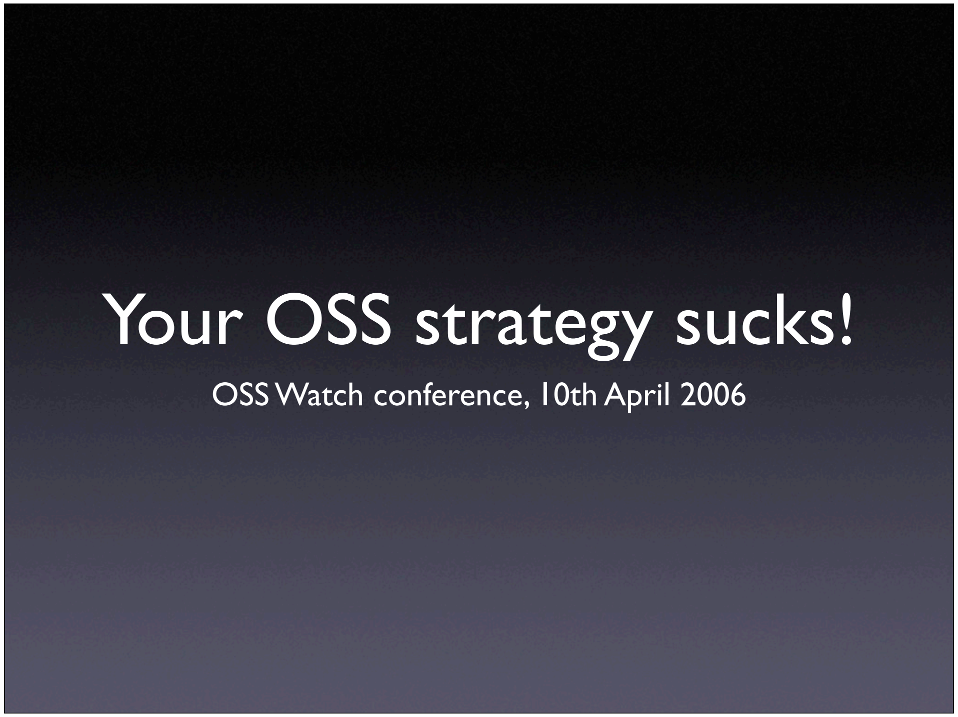


Your OSS strategy sucks!

OSS Watch conference, 10th April 2006

Based on my experiences of seeing FOSS rolled out and used in institutions.

This will be a light-hearted look at the strengths, weaknesses, opportunities and threats of using FOSS.

Institutional information strategy?
Open Source?

Most Open Source strategy sucks!

An alternative, irreverent title for this talk might be “Most open source strategy sucks!”. It’s tempting to think FOSS can just be dropped in place instead of existing commercial off-the-shelf software: unfortunately this is not true, as we shall see.

It will hopefully challenge assumptions and provide food for thought.
It will be delivered in a very Open Source style - informal, perhaps chaotic, and hopefully will be more effective for it.

- Adopting Open Source
- Traps and pitfalls
- Open Source software adoption checklist
- Open Source software adoption strategy

These are the things that need to be considered when using Open Source.
Firstly, why would you adopt FOSS?
Common mistakes in using FOSS
Things to look for when adopting FOSS
How to make a strategy.

Adopting Open Source

(and getting the most out of it)

What are the problems with proprietary software?
What would make you want to adopt OSS?
The burdens of modern software.

