after first beta & first RC

- ▼ Could give more bugs in one run to fix for next build..
 - ▼ But could also lead to less attention from developers?
- ▼ Could be more relaxed for testers..
 - ▼ Do people continue testing when they found a bug?
 - ▼ next build might include fix and is soon
 - ▼ or they might have to wait, when there is a severe bug..
 - ▼ Not mandatory to restart testing with a new version
 - ▼ continue with previous version is possible
 - ▼ and no problem (..) if enough people do switch..
 - ▼ start all the test cases again

With current info, no clear conclusion possible

4 - Faster fixing of issues in master

- ▼ Looking at data available, that looks reasonable
- ▼ Try to collect data about that
 - ▼ do developers have enough time
 - ▼ do they have time left for development work

Smarter finding / handling / fixing bugs should result in less time spent on bugs and more on features

And the winner is ... (what do we choose) ?

- ▼ Try to get more testers doing the testing earlier ?
 - Yes !
- ▼ A 'quiet period' , pre-official freeze on master ?
 - Make sure masterbuild is good week before beta !
- ▼ Two weeks time frame after first beta and first RC ?
 - No evidence for that now !
 - But track data from Bugzilla and from Devs
- ▼ Try faster fixing of issues in master ?
 - Yes !



PARIS2011 CONFERENCE

12th-15th October

Thanks!



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).