Coordinating and Motivating open source contributors

Workshop @ LinuxWorldAsia

James Howison

School of Information Studies

Syracuse University

floss.syr.edu

The Human Side …

FLOSS is code but people too.

Volunteers, Paid developers, Individuals and companies

Varying in aims, motivations and skills

Explore this landscape together

But first …

I’m an Australian doctoral student studying in the US with Prof. Kevin Crowston

Background in Economics and Politics from University of Sydney

Master’s study in Computer Science at University of New South Wales

4th year of my PhD at Syracuse University

Prof. Crowston provides support through a grant from the US National Science Foundation.

Research and Experience

Interviews, surveys and studies of development archives

✓ ApacheCon, O’Reilly OSCon,

✓ FOOCamp, OSDC in Australia

Developer on BibDesk, a small OS X reference manager

✓ Scratching my graduate student itch

But India?

Just starting to learn …

So far learnt more about weddings and Saris!!

Detailed Overview

Lifecycle of FLOSS projects

Motivations of participants

Joining projects

- ✓ For both individuals and companies

Implications for project infrastructure

- ✓ Communications, CVS++, planning

Want to weave your interests throughout

Last hour is to workshop issues or opportunities you face

Resources

floss.syr.edu/Presentations/linuxworldasia/

- ✓ All the links and papers we'll be discussing
- ✓ I would like to add resources suggested by you

Quick Intros from you …

Who are you?

What’s your involvement with FLOSS?

Exercise: What do you want?

Sheet of Paper and a Pen

Jot down a question or topic from ‘the human side’ of FLOSS you’d like addressed

Pass it to your left and jot down another on the sheet you get

And again …

Now from the sheet in front of you, read out the one, from the three, that you like the best

…

The Basics

FLOSS == Free, Libre, Open Source Software

- ✓ Different Ideologies, Similar social and software practices

Licenses and practices lead to collaborative development with freely available code

- ✓ Teams and communities distributed across organizations and geography

What’s being built?

The headliners:

- ✓ Linux, Firefox, Apache httpd, perl

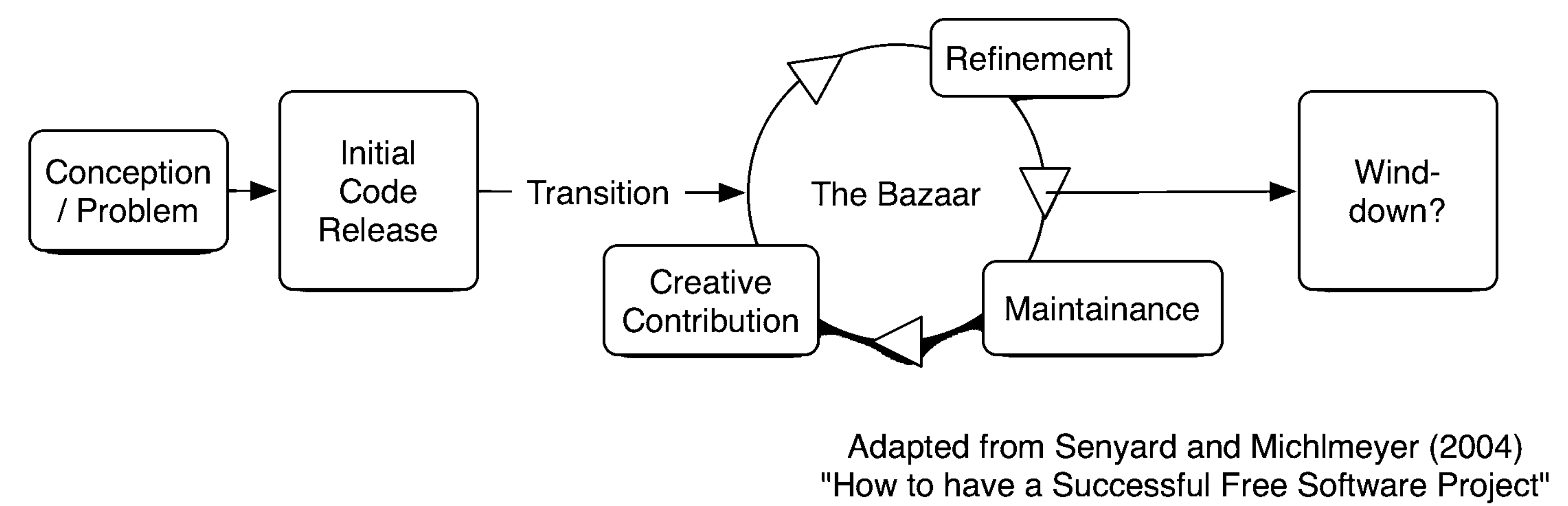
These get all the press, but aren’t necessarily typical of an open source project