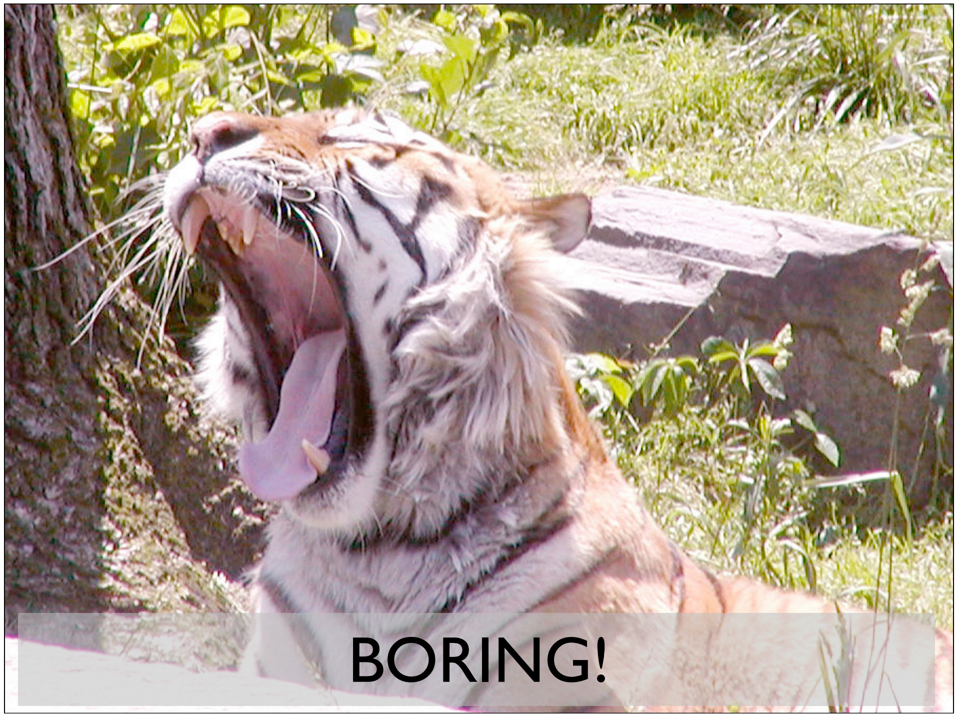
License costs?

FEDERAL RESERVE NOTE
THE UNITED STATES OF AMERICA
ONE DOLLAR

Fed up with paying license costs?
COTS can be expensive. It’s a recurring expense. You are buying a license to use the software, not the software itself.

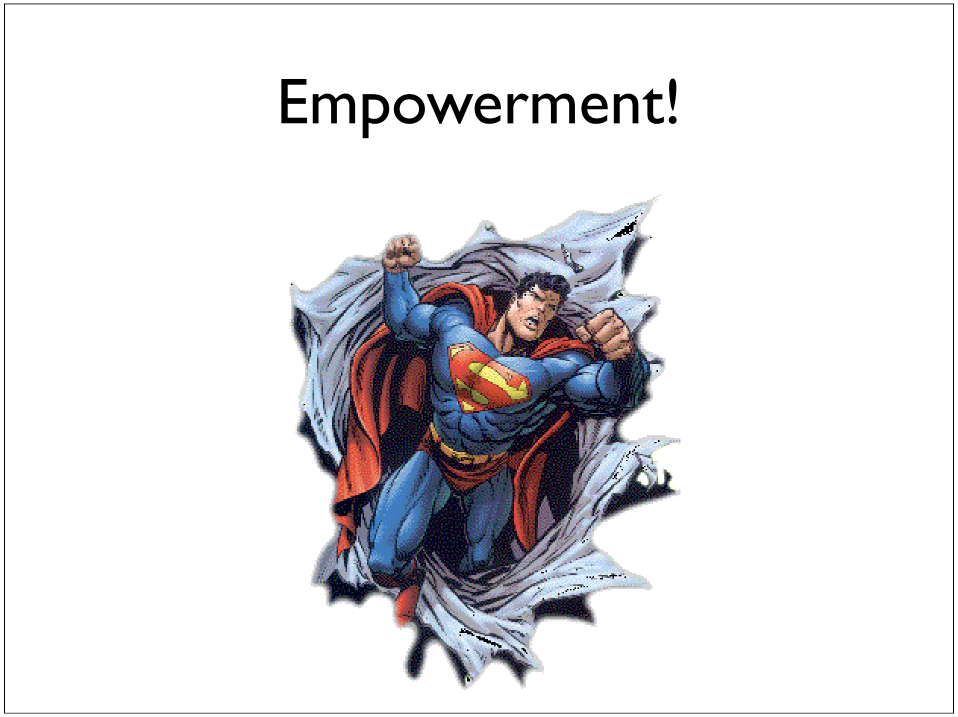
License Management?

License management a challenge?
With COTS, the burden is on you to prove you own the software.



Dealing with licensing issues can be hard work.

Vendors have a vested interest in keeping you using their software. It can be difficult to move from one product to another. Vendor software is often an all-or-nothing deal, and the price is HIGH.
Can you imagine buying a car and being told you can only drive on the vendor's roads?
Or being told you can only cook WidgetCo Dinner in the WidgetCo oven you just bought?



Empowerment!

Open Source is about empowerment. Taking control of your IT situation.

Solution ownership

Who owns the software you use to solve problems?
Are you renting a solution temporarily?
Do you have the CODE?

Data ownership

Who owns your data?
Can you easily migrate your content from one platform to another?

Updates/Patches/Decommission

Who decides when patches, updates and decommission takes place?

Back to business!

