Developer Tasks for Non-Code Hackers

- André Schnabel, Christoph Noack
- 2011-10-15, 10:30, La Cantine
The Observation

- User oriented community feels separated from developers
- “Developers are pushing code without asking us.”
- “Developers don't take care about documentation.”
- “They just do what they want and they do it very fast.”
- Developers (as other people) **focus** on their main tasks
  - hacking code
  - hacking code
  - hacking code
  - (You may change the order)
- But no developer wants to deliver features that nobody uses
- So developers **are** interested in more than hacking code
The Idea

- Take an easy hack with visible effect for end-user
- Play the role of a developer and see where other people can help
- Go through steps of feature implementation
  - Evaluation and analysis
  - Refine and create prototypes
  - Implement and submit
  - Document
  - Test
- Document what has been done
The Self-Experiment

- Bug 30800: grid lines disappeared in Calc for cells with background color
- Introduce option for grid line display on cells with background colors
- Evaluation and analysis
  - What would be the best way from a user's perspective to implement the option
  - Play with option dialog, analyze usage tracking, interview Calc users
  - ~1.5 days of list communication (design, ux-advise, german users) three preferred ways discussed.
- Surely no need to have a developer deeply involved

The Self-Experiment

- Refine and create prototypes
- Hacking code (1.5 days including learning phase)
- Used to document current status find possible layout problems, wording ...
The Self-Experiment

- Implement and submit
  - ~1 day final implementation and cleanup
  - Very friendly feedback, review and corrections at the developer list
- Test
  - No big effort
  - Done by people cc'ed to the issue in daily build
- Feedback some weeks later
  - My change triggered a bug in column header display in calc
  - Bug was fixed by Calc developers
  - They even excused to have my bug fixed without notifying me
The Findings

- ~ 4 work days spent
- 15% coding
- 25% learning and experimenting
- 60% discussion and documentation
- >50% could be done by non-code hackers, but those
  - Are often not aware what is needed and helpful
  - Underestimate the value of their contributions
  - Don't know ways and places to hook in and start collaboration
  - Speak not (yet?) the developer's language (tech)

We need to improve on that!
Contribute on the UX Advise Mailing List

Subject
- Meeting ground for devs and UX concerned people
- Quick and painless feedback for “How do users need that?”

Proposed Tasks
- Quick-check proposals by devs and improve features
- Resolve User Experience questions (no personal opinions!)
- Summarize specific info (e.g. HIGs, OOo specifications)

Benefit
- Features will “just work” when released initially
- Lots of contacts and knowledge about the product

More at http://wiki.documentfoundation.org/Design#Communication
Improve Cross-Team Collaboration

Subject
- Currently only loosely connection between devs and other teams

Proposed Tasks
- Read commit logs, help to improve
- Follow the dev mailing list
- Ping other teams if required

Benefit
- Improved cross-team collaboration
- Devs can focus on hacking code

More at http://sweetshark.livejournal.com/4478.html (Blog posting by Björn)
Maintain Information on Design Whiteboards

Subject

- Space for serious ideas and real dev work
- Provides a bigger pictures for all

Proposed Tasks

- Collect and maintain information (e.g. use cases, issues, DOC hints)
- Conserve mailing list discussions

Benefit

- Makes bigger feature changes manageable
- Be the maintainer for selected functionality

Thank you …

- … for improving LibreOffice!
- … for attending Kendy's upcoming talk “Polish LibreOffice's rough edges”