An Exploration of Sustainability Thinking in Research Software Engineering

Timo Kehrer  
Humboldt-Universität zu Berlin, Germany

Birgit Penzenstadler  
California State University, Long Beach, USA

7th Intl. Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy)  
August 20th, 2018, Banff, Canada, at RE’18
Personal experience with research software development I: Henshin

» Turning graph transformation concepts and theory...
Personal experience with research software development I: Henshin

... into an experimental tool suite for Model-Driven Software Engineering

- Expressive transformation language with a graphical syntax
- Control-flow constructs with parameter passing
- Support for endogenous and exogenous transformations
- Arbitrary mixing of different graph transformation styles (DPO/SPO)
- Efficient interpreter engine based on constraint solving
- Verification using state space tools
- Code generator for Apache Giraph

https://www.eclipse.org/henshin/
Personal experience with research software development II: SiDiff/SiLift

» Model management operators supporting model evolution in MDE
Personal experience with research software development II: SiDiff/SiLift

» Model management operators supporting model evolution in MDE

» Model matching and differencing
» Semantic lifting of low-level differences
» Consistency-preserving patching and merging
» Model refactoring and restructuring
» Semi-automated model repair
» Change impact analyses
» ...

http://pi.informatik.uni-siegen.de/Projekte/SiLift/
More generally: Research software in (Software) Engineering

» In SE, empirical validation of new methods and techniques requires prototypical implementations and tools (the research software).

» These tools are being developed for the sake of gaining scientific insights.

<table>
<thead>
<tr>
<th>Type of question</th>
<th>Submitted</th>
<th>Accepted</th>
<th>Ratio Acc/Sub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method or means of development</td>
<td>142 (48%)</td>
<td>18 (42%)</td>
<td>(13%)</td>
</tr>
<tr>
<td>Method for analysis or evaluation</td>
<td>95 (32%)</td>
<td>19 (44%)</td>
<td>(20%)</td>
</tr>
<tr>
<td>Design, evaluation, or analysis of a particular instance</td>
<td>43 (14%)</td>
<td>5 (12%)</td>
<td>(12%)</td>
</tr>
<tr>
<td>Generalization or characterization</td>
<td>18 (6%)</td>
<td>1 (2%)</td>
<td>(6%)</td>
</tr>
<tr>
<td>Feasibility study or exploration</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
<td>(0%)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>298 (100.0%)</td>
<td>43 (100.0%)</td>
<td>(14%)</td>
</tr>
</tbody>
</table>


» Also applies to other engineering sciences, at least to those which heavily rely on computer-aided analysis and design principles.
Recurring problems with research software

» Distinguishing characteristics
  » Short educational cycles (BSc, MSc and PhD projects)
  » Intermediate academic funding structures (grants)
  » Driven by academic recognition incentives (papers)
  » Extremely distributed development (silos and geographically)

» Rapid research software aging vs. long-term research progress
  » design knowledge gets lost quickly
  » repeated re-implementation of legacy software components
  » research results are often irreproducible

» Actual usage often limited to individual projects
The quest for sustainable (research) software

Existing initiatives and related work:

» Software maintenance and evolution (e.g., ICSME but also prominent at ICSE, ESEC/FSE, ASE, ...)

» Software engineering for sustainability (e.g., guiding theme of ICSE'12)

» Scientific software engineering (e.g., 2016 Dagstuhl perspectives workshop on engineering scientific software)

» Sustainable software for science (e.g., WSSSPE workshop series)

» ...
Our approach: A generic sustainability goal model...

[Penzenstadler and Femmer 2013]

Sustainability of research software

Specific viewpoint (individual, social, economic, technical, environmental)

Expects what a value means for a specific instance

Rationale that is rooted in itself

A means to support a set of objectives, influencing a subset of the indicators

A qualitative or quantitative metric to approximate an objective
... and its instantiation for research software (in MDE)
... and its instantiation for research software (in MDE)

... with the aim of answering three **research questions**:

1) Was does sustainability mean for research software?
2) Which sustainability objectives can be identified for research software?
3) What are the ways in which we could progress towards achieving those objectives?
Zooming in (individual, social and economic dimension)
Limitations

» Evidence w.r.t. applicability and effectiveness of the model
  » Applicability in practice?
  » Validation of objectives and activities?

» Generalizability of the model
  » What about other research software from
    » the field of Model-Driven Engineering
    » the broader field of software engineering
    » other engineering sciences
    » the computational sciences (e.g., natural, economic, social and life sciences)

» Completeness of the model
  » Important values (and according objectives, activities and indicators) may be missing, e.g., in the data-driven sciences:
    » Reproducibility of research results, long after initial publication
    » Continuous evolution and maintenance of computational research software
Limitations and future work

» Evidence w.r.t. applicability and effectiveness of the model
  » Applicability in practice?
  » Validation of objectives and activities?

» Generalizability of the model
  » What about other research software from
    » the field of Model-Driven Engineering
    » the broader field of software engineering
    » other engineering sciences
    » the computational sciences (e.g., natural, economic, social and life sciences)

» Completeness of the model
  » Important values (and according objectives, activities and indicators) may be missing, e.g., in the data-driven sciences:
    » Reproducibility of research results, long after initial publication
    » Continuous evolution and maintenance of computational research software

Application and assessment

Transfer to other kinds of research software and scientific domains
Recurring problems with research software

- Distinguishing characteristics
  - Short educational cycles (BSc, MSc and PhD projects)
  - Intermediate academic funding structures (grants)
  - Driven by academic recognition incentives (papers)
  - Extremely distributed development (silos and geographically)

- Rapid research software aging vs. long-term research progress
  - Design knowledge gets lost quickly
  - Repeated re-implementation of legacy software components
  - Research results are often irreproducible

- Actual usage often limited to individual projects

The quest for sustainable (research) software

Existing initiatives and related work:

- Software maintenance and evolution (e.g., ICSE but also prominent at ICSE, ESEC/FSE, ASE, ...)
- Software engineering for sustainability (e.g., guiding theme of ICSE12)
- Scientific software engineering (e.g., 2016 Dagstuhl perspectives workshop on engineering scientific software)
- Sustainable software for science (e.g., WSSSPE workshop series)

... and its instantiation for research software (in MDE)

... with the aim of answering three research questions:

1) What does sustainability mean for research software?
2) Which sustainability objectives can be identified for research software?
3) What are the ways in which we could progress towards achieving those objectives?