

What we learned

by moving Scout to Open Source

Matthias Zimmermann, Co-Lead Eclipse Scout
BSI Business Systems Integration AG

How we got into Open Source

Asking for Permission

Many years ago

1999 A small Swiss company built its own app framework

2007 Company grows to 100 employee, Scout still there

2009 After visiting the **Eclipsecon**

- Andreas teamed up with Alex and Mischa
- “Let’s open source Scout”

And I wanted to be part of it ...

Lots of Enthusiasm

- “Open Source is cool”
- It will help recruiting
- ... more projects
- ... more money

Let's convince the board!

Enthusiasm cooling down

- “Many things are cool”
- ... doing pretty good with recruiting
- ... we are tight with resources anyway
- It will COST money
 - Conference visits
 - Answer forum requests
 - Fix bugs we are not interested in

Enthusiasm challenged seriously

What if ...

- We go out of business because we are sued?
- Competition “proves” bad code quality?
- Employees will leave the company with Scout?
- We lose the trust of existing customers?

Many grave questions, really ...

But ...

If WE don't go open source, then...

- Someone else open sources **X**
- **X** gets popular
- And customers will ask: “Why don't you use **X**?”

Ok, let's discuss

- Clarify legal issues, get expert help if necessary
- Present risks and opportunities to the board
- Provide a marketing plan
- And a business plan

5 Months later

- Addressed all concerns to the satisfaction of the board
- Including crucial support from the Eclipse foundation
- “Now you go and to talk to our customers”
- Because: “We only do this if customers are positive about it”

“And if only one of our customers should be against open sourcing Scout, the plan will be dead ...”

Additional 3 Months later

- Customer concerns addressed
 - *“Competitors must not get any insight in our software!”*
 - *“What EXACTELY is going open source?”*
 - *“Can anybody then change our software?”*
 - *“How can you guarantee stability?”*
 - *“How will you prevent chaos?”*

And again 4 Months later

- Gone through project creation review at Eclipse
- Eclipse Creation Review successful 😊
- Needed **11 Months** to get the go
 - Board
 - Customers
 - Foundation

And the work could finally start!

Since then ...

2010 Scout code in Eclipse repository

2011 First time in Eclipse release train

- ... start to build a community

- ... go to conferences

- ... organize events

- ... write documentation

- ... maintain forum

- ... acquire Scout customers

Where are we now?

- 170'000 Scout downloads in first 3 months with Luna
- Talking with customers, colleagues, and other people
- Reading about best practices to grow community
- So much good advice

Where are we now?

- 170'000 Scout downloads in first 3 months with Luna
- Talking with customers, colleagues, and other people
- Reading about best practices to grow community
- So much good advice
- Following all this advice is just not feasible ...

Where are we now?

- 170'000 Scout downloads in first 3 months with Luna
 - Talking with customers, colleagues, and other people
 - Reading about best practices to grow communtiy
 - So much good advice
 - Following all this advice is just not feasible ...
-
- To the rescue came Master BERKUS

Where are we now?

- 170'000 Scout downloads in first 3 months with Luna
 - Talking with customers, colleagues, and other people
 - Reading about best practices to grow community
 - So much good advice
 - Following all this advice is just not feasible ...
-
- To the rescue came Master BERKUS
 - Because, you see, WITHOUT a community, it's ...

Where are we now?

- 170'000 Scout downloads in first 3 months with Luna
- Talking with customers, colleagues, and other people
- Reading about best practices to grow community
- So much good advice
- Following all this advice is just not feasible ...

- To the rescue came Master BERKUS
- Because, you see, WITHOUT a community, it's ...
- JUST SO MUCH LESS WORK



The Evil of Communities



<http://www.lapastillaroja.net/MT-archives/images/TS-5502.pdf>

**If Only There Were Some Way to
Rid Yourself of the Community
Menace ...**

Rule #1

Depend as much as possible on difficult tools

Best Practices

- weird **build systems** not found anywhere
- proprietary **version control system**
- **issue trackers** with limited numbers of licenses

What we have

- Maven/Tycho
- Git
- Bugzilla

Rule #2

Encourage poisonous people

Best Practices

- argue with these people at length
- **denounce them** in the forum
- **ban them** from the community
- let them in again, and **iterate** ...

