

2nd Calligra Sprint

Eating Our Own Dog Food

Jarosław Staniek, November 2011

Eat Why?

Calligra 2.4 



- Sends clear message: this software is useful
- Testing by fellow contributors is valuable
- Generates usage scenarios and then requirements

Eat Why? cont'd

- Brings cross-app integration ideas
- Helps to avoid feature duplication
- If right tool picked, development process helps to update requirements

Eat Why? contd



- Team building!
- Easier to understand and acknowledge differences between apps
- Identify competences among contributors

Use Where?

(3 areas)

1. Reusing features of one app in other apps (instead of reinventing)

→ Target: Calligra developers/designers

Use Where? contd

2. Using our apps in the development process

→ Target:
Calligra contributors

Use Where? contd

3. Using our apps elsewhere, in activities not related to Calligra

→ Target: Calligra contributors and advocates

Eat What?



Use Calligra Tables for tabular data

→ Status: used for some ods files

→ Action point: identify problems like usability/performance

Eat What? contd



Use Calligra Plan for project management

→ Status: some contributors use it

→ Action point: collect best practices

Eat What? contd



Use Kexi for various data needs

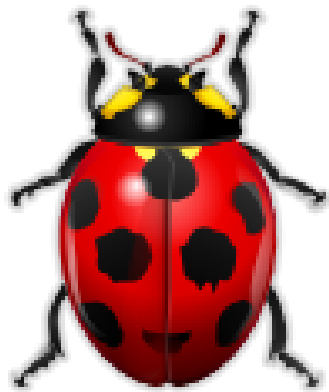
→ Already good for storing and simple queries

→ Not yet good for analyzing data

How to use Kexi?

→ Only simple
relational features

→ Status: not used,
let's start!



([discussed with Inge](#))

Idea: Kexi as Bugzilla client

- Use Bugzilla's web services for this
- Separate plugin?
- Online/offline operations with sync.

Idea: Kexi as Bugzilla client contd

→ Use Kexi to
maintain data for
automatic
changelogs
(server db preferred)

How to use Kexi? contd

Action point:

Provide use cases,
e.g. CSV
import/export

How to use Kexi? contd



Action point: Provide server for shared db

- Some of public access
- Some only for contributors

Eat How?

Provide “Best practices for own dog food consumers”

Best practices

Keep separate setup
of stable Calligra
apps

→ stable for using

→ unstable for
development

Best practices

contd

Developers becomes
users too

→ This leads to more
valuable use cases

Thank
You