From small tools to entire software stacks

Projects are diverse socially

- ✓ Personal stories are invaluable
- ✓ But empirical research is needed to balance

A model project lifecycle



Adapted from Senyard and Michlmeyer (2004)
"How to have a Successful Free Software Project"

Typically Individual or Small co-located Group

An Open Source Team and Community

A Cathedral before the Bazaar?

“It's fairly clear that one cannot code from the ground up in bazaar style. One can test, debug and improve in bazaar style, but it would be very hard to originate a project in bazaar mode. ... When you start community-building, what you need to be able to present is *a plausible promise.” ESR, CatB*

Initial Conception and release

The ideal:

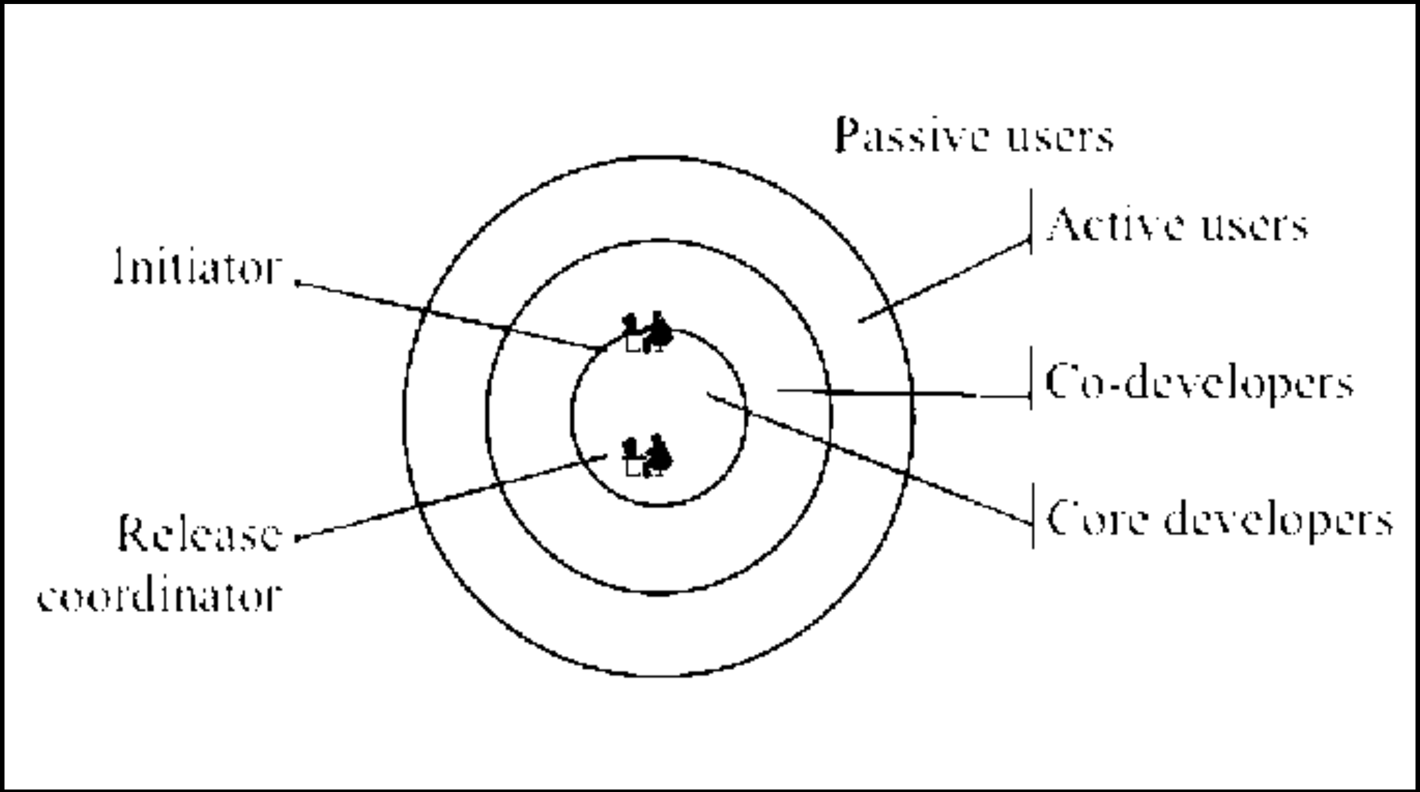
- ✓ Simple aim, well understood locally
- ✓ Release that works, but not too well
    - ✓ Yes, leave bugs (low hanging fruit …)
- ✓ Grows very slowly, resist early contribution