Let’s get back to business: let’s revert back to the roots of the supplier/customer relationship.

- Open Source encourages higher commitment from suppliers.
- Ensures competition works throughout the cycle.
- Putting the customer first (again).

“What do you want to achieve or avoid?
The answers to this question are objectives.
How will you go about achieving your desired results?
The answer to this you can call *strategy*.”

- William E. Rothschild

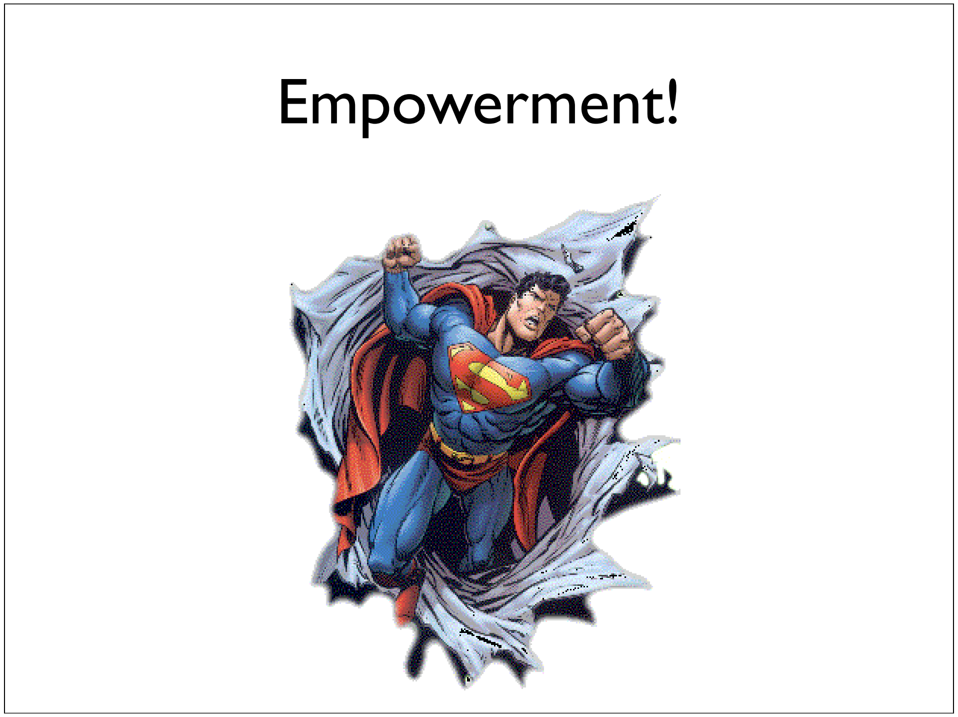
Objectives

Work out what you want to achieve or avoid by adopting open source.

License costs? Lock-in? Solution ownership? Data ownership? Control over the software lifecycle?

Solutions

Identify the solutions that might meet your needs: proprietary or OSS?



Empowerment!

This is about taking control of the business of your software. Empowering you, not the vendor.

Return on investment, ultimately

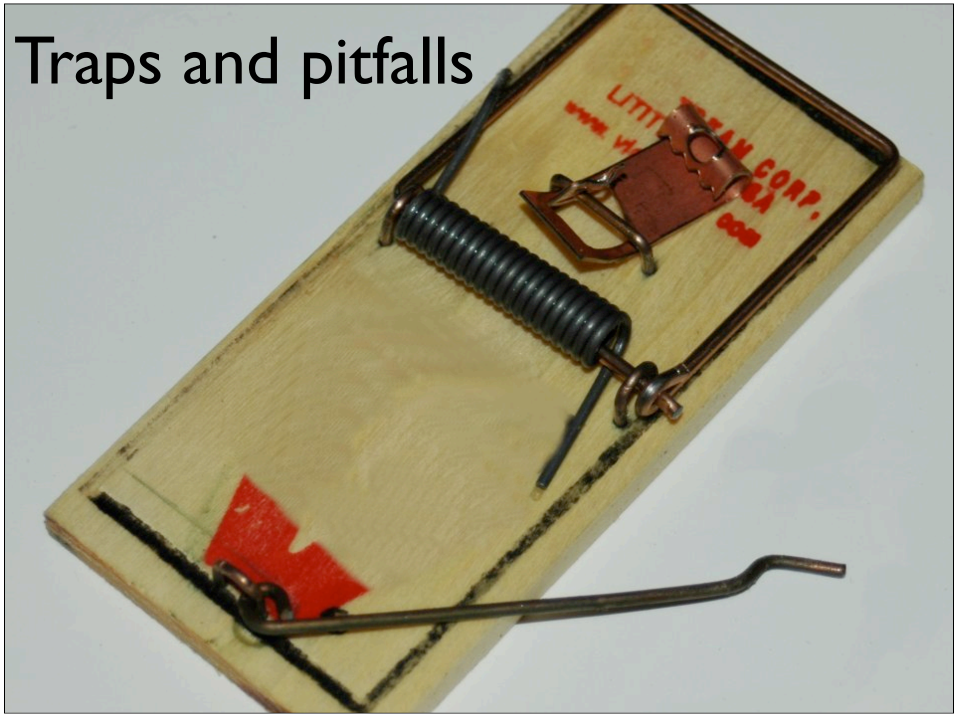
FEDERAL RESERVE NOTE
THE UNITED STATES OF AMERICA
THIS NOTE IS LEGAL TENDER FOR ALL DEBTS, PUBLIC AND PRIVATE
C 17853312 D
WASHINGTON, D.C.
ONE DOLLAR

