

Why sustainability matters for open and closed source software

James Dalziel
Professor of Learning Technology
Macquarie E-Learning Centre Of Excellence (MELCOE)
Macquarie University
james@melcoe.mq.edu.au
www.melcoe.mq.edu.au

Presentation for OSS Watch “Open Source and Sustainability” Conference,
April 10th, Oxford University, UK

Overview

- Why sustainability matters
- The big question
- Sustainability challenges from closed source
- Sustainability models for open source
 - Examples
 - Case Study: LAMS
- Business models: Investment, reward, profit

Why sustainability matters

- Total cost of ownership for organisational software
 - Cost of acquisition
 - Cost of implementation
 - Cost of maintenance/ongoing support
 - Cost of exit
- Cost of acquisition is rarely a major part of TCO
- Many paths to poor return on investment in software
 - Wrong choice – not appropriate to organisational needs
 - Implementation too costly relative to benefits realised
 - Low organisational adoption undercuts promised benefits
 - High cost of exit leads to vendor lock + rising prices

Why sustainability matters

- Poor return on investment in software (vendor side)
 - Software no longer available (company failure, etc)
 - New feature development slow/buggy/wrong direction
- None of these challenges are unique to open or closed source software
- However, open source software focussed attention on certain aspects of sustainability due to the financial model

The big question

- What factors ensure the ongoing sustainability of an open source software system?

We don't know

- What factors ensure the ongoing sustainability of a closed source software system?

We don't know

The big question

- That's not to say that sustainability factors for open and closed source are the same, just that there are risks (and benefits) on both sides
- Consider the issues of (a) failure and (b) license fees:
 - Open Source
 - Pro: Source code remains open if project fails
 - Con: Lack of license revenue may inhibit development
 - Closed Source
 - Pro: License and maintenance fees helps pay for development
 - Con: Software may disappear if company fails, etc

Challenges from closed source

- When considering open source sustainability, it's worth keeping in mind challenges from closed source
 - “End of life” products
 - Company acquisitions
 - Changed company focus
 - Company failure
- Compared to open source, a key factor here is that organisations are often at the mercy of their vendor
 - Rarely able to take the code and continue to use and improve it themselves when the company stops support

Sustainability models for OSS

- No “one” model for sustainability
- Increasingly projects are adopting multiple models
- Sustainability is not simply a quality of a project (or team) – it also depends on the context of adoption
 - Especially important in cases of government use of open source software – that is, an OSS project that may not be sustainable can become so with government involvement

Sustainability models for OSS

- Example A: Direct government support
- Eg, Shibboleth for UK HE/FE and schools
 - National middleware infrastructure for federated authentication/trust
- Following pilots and evaluations of features, national adoption was recommended
 - Level of understanding of key gaps? (ShARPE)
- This decision was made despite other approaches
 - Eduserv's original Athens system
 - Products from large vendors (eg, Sun & Novell Federation systems)
- Details of sustainability yet to be finalised, but given national commitment, they're likely to be worked out somehow

Sustainability models for OSS

- Example B: Charity/government/home institution funding + partner contributions
- Eg, Sakai
 - Learning Management System
- Initially a combination of pre-existing software (CHEF), Mellon and home institution funding (\$6.8M) + partners (\$10K/year for three years – approx 100 partners = \$1M/year)
- Commercial partner affiliates (12) – support implementation
- Ongoing sustainability from
 - Home institution development staff contributions
 - Partner funding to Sakai Foundation (for management, not code)
 - Code contributions from partners and others

Sustainability models for OSS

- Example C: Commercial services and support around OSS
- Eg, Moodle and Moodle partner network
 - Learning management system and network of support partners
- Moodle leader (Martin Dougiamas) certifies certain Moodle partners (generally geographically based) to provide Moodle branded services and support in return for % of earnings
 - Earning help to support core OSS development
- Core development also complemented by earnings on custom feature development, advice, etc
- Ongoing sustainability from:
 - Long period of modest returns, but now growing with widespread adoption of Moodle - much early development done “for free”
 - Strong developer community, inclusive development process
 - Increasing desire for certified Moodle service and support partners

Sustainability models for OSS

- Example D: Dual licensing
- Eg, MySQL
 - Open source database company
- Software freely available under open source licence (GPL)
- Fee-based alternative licence for those want to avoid the GPL
- Dual licensing is complement by other services and support
 - Training & training certification, support network, hosting, etc
- Investment from Venture Capitalists
- Sustainability
 - Mixture of licensing revenue and services revenue
 - Widespread adoption leads to interest from investment community

Sustainability models for OSS

- Example E: Income from content-related activities
- Eg, Firefox
 - Open source browser with Google default start page
- Open source software – but most users see it as just another no cost desktop application
- Choice of default page as Google search engine reported to be earning significant advertising related revenue
- Sustainability arising from:
 - Services and support
 - Earnings from content related activities