What we have

- No poisonous people
- Constructive comments in forum
- Even when things are not ok

Rule #3

Provide no documentation

Best Practices

- nothing useful about code
- nothing about building
- patch submission
- release process

What we have

- very few comments in sources
- documentation badly organized
- wiki tutorials, a book, information in forum
- there is a contribution setup wiki
- tons of info about eclipse release train

Rule #4

Project decides in closed-door meetings

Best Practices

- hold meetings in person at the corporate headquarters
- **if unavoidable**: online meetings
 - with very short notice
 - inconvenient time zone

What we have

- in-person meetings at headquarter
- no public online meetings

Rule #5

Employ large amounts of legalese

Best Practices

- involve complex contributor **agreements**
- difficult non-disclosure agreements, trademark **licenses**, ...
- content should **change frequently** without notice

What we have

- Eclipse CLA
- Not so many other documents

Rule #6

Carefully choose the community liaison

Best Practices

- somebody who doesn't like people
- a person that doesn't know the technology
- go with the busiest person
 - do not remove any of the previous responsibilities

What we have

- 2 co-leads (1x technical, 1x organizational)
- open source is now my only responsibility

Rule #7

Governance obfuscation

Best Practices

- learn from the United Nations
- keep the decision-making powers unclear

What we have

- no project governance document
- Eclipse guidelines

Rule #8

Screw around with licensing

Best Practices

- changing licences makes community go elsewhere
- only talk about switching to some other licence

What we have

- EPL

Rule #9

No committers from outside. Ever.

Best Practices

- only employees may have commit rights
- rule must be undocumented
- respond along :“legal issues, we're working on it”

What we have

- so far, no outside committer candidates
- outside contributions are starting
- Eclipse says “no rules to exclude any potential contributors ”

Rule #10

Silence

Best Practices

- do not answer questions
- do not say anything
- most effective **community destroyer** of them all

What we have

- patchy social media presence
- we talk at events
- we answer questions on forum, email, ...

