CANARIE Research Data Software Workshop
Breakout Groups

1. Hackathon
    a. A COE could report on the results of the Hackathon.
    b. One possible approach would be for each project to identify 1 weakness that the group could then discuss to help each other arrive at a solution.
2. Building Software for Re-Use
    a. Build Community
        i. We don't know if people are re-using it (VESTA), was one comment made, which speaks to this issue. How do you build
    b. Build independant/modular services. (Separation of Concerns - SOC - https://en.wikipedia.org/wiki/Separation_of_concerns)
        i. A good idea to build services/modules that can be used in other contexts.
        ii. In the case of VESTA they are building APIs for each component.
        iii. Developers need to be able to extract the services they need easily.
    c. Build to APIs Inside and Out
        i. Use APIs internally as well as externally. Don't build in one way internally but differently for the external open source community.
        ii. Good practice would be to have a Web API for testing even if an internal API is used.
    d. Adopt CI/Automated testing approach
        i. Require developers to create Unit tests.
        ii. Then use a system like Jenkins to perform the full testing.
    e. Adopt an automated approach to generating documentation.
        i. Systems like Swagger, Sphinx and DOxygen. Sphinx and DOxygen work together.
    f. Create galleries/tutorials and make it easier to enhance or work with the code.
        i. Simple demos with code examples.
        ii. Make sure to expose the examples so they are obvious and not buried somewhere. Best in the README if in github, even better if toward the top of the README.
    g. Use open standards for storage and retrieval.
        i. In the case of the SensorUp project they are using the OGC standard, which provides interoperability for the Internet of Things. It has a data model (data and metadata) that facilitates storage and access.
        ii. Another example is Climate Change, where NetCDF which is
        iii. Some standards are complex, so implementing them can be a challenge. This is a difficult issue and could be facilitated by a body like a COE.

iv. Create examples of how “domain-specific” standards could be used in other disciplines.

h. Commit often and make sure the code is clearly being worked on. More commits suggest to others that the project is active. More contributors similarly suggests greater community support.

i. Make the decision to make it open source and commit to maintaining the project.

j. Develop community standards for coding and participation.

i. Ensure that code standards are good, but not too strict that people will leave the project.

k. Choose a name for the project that encourages participation based on the core functions.

l. Consider using Workflow tools to automate back-end processes.

i. Systems like Kepler and Taverna.

3. Centre of Excellence

a. A COE could suggest best practices for developers when it comes to what to adopt. (g from above -- contributor standards, d from above -- open standards, templates for github README.md, template for github contributing.md)

b. Could look to “sanction” or build a generic research platform that could bridge tools that are being developed in domain-specific contexts.

c. Promote a standard environment based on the development environment.

i. Including build tools - staying on top of things like build tools and making recommendations.

d. Create a network of PIs and knowledgeable people that can bring multiple strengths to a COE.

e. Bring the different funders and funded projects together to share experiences.

f. Develop a publication that focuses on articles about building research software.

i. Facilitate the promotion and knowledge of research platforms from the Canadian and international contexts.