

GT2003Bertrand

Bertrand Delacretaz

Lightweight tools for successful projects: the Open Source Way

How do we organise ourselves as an OpenSource project?

Independant consultant working in Swizerland

Intro: What are we talking about

"A loose community can be successfull given the right tools." Impressed with how good the cocoon community works

We were able to build very reliable software at regular intervals. A lot of power in the community.

He's going to talk about tools for open source programming. The project management tools, because he found that not everyone knows how to use those tools to coordinate their own internal teams. not even only for computer projects.

Tools:

He's speaking about the concepts about these tools, doesn't really matter what tool you actually use as long as they have similar functionality.

- Bugzilla
- CVS
- Weblogs

He is using [YAPT](#), a cocoon project, for his slide show.

Succesful projects

Lots of times where a project fails is because lack of efficient communication. Stories help you get up to speed. We need stories in our projects to have new people joining.

Email, post-it notes and documentation are normally used for communication. Lack of docs makes it failing. You need self-service information tools.

Project Questions

- Who did that?
- Who knows what?
- Who usually does what?
- Who did what, precisely?
- Why is this Feature here?
- What happened while I was away?
- Wasn't this said before?
- Who made this comment on this topic

CVS (Code Versioning System)

- You can't work today without a versioning system.
- Much information can be had looking at difference or history of files
 - History of changes
 - Who wrote what when?
 - Visual differences

Bugzilla (bug/issue tracking system)

- He uses bugzilla also for design discussions.
- Much more structured information than email.
- You can assign priorities.
- Why did we do it this way?
- Who reported a problem
- Assign priorities

WIKI

- Public whiteboard
- Table of contents

- Pointers to other pieces of information
- Keeping it alive is very hard

The cocoon wiki is becoming a goldmine of information.

WebLogs

- Project weblogs
- Project milestones
- Helps keeping history of the project
- Reminds the community of their progress
- Very low-tech and friendly to low-tech people
- Helps put the project in perspective
- very useful for team coordination

Mailing Lists

- Without an archive they are useless
- Threaded broadcasts.
- Classified in threads
- It can be hard to find information in a mailing list.
- Starting with mail-list based discussions for people that are not used to it can be hard.
- Need people who have good list discipline

In:

- questions, answers, thoughts, noise

Out:

- knowledge, decisions, noise

What does CVS bring to our project?

- Traceability
- Tells the story of the code
- Why did something stop working?
- You can get back in time and recreate a previous state of the system
- Reliable information
- Changelog
- Blame info, who committed what broke us?
- Mail from CVS gives a good heads up on what is happening

In:

- code changes
- change logs

Out:

- change events
- change logs
- differences
- blame info

Bugzilla

- Desert Island tool
- traceable - who said what when and why
- needs discipline
- could tell the story of a project in detail
- workflow: you can dispatch tasks to people
- What is urgent
- What can I do to help?
- Critical path: why is this not working now?
- Dependencies between tasks (might also reflect code dependencies)
- Good for building a knowledge-base, organically

In:

- new tasks
- work orders
- solutions
- dependencies

Out:

- task lists
- task histories
- critical paths
- solutions

Wiki

- Unstructured, shared information
- unprotected (everybody can edit)
- traceable (you can go back in case of abuse, the mail list gets a message of the changes so there is oversight)
- brought more advantages than disadvantages to the Cocoon team
- Wiki empowers Users

In:

- content
- links
- attachments

Out:

- content
- links
- attachments

Project Weblog

- Condensed news and public information.
- Official information what goes on in a project
- Notification via RSS feeds
- (Use example): Asks his students for 3P record progress, problems and perspectives

Desert Island Tools

The things Bertrand can't go without:

- CVS - cannot drive nails without a hammer
- Bugzilla - he uses it in a small project team instead of a mailing list
- Mailing lists - they work but require lots of discipline (it works well for apache communities, but for newbies it takes a while)

How can we improve?

Very little computer-specific tools - these tools could also be used in other projects Difference between building actual homes and software projects: why does a complex project as building a house most of the time doesn't fail like SW projects do?

He's getting more specific on cocoon now. what we do and what we can improve.

How are we doing in The Cocoon Community?

The good

- who knows what is very easy to ask in cocoon
 - mail list + cvs commits
- who did what is very easy as well (CVS)
- doing ok on what happened when I was away
 - *thru mail list, but going thru 3000 messages takes a lot of time.. but doable
- Who gave which comment on what topic?
 - Works so-so, we could use better search/classification tools on mail-list
- Was'nt this said before?

There are two questions hard to answer:

- how are we doing?
- when will things be ready?
- he suggests to use more bugzilla and its dependencies to show a better picture and be more precise on scheduling.
- we have to bear in mind that people are not required to work on Cocoon - there's a different commitment (time-wise)

7/9 questions have good answers in the cocoon community.

Possible improvements

- More monday morning staff meetings (NOOOOOT!!!)
- Shared test scenarios
- Testing, not enough
- Project weblog (better change notes, syndicated information what happened since last release)
- More synchronous communication (instant messaging, chat, virtual?? pair programming? ??)
- Virtual Hackathons!

The RT effect (brainstorming)

Everyone can send an RT to cocoon, often wild ideas, triggers other people's imaginations in unexpected ways, help us get to know each other better. What are your wild dreams and ideas. Improve communities, we get to know other visions and dreams.

Careful about written communication

- 5% goes thru, so be nice, it's hard to convey your information when too much emotions are going thru.
- Assume you are wrong
- Assume you did not understand
- Ask for clarification - before flaming
- Be patient
- Think how you would solve it if you were sitting around the same table

The "two email pattern": when you are angry, write a nasty email once, then throw it away and start over. You'll find your emotions gone and information would be much easier to convey.

Conclusions

These are great tools and are very important for your project. For the cocoon team is hard to answer the "when" question but we should make an effort to improve this (do we need more bugzilla? we should also discuss the way we use bugzilla)

Remember the questions!

This was a very good meta-talk: talk not about the project itself but about how the project and the community is working and managed.

(Notes taken by Stefano Mazzocchi, Rogier Peters, Steven Noels and Jeremy Quinn)