Solution ownership and empowerment are about making the most of your IT budget.

No more brick walls.

No more lock-in.

Traps and pitfalls



Fake Open Source

A license is not enough.

Proprietary solutions with open code.

Tell-tale signs:

- no community
- “enterprise” version with closed source functionality

Code dumps

Dead software
OSS as a last resort
Tell-tale signs:

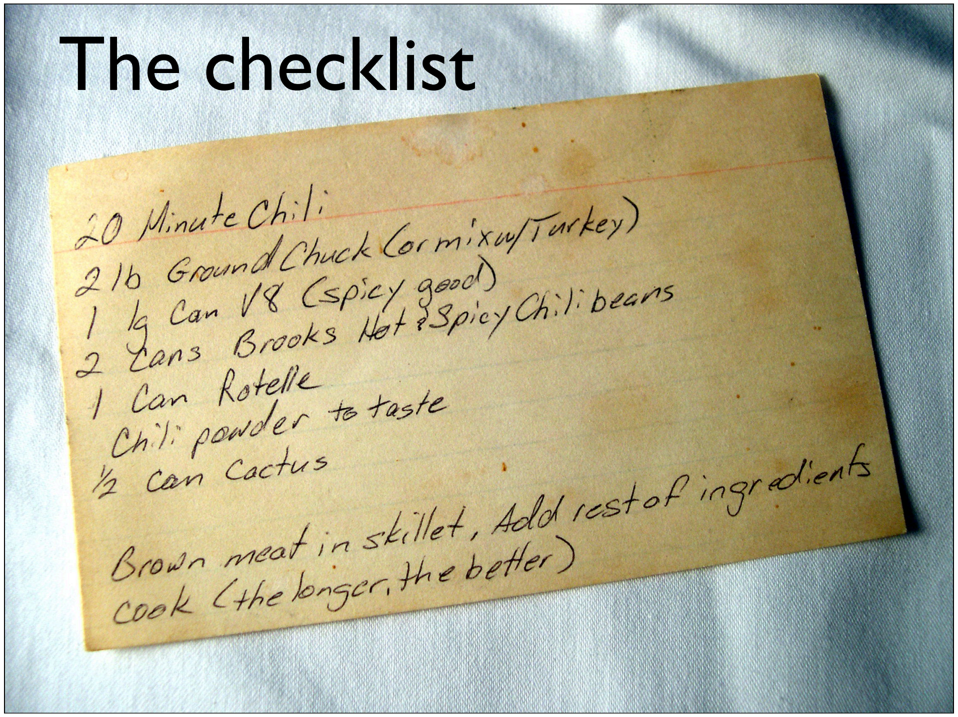
- no community
- no development

Legal issues

IP matters
Licenses matter
Side effects of license mix matter

The need for a strategy

The checklist



20 Minute Chili
2 lb Ground Chuck (or mix w/Turkey)
1 lg Can V8 (spicy good)
2 Cans Brooks Hot & Spicy Chili beans
1 Can Rotelle
Chili powder to taste
½ can Cactus

Brown meat in skillet, Add rest of ingredients
cook (the longer, the better)

A recipe for successful open source adoption

Environment

Is your organisation ready for Open Source?

- Management commitment
- Cultural change
- (some) DIY IT

If not:

- Contact an Open Source vendor
- Manage Open Source the traditional, proprietary way
- Get only the technical benefits (still not bad)

Solution background

Study the proposed solutions, to determine their fitness for purpose. Specifically, apply important open source metrics to the solutions.

Look at the project, look at the community driving the project.