BERKUS Rating for Eclipse Scout

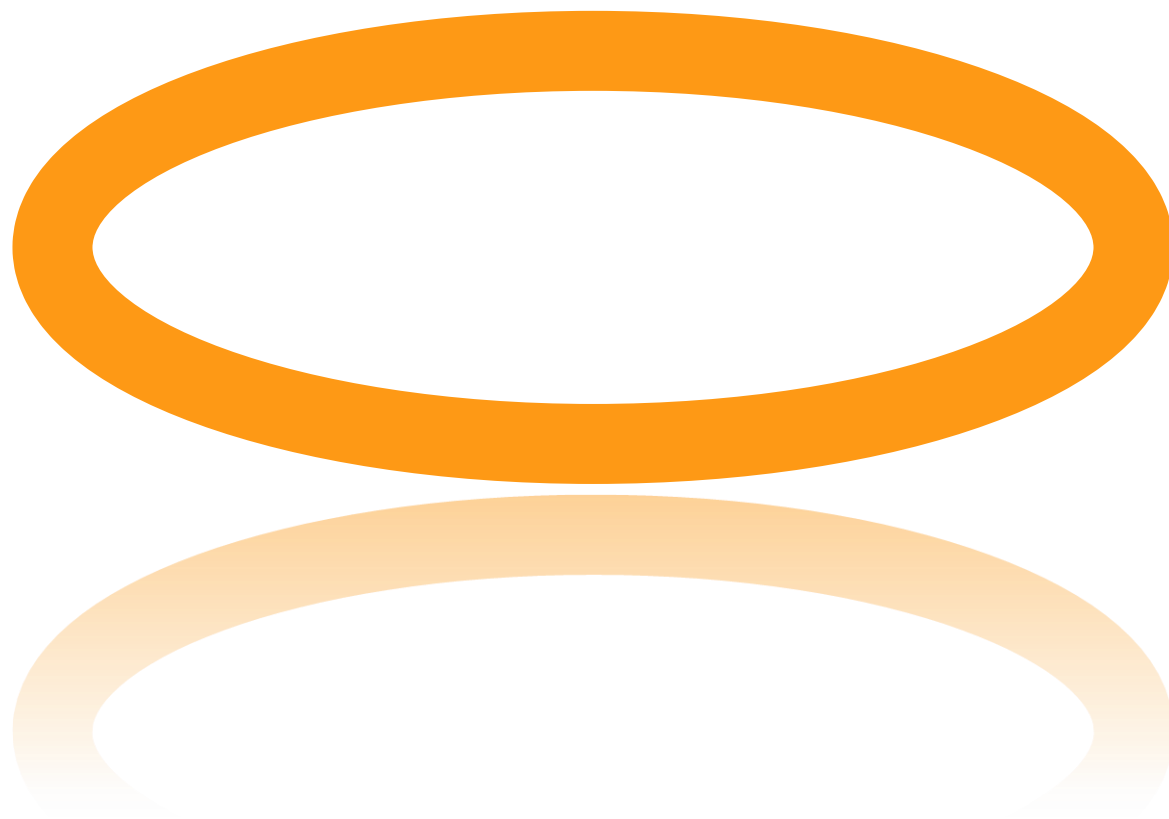
The 10 Rules

- Depend as much as possible on difficult tools
- Encourage poisonous people
- Provide no documentation
- Project decides in closed-door meetings
- Employ large amounts of legalese
- Carefully choose the community liaison
- Governance obfuscation
- Screw around with licensing
- No committers from outside. Ever.
- Silence



BERKUS becomes ...

Josh



Josh's 10 Ways

1. Familiar tools
2. Discourage poisonous people
3. Document everything
4. Accessible online meetings
5. Minimize legalese
6. Expert liaison
7. Governance simplification
8. Treat licenses with respect
9. Promote outside committers
10. Communicate

Josh's Assessment

Eclipse Scout is good at

- Minimize legalese
- Treat license with respect
- Answering questions in forum
- Beginner tutorials

Areas to improve

- Communication
- Online meetings
- Project governance
- Contribution process

Costs and Benefits

Is it worth the trouble?

Costs

Costs

→ Marketing

- Sponsoring of conferences/events
- Sending people there
- Homepage, Fact Sheet, Press releases
- Writing Articles/Blog posts

Costs

→ Documentation

- Tutorials
- Wiki
- Book
- Forum
- Maintenance (do not under estimate this)

Costs

→ **Product complexity**

- Consists of open source and internal parts
- Requires more complex build infrastructure

Costs

→ Release train

- Additional builds
- Additional testing
- Updating external dependencies
- Support of platforms not relevant for current customers

Benefits

Benefits

→ **Bottom Line**

- Open source based business (slowly) growing

Benefits

→ Sales

- Benefit for our commercial products
- Open source is seen as an asset by internal IT
- “What is Scout?”, “Why not use X for CRM?” go away
- Project hosted at Eclipse implies stability and quality
- Lowers risk of choosing BSI as a supplier

Benefits

→ Human Resources

- Helped to win additional employees
- Seen as a plus for recruiting

Benefits

→ Better documentation

- Builds trust during acquisition of commercial projects
- Helps new employees to be more efficient, faster

Benefits

→ Release train process

- Predictability of release schedule
- Stability improvements for internal projects
- Consequences of being part of the Eclipse release train

Wrap up

Lessons Learned

- Open sourcing existing software takes time
- Growing initial community and customer base took years
- Requires resources (1.5 FTE, no less)
- Helped to improve code stability
- Resulted in documentation
- Helps to sell commercial products on top of Scout
- Helps acquiring and holding good people
- Started our open source based business
- Was a good (strategic) decision

Do it yourself!