Failed Projects

- ✓ Check out the vast majority of Games on SourceForge
- ✓ … huge teams early, big ideas, little or no code.

Transition: Building a team and a community

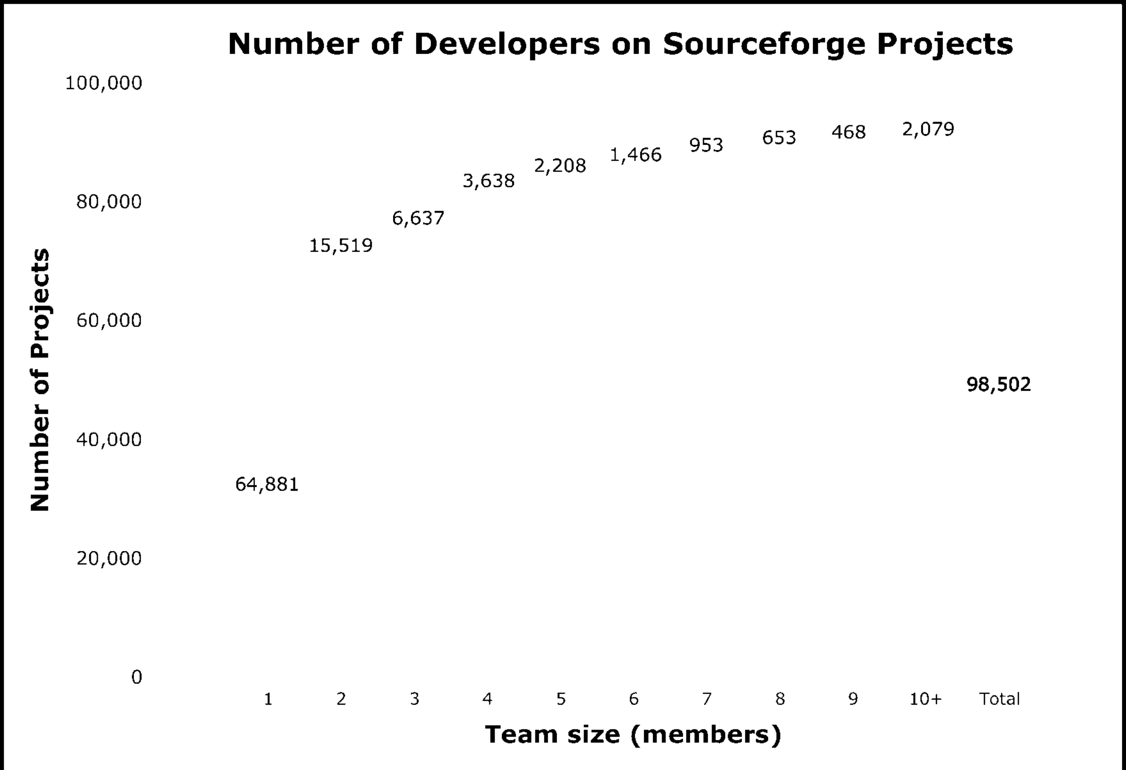
First Question: What does one want?



But How Big?

Teams on the road to success almost never have above 10 core developers

Very large teams are either massively successful, or clear failures



Co-developers

Incremental improvements

Often without write access to CVS

Unexpected uses and solutions

Active Users

The ‘many-eyes’ making bugs shallow

Report and characterize bugs and feature requests, usually for others to fix

Provide ‘buffer’ between developers and new users, especially for configuration heavy projects (eg MythTV)

Highly transitory, vast majority only participate for around 10% of the lifetime of a project

Motivations

Understanding why people are involved is useful for

- ✓ Understanding their actions
- ✓ Understanding what will interest and excite them, and
- ✓ Understanding what is likely to annoy or antagonize them

Brainstorming motivations

Think of your involvement, why are you there?