Project stability

Project stability (easy questions)

How long has the project been around?
How many reference customers of your size has it got?
Are there books published?
Download statistics?
Commercial software?

Community

Community (much more difficult)

- How diverse (multivendor) is the community?
- Democracy or oligarchy?
- Governance rules?

FOSS is about community: the community of developers and users that work on and with the software, providing code, support and bug reports.

Once you start using FOSS, you become an important part of a global **network** of individuals and organisations.

Legal Issues

Are / have there been IP issues?
Is there a process to ensure license compatibility within the project?
Is the license compatible with the intended usage?
Are you prepared for the worst case scenario?

Community

Community assessment

- Diverse participation?
- Governance?

Technical Issues

Does it solve the problem in full?
If not, how can it be customised/extended/integrated?
Is technology compatible with your environment?
How is deployment managed?
How is configuration managed?
Does the solution support High Availability?
Can you ensure a business continuity plan?

Risk management

Technical risks:

- Software quality
- Security
- Scalability

Community risks:

- Why is this software Open Source?
- Will the project stay?
- What if the community divides?
- What if the project diverges?

Support

Shameless plug

Can you easily get commercial support?
Do you have alternative support channels?
Does support come from professionals or companies?
Will your support channel take care of your customisations/integration/extensions as well?
Can your support channel ensure that patches will be submitted and could be included in the mainstream distribution?
Can your support channel (co)drive the release process?

The plan

1. Commit

From   [PROTECTED]
Subject   Re: ClassCastException in CountAggregator
Date   Wed, 07 Dec 2005 20:04:09 GMT

I solved my problem not using derby :(
Appear the list is more interested in voting logos,
than solving critical bugs.

[PROTECTED]

2. Set your expectations

3. Get ready

Get trained
ensure bottom-up willingness to participate
Ensure top-down willingness to participate

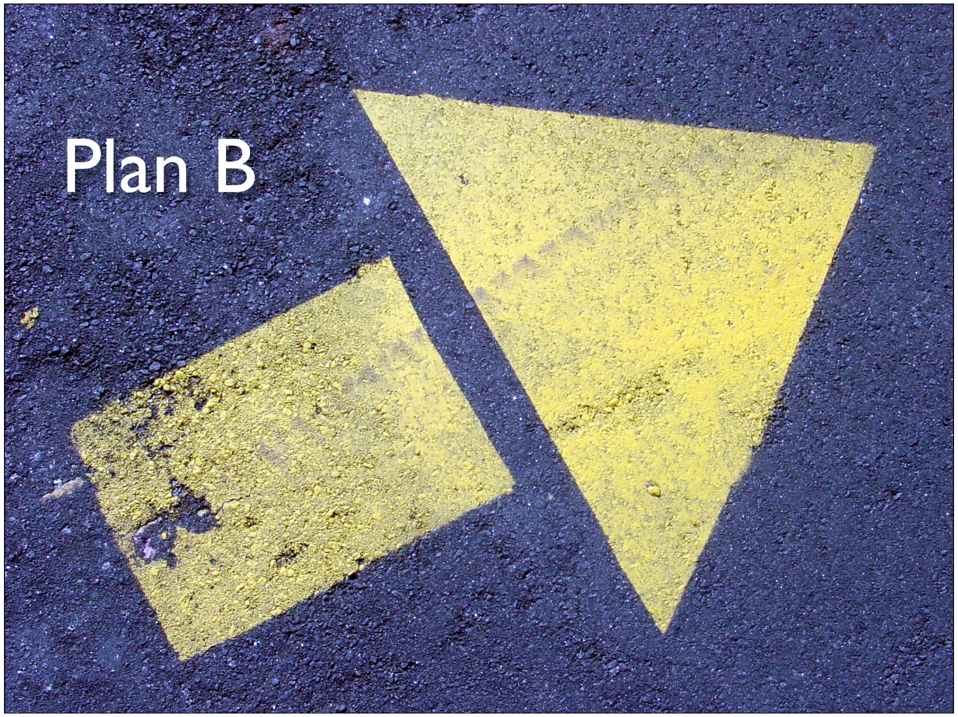
11 A B E
12 A B E
13 A B C E
14 A B C D E
15 A B C D E
16 A B C D E
17 A B C D E
18 A B C D E
19 A B C D E
20 A B C D E

4. Evaluate

5. Plan

Adoption
Deployment
Support
Decommission

Plan B



Approach OSS as if it was commercial software
Deal with vendors
Leverage the technology advantage
Consider a roadmap to the Real Thing

Thank you