Case Study: LAMS

- LAMS (Learning Activity Management System) is a new generation of e-learning software
- Based on the evolving field of “Learning Design”
- LAMS helps teachers/lecturers to create and run “digital lesson plans”
 - Sequences of content and collaborative activities
- LAMS sequences can be shared and improved
 - LAMS Community and “open source teaching”
- Example of innovative OSS (no equiv closed source system)
- Originally developed as commercial software, shifted to an open source business model



New



Open



Save



Copy



Paste



Transition



Optional

Activity toolkit

Q & A

Q & A



Grouping



Chat



Polling + Notebook



Submission



Q & A



Read noticeboard



Chat & Scribe



Resources & Forum



Polling



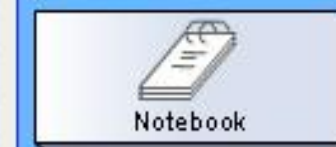
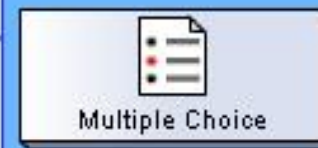
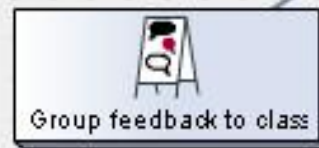
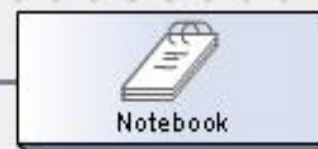
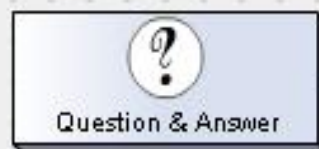
Share resources



Forum



Notebook

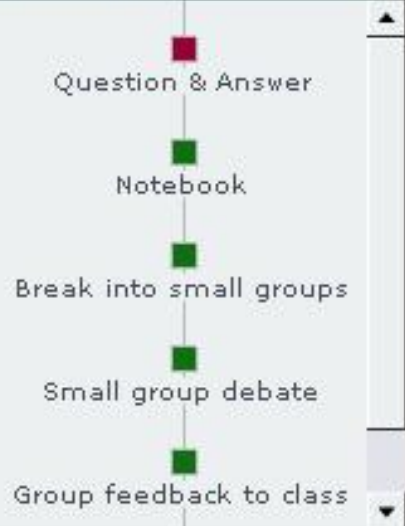




- Resume
- Exit

Available sequences

Progress



Notebook

Title

Question and Answer

Question and answer

What is greatness (in a human being)?

Case Study: LAMS

- Why was open source a good decision for LAMS?
 - Encourage rapid widespread adoption of the Learning Design approach and the LAMS software
 - Foster open source development of new activity tools (as well as extend the core platform)
 - Combines open source and open content approaches
 - LAMS as open source helps inform future open standards development for Learning Design
 - LAMS as open source helps avoid file format lock-in
 - Non-profit foundation: www.lamsfoundation.org
Commercial services: www.lamsinternational.com
Software/resources: www.lamsfoundation.org/CD/

Case Study: LAMS

- How will LAMS be sustainable?
 - Dual organisational structure:
 - LAMS Foundation (non profit) for public funding, charitable grants, etc; owns software, releases as OSS
 - LAMS International Pty Ltd (services company) for paid services and support (eg, hosting, tech support, training, custom development, custom content, etc)
 - Profits on services help support ongoing development
 - LAMS chose the GNU General Public License (GPL) to allow for the possibility of “dual licensing” (cf MySQL)
 - Software free to all under the GPL
 - Commercial software that wants to integrate and distribute LAMS can pay for a non-GPL license

Business Models: Investment, reward, profit

- Open source services models can be viable commercial approaches to building a profitable business
 - Largely depends on ability to tightly manage a services business
- However, the nature of the profits achieved may not be of interest to many investors
 - Especially those who require high rates of return (eg, VCs)
- Compare open source system X with commercial system Y
 - X: 2,000 days of development, mostly unpaid, sales and marketing through word of mouth, shoestring operations
 - Modest returns to support decent salaries for a few people
 - Y: 10,000 days of development, commercial salaries, VC investment of \$100M expecting 50% IRR, large sales and marketing & operations
 - Yearly revenue needs to be over one hundred million dollars...

Business Models: Investment, reward, profit

- Apart from sustainable services companies (do-able), a key challenge is ongoing development. Solutions include:
 - Building a successful volunteer developer network
 - Investing services profits back into paid developers
 - However, margins may come under pressure as services competitors arise
 - Being selected by government for major infrastructure that can't afford to fail
 - Obtain research and development funding (grants)
 - But rarely ongoing, and subject to vagaries of funding cycles
 - Obtaining other sustainable revenue sources (dual licence, content)
 - Reach system maturity so ongoing development becomes modest
 - Or.... slow/stop development