Think of others

Write a few reasons down and pass to your left.

Let’s get them on the whiteboard

What does the research say?

Quite a lot of survey research on FLOSS participant’s motivations

Ghosh et al (2002) Infonomics survey

- ✓ Over 3,000 developers, many topics

Lakhani and Wolf (2003) “***Why hackers do what they do”***

- ✓ 678 participants in over 287 projects

Ghosh et al (2002)

Learning Skills,

Sharing,

For the products themselves,

Ideology,

New form of working,

Jobs and other personal extrinsic rewards

Ghosh continued …

What about reputation?

- ✓ Found that it wasn’t crucial initially
- ✓ But became more important amongst longer term contributors

"94% of the OS/FS developers mark their contribution to software projects as theirs (figure 50). Almost three fifths even declare that they consider this as very important.”

- ✓ We’ll come back to this

Lakhani and Wolf (2004)

Enjoyment

- ✓ Intellectually stimulating
- ✓ Like to work with their team

“User need” for the product

Obligation to Community

Their results emphasize creativity. Autrijus Tang (2005) calls this -Ofun (optimizing your project for fun)

Does getting paid matter?

Theory expects it to

✓ ‘crowding out’ of intrinsic motivations by extrinsic motivations

Lakhani and Wolf found

✓ 40% of participants were getting paid (up from Ghosh survey)

✓ Did not find ‘crowding out’ to matter but paid participants did more work

✓ Other research shows that they move more quickly to the centre of a project.

How’d we do?

Did we get all those?

Any we had up there that are missing?

Mythconceptions

“Beating proprietary software” always shows up low on the list

Motivations are surprisingly varied and multi-faceted.

Ideology plays a secondary role

Summing up

Projects have a development lifecycle

- ✓ Different implications for interaction

Participants have diverse motivations

- ✓ Learning and challenge are key motivators

15 Minute Break

Back at 15.50?

Welcome back

What was the discussion at break?

Anything to add to the agenda?

Motivation implications

Knowledge, Sharing, Product

Open Communities

- ✓ Anonymous CVS that always builds
- ✓ No log-ins to get the source etc

Ensure that potential developers can immediate act on their motivation

Roadmaps

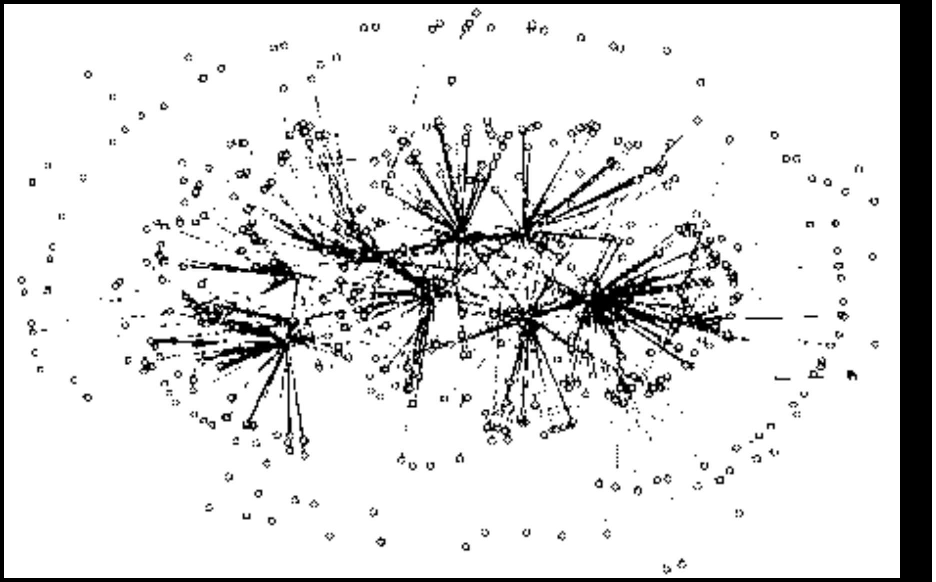
- Classic and traditional aid to project planning
- Helps customers understand migration and development aims
- Can de-motivate a community …
  - ✓ They can imply that everything is ‘under control’
- Plans are not action
- Free floating to-do lists are a more motivating option
  - ✓ Keep some ‘low-hanging’ fruit on there.