- Listen to people that have done this before
- Talk to these people
- Consider Eclipse as project home
- Start with a lot of enthusiasm
- Don't give up too soon

Thanks

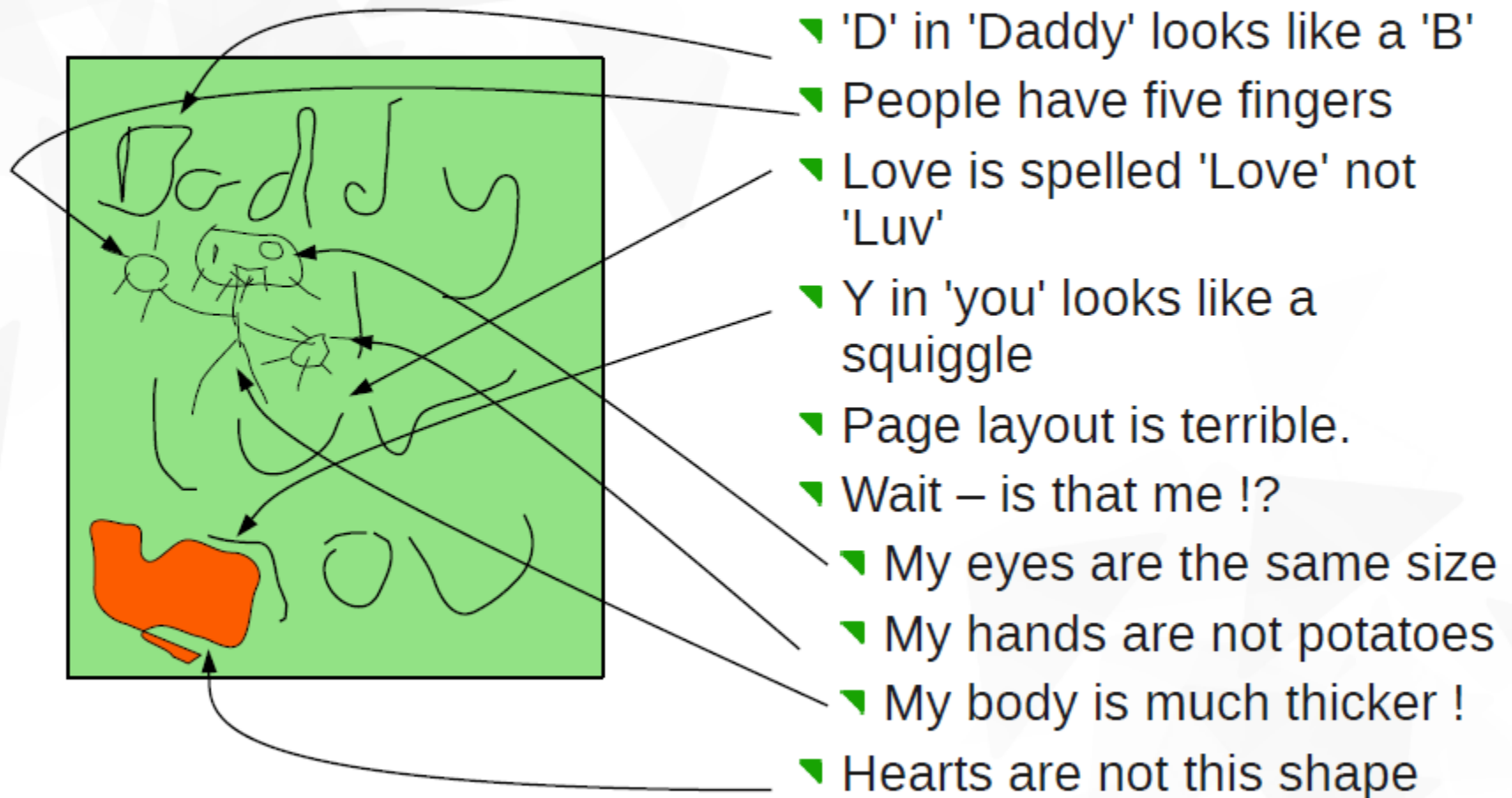
“Interacting with new Developers”

Michael Meeks, LibreOffice



A new developers Contribution

Example one: how not to do it ...



Perhaps a more winsome approach ...



- ▼ Wow !
 - ▼ You love me :-)
 - ▼ I love you too ...
- ▼ Thank you so much for making that for me
 - ▼ I'd really like to meet up and practice some of the hard letters sometime
- ▼ I particularly like the smile
- ▼ I've pushed it with a few minor corrections to master
- ▼ Thanks again & looking forward to your next piece.