Measuring Software Sustainability

Steve Lee, OSS Watch
http://www.oss-watch.ac.uk
@SteveALee
info@oss-watch.ac.uk

Copyright © 2009 University of Oxford.
Licensed under the Creative Commons Attribution-ShareAlike 2.0
What is software sustainability?

• Ability to survive
  – Financial fluctuation
  – Membership changes
  – External factors
• Support sustained use
• In context of open source
Project view of Sustainability

- Continuous forward motion
  - Innovation
- Secure resources
  - Development
  - Support
  - Contributions
  - Collaborations
- Enable exploitation
User view of Sustainability

• Reduce risks of dependency
  – Enterprise procurement (eg VLE)
  – Ad hoc use, eg in classroom or home
  – Software development
    • Tool or component

• Transparency

• Engage, contribute or collaborate
Open Innovation

• From idea to operation
• Research projects
• Classroom activities
• Enterprise deployment customisation
• Supports sustainability
Open Innovation in Software

• Open Source Software, plus
• Open Standards, plus
• Community development processes
  – Strong leaders
  – Vision
  – Continual innovation
  – Sharing of expertise and resources
Consider the Internet

• Ideas
  – A few big ideas (e.g. protocols, web server/client, REST)
  – Many “small” ideas (e.g. Twitter, social networking)

• Implementation
  – Prototypes and experiments (e.g. Gopher)
  – Incremental improvements (e.g. Web browser)
    • Software reuse

• Operation
  – Sustainability (e.g. Google, Facebook and even .ac.uk)
Measuring Software Reusability
Technical and IP reusability models

• Reuse Readiness Levels
  – See http://oss.ly/rrl

• Business Readiness Rating
  – See http://oss.ly/brr

• Open Source Maturity Model
  – See http://oss.ly/osmm

• Qualification & selection of open source software
  – See http://oss.ly/osmm
Measuring Openness
Capability Maturity Model

• Measures process:
  – Quality
  – Repeatability
  – Flexibility

• Nothing to say about openness
Sustainability Rating

• Focus on ability to engage in community development/open innovation
• Evaluates 5 key enablers for open innovation
  – Legal, Data Formats and Standards, Knowledge, Governance, Market
• Provides a % score of project openness
• No technical evaluation
• See http://oss.ly/or for more info.
The Apache Software Foundation Incubator

- Community before code
  - Develop community with overlapping needs
  - Give them development resources
  - Give them an independent process
  - Code will emerge
- Incubator incubates community (not code)
  - Graduation requires three independent committers
  - No direct funding of projects
- Process refined over 15+ years
Technical Evaluation Techniques
Automated Techniques

• Structural Integrity
• Understandability
• Completeness
• Conciseness
• Portability
• Consistency
• Maintainability
Software Sustainability Maturity Model
Why bother?

• Encourage an open innovation culture
  – Collaborative development
  – Reusable software

• Encourage sustainable software development
  – Reduce duplication of effort
  – Improve quality
  – Survival of software beyond project funding

• Increase ROI in software
Does it work for self evaluation?

• Informally trialled with 50+ project managers and developers

• All felt that the exercise was valuable
  – Highlighted project issues requiring attention
  – Provided a focus for skills development
  – Useful for both open and closed source activities
Does it work for reuse evaluation?

- Informally trialed against 15+ projects
- Highlighted potential risk factors
- Only truly useful for open source
- Allows alternatives to be compared
- Many aspects of evaluation can be automated
  - If project meta-data is available
Current Status

• Evaluation form
  – http://oss.ly/or

• Summary Results (cleaned periodically)

• To Do
  – Verify coverage of all evaluation models
  – Define maturity levels
  – Recommended maturity progression paths
  – Integration with http://registry.oss-watch.ac.uk
    • Automate evaluation where possible
Current Status

• Evaluation form
  – http://oss.ly/or

• Summary Results (cleaned periodically)

• To Do
  – Verify coverage of all evaluation models
  – Define maturity levels
  – Recommended maturity progression paths
  – Integration with http://registry.oss-watch.ac.uk
    • Automate evaluation where possible