Research Infrastructure for Empirical Science of F/OSS

Les Gasser, Gabriel Ripoche, Robert J. Sandusky

Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign
{gasser,gripoche,sandusky}@uiuc.edu

ICSE – MSR Workshop
May 25, 2004
Introduction

- UCI/UIUC 2003 “Design in F/OSS” workshop: Pressing need for research infrastructure

- What are the objects and methods of analysis?
- What are the data requirements?
- What are the available data?
- What are the common issues?
- How can these issues be addressed?
Our research

Questions

• How are software problems managed in practice, in large-scale, distributed communities?
  - What are the factors and processes that impact performance?
  - How are these processes enacted? How do they unfold?

• How does information shape activity? How does activity shape information?

• Bug Report Networks: How information networks structure social activity?
Our research

Bug report networks
**Objects of study in F/OSS research**

<table>
<thead>
<tr>
<th>Objects</th>
<th>Success measures</th>
<th>Critical driving factors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Artifacts</strong></td>
<td>Quality, reliability, usability, durability, fit, ...</td>
<td>Size, complexity, software architecture (structure, substrates, infrastructure), ...</td>
</tr>
<tr>
<td><strong>Processes</strong></td>
<td>Time, cost, complexity, manageability, predictability, ...</td>
<td>Size, distribution, collaboration, knowledge/information management, artifact structure, ...</td>
</tr>
<tr>
<td><strong>Communities</strong></td>
<td>Ease of creation, sustainability, trust, social capital, ...</td>
<td>Size, economic setting, organizational architecture, behaviors, incentive structures, ...</td>
</tr>
<tr>
<td><strong>Knowledge</strong></td>
<td>Creation, use, need, management, ...</td>
<td>Tools, conventions, norms, social structures, technical content, ...</td>
</tr>
</tbody>
</table>

- RI should support variety, and allow for extension
Current research approaches

- Large-scale quantitative cross-analyses
  - Code size, code change evolution, group size, composition and organization, development processes

- Small-scale qualitative case studies
  - Specific processes and practices, hypothesis development and testing

- Main issues:
  - Scalability
  - Richness

- RI should facilitate articulation of the two sides
## Data requirements

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Reflect reality</td>
<td>Empirical and natural</td>
</tr>
<tr>
<td>• Adequate coverage</td>
<td></td>
</tr>
<tr>
<td>• Representative level of variance</td>
<td>Sufficient size and variety</td>
</tr>
<tr>
<td>• Statistical significance</td>
<td></td>
</tr>
<tr>
<td>• Comparable results</td>
<td>Common frameworks and representations (sharable)</td>
</tr>
<tr>
<td>• Repeatable, testable, extendable</td>
<td></td>
</tr>
</tbody>
</table>
## Data available

<table>
<thead>
<tr>
<th>Variety of</th>
<th>Types</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content</strong></td>
<td>Communication</td>
<td>Discussion forums, newsgroups, chats, community digests, ...</td>
</tr>
<tr>
<td></td>
<td>Documentation</td>
<td>HOWTOs, FAQs, user and developer documentation, tutorials, ...</td>
</tr>
<tr>
<td></td>
<td>Development</td>
<td>Source code, bug reports, design documents, ...</td>
</tr>
<tr>
<td><strong>Medium</strong></td>
<td>Communication</td>
<td>Mailman, Phpbb, ...</td>
</tr>
<tr>
<td></td>
<td>Source control</td>
<td>CVS, Subversion, Bitkeeper, ...</td>
</tr>
<tr>
<td></td>
<td>Issue tracking</td>
<td>Buzilla, Scarab, Gnats, ...</td>
</tr>
<tr>
<td></td>
<td>Content mgt.</td>
<td>Wiki, Plone, ...</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>Project sites</td>
<td>Mozilla, Linux, KDE, Gnome, Gimp, ...</td>
</tr>
<tr>
<td></td>
<td>Community sites</td>
<td>Slashdot, Newsforge, FSF, ...</td>
</tr>
<tr>
<td></td>
<td>Repositories &amp; indexes</td>
<td>SourceForge, Freshmeat, Tigris, ...</td>
</tr>
</tbody>
</table>

- Data available as byproducts, not generated for research
Issues with empirical data

- Discovery and selection
- Access and gathering
- Cleaning and normalization
- Linked aggregation
- Evolution
Issues with empirical data

Cleaning and normalization

- Bug report normalization
  - Multiple formats of the “bug report” object (Bugzilla, Scarab, ...)
  - What information is necessary for research? (and is that information readily available?)

- Bug reference normalization
  - Various types of references: How do we normalize them?
    - E.g.: depends on, blocks, duplicate, ...
  - Some of them not formalized: How do we mine them?
    - E.g.: see also, related, ...
Issues with empirical data

Linked aggregation

- BRN complete only if multiple repositories are aggregated

- Some issues span across multiple repositories
  - Gnome & Red Hat: Who's got responsibility for a bug?
  - Debian, Gentoo bug posting instructions

- The need for aggregation is two way:
  - Same tool, different projects
  - Same project, different tools
Components of a research infrastructure

- Representation standards
- Metadata
- Tools (downstream & upstream)
- Centralized data repositories
- Federated access
- Processed research collection
- Integrated data-to-literature environments
Components of a research infrastructure

Representation standards

- Bug report XML representation

- Abstracted properties
  - Smallest or largest common denominator?

- Additional information for research purposes
  - Metadata
  - Mined/inferred properties

```xml
<!ELEMENT bug_report ( id, alias?,
  creation_ts, last_modification_ts,
  status, resolution, product, component,
  hardware_list, os_list, version_list,
  severity, priority, target_milestone,
  reporter, responsible_party, qa_contact,
  cc_list, manifesting_url, summary,
  status_whiteboard, keywords,
  dependency_list, attachment_list,
  vote_list, comment_list,
  bug_activity_transaction_list,
  provenance )>

<!ATTLIST bug_report id ID #REQUIRED>

<!!-- Identification -->
<!ELEMENT id ( #PCDATA )>
<!ELEMENT alias ( #PCDATA )>

<!!-- Timestamps -->
<!ELEMENT creation_ts ( %timestamp; )>
<!ELEMENT last_modification_ts ( %timestamp; )>

<!!-- Properties -->
<!ELEMENT status ( #PCDATA )>
<!ELEMENT resolution ( #PCDATA )>
...
```
Components of a research infrastructure

Tools

- Extraction of bug cross-references
  - 100% of formalized references are automatically minable
  - 40-70% of non-formalized references are minable (regex) but hard to automatically categorize
  - Remaining % require help of a human

- Three possible approaches:
  - Facilitate human mining (downstream)
  - Improve automated extraction tools (downstream)
    E.g.: more complex regex, NLP
  - Increase formalization at creation time (upstream)
Recommendations

- Refine knowledge of F/OSS research needs
- Exploit experience from other domains
- Develop data selection policies
- Develop data standards
- Instrument studied tools
- Create federation middleware
- Create prototypes
Conclusions

Research infrastructure might increase collaboration and lower “entry cost” of doing F/OSS research, but:

- Is there a sufficient drive for a common infrastructure?
  - What are the common questions?
  - What are the common needs?

- Risk of limiting research to “low hanging fruits”
  - Features easy to measure and extract
  - Many studies on few common corpora
  - Same underlying assumptions about data