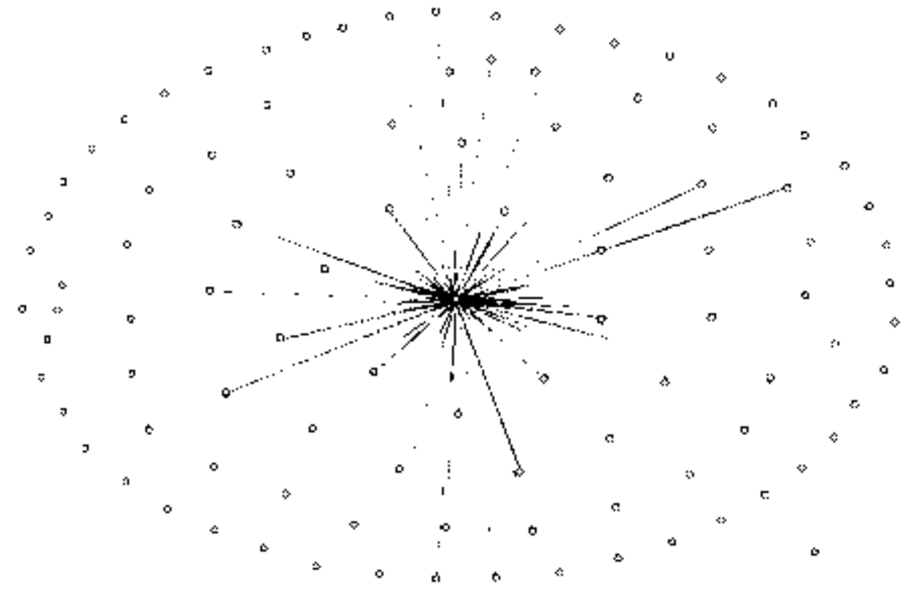
Legitimacy comes from action

- No org chart
  - ✓ But not without structure
- (Show pretty SNA pics)

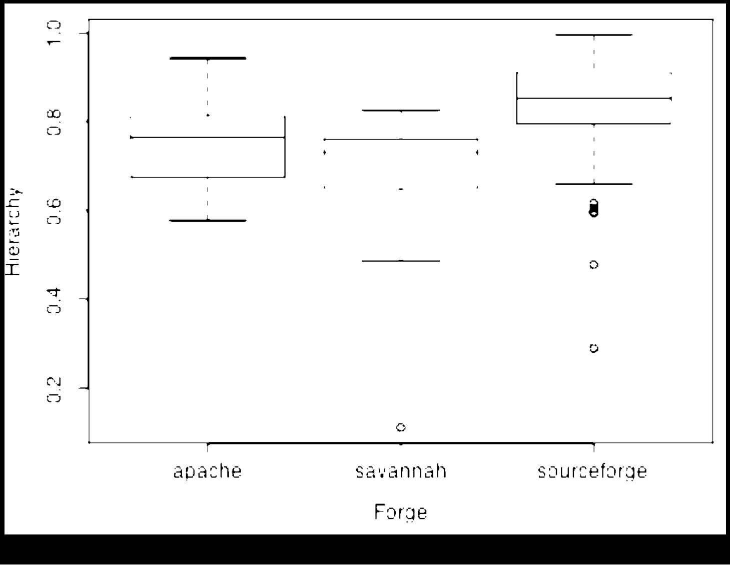
Informal Structure

Centralization varies
— and varies over time





Projects are generally hierarchical
in their communications structure



Leadership change

Some high profile cases

- ✓ Some institutionalization of change
  - ✓ Perl’s Pump-King
  - ✓ Debian’s Leadership elections

Empirical evidence suggests, however, that it is quite uncommon.

- ✓ > 50% of teams only ever had 1 person at centre, another 20% had 2 throughout their lifetime.
- ✓ Large projects do not have more leadership changes (stability permits growth?)

Legitimacy from action, skill

Action, contribution are source of legitimacy

- ✓ Presumption in favour of integrating work in many, but not all projects

However, technical skill and correctness plays a role

- ✓ Gaim example. 3rd patch queue for “never to be integrated”

Need or “cash riding on it” are not reason enough for people to cooperate.

Joining Projects

I, or my company, wants to get involved in a FLOSS project, how do I best go about it?

