

Policy and Advocacy -

- URSSI would do two things:
 - Collect data around research software
 - Analyze that data in order to advocate for particular issues
- (NB: both of these are about empowering other people to help achieve URSSI's mission)

What fits policy?

- Almost everything that is hard for URSSI to do directly
 - E.g software credit & citation, career paths, sustainability models,

Policy challenges

- Selectively address areas where change/impact is possible during URSSI
 - More specifically: What issues are not being advocated for... OR, what is URSSI uniquely positioned to advocate for...
- Perhaps also invest a little in a few long-term activities
- May not be possible to know which is which without some data
- Want to leverage community whenever possible - get help with gather data, need help with advocacy
- Also want to leverage fellows - light paid 'staff' who already want to work on this, but need a bit of funding/title
- Portfolio balance:
 - Short term more risky investments
 - Long-term important / critical issues that effort should be sustained for change.

How would URSSI get data?

- Existing grant databases
- Surveys
- (in other words collecting data ourselves)
 - URSSI could also be seen as a place for data ...

Good strategy, bad strategy book

- Not just picking a strategy, but picking them in a clever way so that they overlap and reinforce each other.
- How to be effective and uses activities that are already happening
 - Kernel of a good strategy (three elements)

- Diagnosis
- Guiding policy
- Coherent action

Questions:

- What topics could URSSI policy activities address?
- For each
 - What's the timeframe for possible impact?
 - Scale of impact that's expected? (low medium high)
 - Which policy phase(s)? (data collection, analysis, advocacy)
 - Who else needs to be involved?
 - What community resources can be leveraged?
 - What amount of work is needed by URSSI, and by what kind of URSSI staff member?
 - Would this happen without URSSI?

Three strands: (impacts in different spaces)

- People
- Software
- Software ecosystem

People needed:

- Some fraction of a community manager/policy officer (organize, negotiate with stakeholders) [community manager might be shared with other URSSI functions]
- Some expert writers on topics (communications specialists)
- One data analyst/data scientist
- Maybe organized as small group who define projects, publicize them, get feedback, work on them, get others to work on them
-

Work with other organizations? (ECAR, etc.)

People

- **Knowing how to do software development in research (standards? practices?)**
++++++
 - Could be developed into Carpentries material
 - Some material exists in pieces today at IDEAS, BSSw, ...
 - Define roadmap (few months)
 - Map existing material to roadmap, find gaps (few months)

- Get Carpentries to fill gaps (easy)
- Advocacy part could be getting institutions and stakeholders to accept that these skills are important
- Influence computing centers to define a policy that defines something that relates to software quality. Enabling people access by have a 'gate' around software practice
- Propagation of software development practices in research ++++
 - This would also cover understanding of how many people this was reaching
- Enable complete pathways for promotion in research software related jobs up to Director level (and more than just RSEs) +++
 - Including equal footing with research career tracks ++
- Increase underrepresented minority involvement in research software
- Adoption of software engineering research in research software development as a research discipline +++
- Funding for non-postdocs in grants to do software development +
- Funding (models) more than just novel/new software development (e.g. maintenance, reuse) ++
- Funding for community management and other non-developer roles +++
- Require RSE-mentoring plans in proposals, like postdoc mentoring plans
- Advocate for accessible tools & resources that allow everyone to participate in software
- Help researchers make better software adoption decisions (i.e. less greenfields wheel reinvention) +++
- Measure the amount and quality of software development training in each discipline across US institutions at undergraduate and postgraduate levels ++
- Training in interpersonal skills
- Document and explain the benefits of open sourcing software to research careers +++++
 - Create policy for when it is and is not appropriate for open-source decision
- Collect and circulate sample language on the impacts of software work for tenure and promotion letters ++++++
- Collect tenure and promotion guidelines and highlight how software is evaluated; suggest improvements +
- Encourage the National Academies to study the role of software work in careers and make recommendations on how to improve careers ++

Software

- Expert software reviewers on grants / review panels +++++++
- Explicit statements on how proposals are addressing software best practice (software management plans) ++

- Encourage funders to require/encourage software they fund to be open source +++++
- Encourage funders to adopt a sunset policy for software
- Encourage funders to make conscious decisions about which software to sustain and which to sunset ++
- Document and advocate for use of best practices (in SE) +++++++
- Encourage succession planning for software projects
- Analysis of different software sustainability models and publishing of data around this +
- Understand research software project stages and what is required to go from stage to stage (e.g. code of conduct etc, cf Fogel's work on OSS projects) +

Software ecosystems

- Understand/highlight the role and amount of software in research +
 - Universities are becoming software development shops, need to convince them of this
- Advocate for universities to have software development centers (could involve libraries), which help researchers develop better software
 - URSSI could get funding from IMLS to study this, how libraries could help with software development practices, archiving, preservation, etc.
 - <https://www.imls.gov/grants/available/laura-bush-21st-century-librarian-program> (that would be the relevant funding mechanism)
- Connecting software development and archiving with libraries (leveraging the libraries) - Encouraging libraries to support software development and archiving +++
- Get Campus IT centers to offer more support for research software development ++
- Encourage software artifacts to be required as part of all publications +
- Define and help research funders understand critical software infrastructures (e.g. dependencies that if they go away will lead to software collapse) +++++++
- Introduce SE courses in science departments +
- Responsible Conduct of Research training that includes software development, and software ethics ++
 - Put together a checklist around what it means to responsibly develop research software (6 months, including community buy-in time)
 - Survey or study to see how much of the checklist is in use at present (6 months)
 - Identify the places where this isn't used much, but there is a stakeholder who feels it's important (1 month)
 - Consult with stakeholders around it
 - Advocate for this list to be used in funding and publication decisions (6 months or more)

- Build training around the list (a few months, perhaps much longer to get this into existing RCR training)
- Get community buy-in
- Use as framework an existing set of rules for ethics in engineering
- Advancement of software as being integral to research ethics
- Address intellectual property rules for software within universities - encourage them to promote open source and sharing
- Develop an URSSI community (or URSSI-aligned communities within various disciplines) that defines what is adequate for software to be "labeled" as part of the community - basic checklists, as well as deeper standards, on topics like documentation practices, CoC, ...++++
- Define FAIR (or some other rubric) for software, and make it required by various stakeholders (funders, publishers, repositories, etc.) +++