

Research Collaboration and Software Sustainability

Ross Gardler, OSS Watch

<http://www.oss-watch.ac.uk>

@rgardler

info@oss-watch.ac.uk

Types of Research Collaboration (Innovation)

- Intra-organisational open innovation
 - Internal collaboration
- Inter-organisational open innovation
 - External collaboration
- User innovation
 - Involve users in development process
- Collective innovation
 - Mass participation (crowdsourcing)

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

Reuse Readiness Levels

- Defines 9 levels of reuse
 - 1 No reusability
 - 5 Reuse is possible
 - 7 Reuse with minimal risk
 - 9 proven reuse
- Defines 7 evaluation topics
 - Documentation, Extensibility, IP Issues, Modularity, Packaging, Portability, Standards compliance, Support, Verification and Testing
- See <http://oss.ly/rrl> for more info

Business Readiness Rating

- Four steps
 - Short listing
 - Ranking and weighting of the selection criteria
 - Data gathering for each criteria
 - Calculation and publication of results
- Twelve evaluation criteria
 - Functionality, Usability, Quality, Security, Performance, Scalability, Architecture, Support, Documentation, Adoption, Community, Professionalism
- See <http://oss.ly/brr> for more info

Open Source Maturity Models

- Same four phases as BRR
 - Short listing, weighting, Data gathering, calculation and publication
- Six evaluation categories
 - Software, Support, Documentation, Training, Integration and Professional Services
- See <http://oss.ly/osmm> for more info

Qualification and Selection of Open Source Software

- Four steps
 - Definition of domain evaluation template
 - Evaluation
 - Qualification (weighting)
 - Selection
- Three axis of evaluation criteria
 - Functional coverage, risks for new users, risks for service providers
- May sound simplistic, but is customisable on a per application basis
- See <http://oss.ly/qsos> for more info

Measuring Openness of Innovation Practice

Capability Maturity Model

- Measuring process:
 - Quality
 - Repeatability
 - Flexibility
- Capability Maturity Model Integration for Development (CMMI-DEV)
- Software Process Improvement and Capability dEtermination (SPICE or ISO 15504)

Openness 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
- Testability
- Usability
- Reliability
- Efficiency
- Security
- Viability

Software Sustainability Maturity Model

Bringing it all together
(in development)

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 research 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)
 - <http://oss.ly/orsum>
- 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