Questions:

- ✓ Why?
- ✓ What level of involvement
  - ✓ Base a business on FLOSS?
  - ✓ Just fix a few bugs?
- ✓ What length of commitment do we expect?

“Joining Scripts”

von Krogh et al (2003) studied participants joining projects, including FreeNet

Eventual joiners started with code, not with plans or offers

- ✓ Anonymous CVS facilitates this

Strategies for involvement

Be there for the long term

- ✓ “FLOSS is only free if your time is worth nothing”
- ✓ Create a cost centre for your involvement for employee time to be charged against

Be a consistent ‘active user’

- ✓ Ask good questions (more on that next)
- ✓ Create useful bug reports

Turn your use-case into a learning opportunity for the developers

- ✓ Means sharing more than usual

What about ‘Bounties’?

Tempting to replace long-term involvement with cash

- ✓ Blunt, extrinsic reward

Can be a risky proposition

- ✓ Failure of most ‘cash for code’ exchanges
- ✓ Risks offending the community
- ✓ Strangely enough, developers might have difficulty accepting cash
  - ✓ Might need a foundation for Tax purposes
  - ✓ Difficult to divide, can cause resentment

Cash doesn’t make up for legitimacy

MythTV example -> PDF

Compare to Pluto@Home’s attitude

- ✓ Packager of OpenSource media and home automation solutions
- ✓ Release code but also detailed documentation (dev.plutohome.org)
- ✓ Also are known to answer their phones for developers

Remember the motivations

- ✓ Skills, Sharing, Product

Creatively using cash

“But I’ve got a budget”

- ✓ Internal cost centre for participation

Consider organizing a “sprint”

- ✓ Focused face to face gathering
- ✓ Used in Python, Zope and Plone
- ✓ Computer Associates provided funding

Clear Problem, Venue, WiFi, Food, Beer

Best done by someone already familiar with project and ideally gain support of lead dev

Friendly Forks

There is a strong norm against forking FLOSS projects

Friendly forks hope to fork the code-base but contribute back fixes relevant to both branches

- ✓ Apple’s Safari browser forked from khtml
- ✓ Knoppix forked from Debian

Difficult, high effort and can spilt the user community

ESR: “How to ask questions the smart way”

A gem of practical FLOSS culture advice

“hackers actually like hard problems and good, thought-provoking questions about them”

“good questions are a stimulus and a gift”

Do the homework and describe it in the question

- ✓ FAQ, Archives, Web
- ✓ Read the source (if you can)

“Display what you have learned from doing these things”

Make it clear that you will follow through

More on smart questions …

Get the forum right

- ✓ Ask in public (Larry Wall “Learning in Public”)
- ✓ If not hacking on code, ask the user list

Use good, old-school email technique

- ✓ Use Plain text
  - ✓ Use links to pictures, and large files rather than attachments (YouSendIt.com)
  - ✓ Outlook *can* be properly configured
- ✓ Don’t ‘top post’
- ✓ Selectively quote earlier discussion creating a thread with ‘>’ characters

Most of all, show your adventure towards an answer, not your frustration.

Reporting Bugs

Re-try with the latest code from CVS

- ✓ Nightly release also fine

Persistence

- ✓ Be part of the conversation to characterise and fix the bug

Steps to re-produce are crucial

Project Communications Infrastructure

Goal is to balance incidental awareness against overload

Briefly skimming discussion not directly relevant creates shared understand

- ✓ Also knowledge about “who knows what”

Only split lists after overload has become a problem

- ✓ Start with one list called “dev” and one tracker
- ✓ Split to “users” and perhaps Bugs and RFE
- ✓ ‘Support’ Trackers are usually pointless

Project comms II

Mail is generally preferable to Forums

- ✓ All in one place (the mail reader)
- ✓ Large projects with active users and many new users do well with Web Forums

Synchronous vs Asynchronous

- ✓ It’s fast, but excludes people
- ✓ Tends towards less thought through questions

Break …

Back at 4.30?

Round Table Discussion

Let’s review our generated list of interests

- ✓ Did we cover these?

What would you like to discuss?

- ✓ Practice “the smart way”
- ✓ Concrete detail
- ✓ Your ‘adventure’

Any suggestions for research?

I’ve been jotting down notes …

Any topics you’d like to know more about?

Any ‘conventional wisdom’ that you are not seeing come true?

Thanks

- Remember resources at:
  - ✓ floss.syr.edu/Presentations/linuxworldasia/