what does software sustainability mean?

- in the context of research, it means either:
  - preservation - in the sense of ‘archiving’

- or
  - ongoing provision as a service (e-infrastructure)

- debate dogged by frequent mutual misunderstanding with these two aspects being conflated or confused

- difficult (undesirable?) to entirely separate these - too many factors in common
when to sustain....

• in the research (science) context:
  • sustain to preserve so that experiments can be repeated - reproducibility of results - software is ideally unchanged

• in the eResearch/infrastructure context:
  • sustain to achieve community-wide efficiencies - software is often continuously changing

• in a business context:
  • sustain to maintain existing-customer satisfaction - mostly a sunk cost....
  • much software development discipline comes from business

...and when not to sustain

![Diagram](http://web.mit.edu/6.933/www/Fall2000/teradyne/clay.html)

from The Innovator's Dilemma, Clayton Christenson
issues

• **shelf-life** of un-attended software is frighteningly short (6 months!)

• **complexity** - nearly impossible to write significant software without it having dependencies on other code-bases/libraries

• **amateur production** - the people writing software to support e-infrastructure are often (usually?) not professional software developers - e.g. **grad-students**

---

**a response: incentive & capability**

• better disciplined engineering?

• is software engineering an appropriate response?

• NSF Workshop on Cyberinfrastructure Software Sustainability and Reusability (2009) recommended **good software engineering practice**

• however, there is a backlash against ‘engineering’

• also - our CS departments also don’t often teach these skills!

• Greg Wilson and software **carpentry**

• artisanship?
more understanding needed

- the properties of software in terms of digital preservation
  - some good DCC and JISC-funded work in this area
- applying this to the development process
- knowledge exchange between professional developers and researcher-developers
- business models for sustained open source development and maintenance
- how to recognise and then incubate potentially important software

mandate open source?

- well yes, but....
- open source is a potentially powerful approach to developing software
- it is not sufficient by itself
  - Sourceforge (and more recently Git) is a graveyard of opensource software code that no one cared to maintain
- open source as a default requirement is sensible, when backed up with a sustainable development model
incubate?

- software which is **ahead of its time** needs to be **looked after** and kept alive before it is more widely adopted

- note - good software often is developed in advance of large-scale demand

- Apache Foundation offer an interesting model of requirements for incubation:
  - evidence of interest
  - evidence of growth of interest and commitment
  - no explicit need for running code to get through the incubation stage

bring researchers into ‘developer space’

- software developer contact with users is rarely on the terms of the developer

- helping non-developer researchers to understand a little of what developers do and **how they think** is **empowering** for both
takeaways

• this space is **insufficiently understood** - to the point of occasional **basic confusion**

• software development is a **discipline**, like research, and like data-curation. Training and support are essential

• even more than with data, sustainability must be **addressed at the outset** of any development

• approaches to improving the sustainability of software in our sector may be found in addressing the **practice of production** rather than in the properties of the software itself

Software, Community and Sustainability

Ian Dolphin

Executive Director, Sakai Foundation

iandolphin@sakaifoundation.org
What are Jasig and Sakai?

- Meeting points to pool resource and coordinate software projects in Higher Ed
- Multiple projects, Sakai CLE, OAE, uPortal, CAS, Bedework, uMobile, Portlets
  - Conferences
  - Technical/social infrastructure
  - Licensing, Legal
  - Outreach and engagement
- Membership organisations -
  - Collectively - 101 Higher Ed Institutions, 23 Commercial Affiliates
  - 66% North American, 33% outside North America
    - Strong clusters in Europe, Japan, South Africa
  - Collegial organisation - not open source “pyramid”
- Registered as non-profit corporations in the US

What’s the problem?
What’s the problem?

How many non-profit umbrellas do we need?

• A: Not “one”, but fewer than we have now ...
Jasig and Sakai Merger

- Common organisation - focus on software to support academic mission
  - Infrastructure and superstructure
- Maintain collegiate nature of Jasig and Sakai - elected Board, etc
- Software community - governance diversity - no universal model for each
  - Organic <-- Managed
  - Determine by position in lifecycle, position in software stack, context, culture
  - Individual community choice informed by substantial pool of expertise
- Communities of interest
- Branding at software community level - grow foundation brand over time
- The challenge: standing at the juncture of network and hierarchy

Objectives

- Immediate: Low hanging fruit - efficiency gains
  - Conferences
  - Technical Infrastructure
  - Licensing, Legal
  - Outreach and engagement
- Medium
  - Develop an inclusive incubation process
- Medium/Long
  - Build a better understanding of, and voice for, open software in higher ed
  - Stimulate interaction between software communities
  - Engage in other re-alignments
More at ...

http://www.jasig.org/jasig-sakai-final-merger-plans