Starter Kit
Local Research Software
Support: Competitive Funding
Call 1

Support: Scott Henwood, Director, Research Software Program
scott.henwood@canarie.ca
# Table of Contents

Background ................................................................................................................................................... 4  
  Why Local Research Software Support? ................................................................................................... 4  
Typical Team Activities .................................................................................................................................. 7  
Designing the Team ...................................................................................................................................... 8  
  Fitting into the Organizational Structure .................................................................................................. 8  
  Team Organization .................................................................................................................................... 8  
  Team Onboarding .................................................................................................................................... 8  
  Rules for Selecting Projects ....................................................................................................................... 9  
Engaging the Research Community ........................................................................................................ 10  
Demonstrating Value to the Institution ...................................................................................................... 11  
  Project Metrics ........................................................................................................................................ 11  
  Reuse Metrics ......................................................................................................................................... 12  
  Research Outputs ................................................................................................................................... 13  
  Institutional Metrics ................................................................................................................................ 13  
  Grants ...................................................................................................................................................... 13  
  Qualitative Metrics .................................................................................................................................. 14  
Working with Existing Research Computing Staff ....................................................................................... 15  
Working with IT ........................................................................................................................................... 15  
Other Resources ......................................................................................................................................... 16  
Appendix 1: Research Software Steering Committee Charter ................................................................... 17  
Appendix 2: Team Lead Job Description (1) ............................................................................................... 19  
Appendix 3: Team Lead Job Description (2) ............................................................................................... 21  
Appendix 4: Research Software Developer Job Description (1) ................................................................. 24  
Appendix 5: Research Software Developer Job Description (2) ................................................................. 26  
Appendix 6: Sample Proposal Template (1) ............................................................................................... 28  
Appendix 7: Sample Proposal Template (2) - With Scoring Criteria ........................................................... 29  
Appendix 8: Sample Rules for Selecting Projects (1) .................................................................................. 32  
Appendix 9: Sample Rules for Selecting Projects (2) .................................................................................. 33  
Appendix 10: Sample Rules for Selecting Projects (3) ................................................................................ 35
Background

For more than a decade, CANARIE’s Research Software program has been funding researchers to create the purpose-built software they need to carry out their research. In the early days of the program, relatively few researchers needed this type of software and significant inroads were made in supporting the Canadian research community. Today, as more and more research relies on computational components, funding individual projects is not always the best option.

Following models successfully deployed in the UK and other European countries, in 2018 CANARIE embarked on a pilot program to support researchers directly with software developers located at the institution. Known as the Local Research Software Support Pilot, CANARIE funded teams of up to 3 dedicated, full-time research software developers at 3 participating institutions. These developers are available to all researchers at the institution, regardless of discipline, to provide guidance, training, and expertise in development of software to support research.

This document describes the lessons learned from the pilot and provides resources that may be helpful for other institutions in establishing their own Local Research Software Support Teams.

Why Local Research Software Support?

It’s all about getting researchers to their research faster. Funding for research software is often targeted at large projects where the funding level is sufficient to allow software developers to be hired for long-term assignments. This is great for this type of project, but there are a number of issues not addressed by this type of funding:

*Not all researchers need a large software system to complete their research.*

Some require only small amounts of new development, maintenance, or integration, which doesn’t justify hiring a software developer. Often such work is done by grad students whose career focus may not be software development. This is not optimal for the institutions because...

*Software for research is highly reusable.*

CANARIE’s 12+ years of funding and supporting Research Software has shown us that software to support research is highly reusable. This may seem counter-intuitive, but researchers often need the same types of tools to support their workflow: search, storage management, compute scheduling, visualization and user authentication/authorization. Even the science-facing software has potential for cross-discipline reuse. By reusing or adapting existing software instead of undertaking from-scratch development, researchers can get to their research sooner with more funds spent on research and less on software development.
In order to be reusable, research software has to be well-architected, well-written, fully tested, and properly documented – in short, it has to be developed with software engineering best practices in mind. As a group of professional software developers, the members of the Local Research Software Support Team are ideally suited to support reuse. But...

**To be reused, it has to be findable**

Experience has shown that once you start to look for it, there is a lot of software for research available at most institutions. This software has significant value in reuse potential, but the problem is finding it. Most institutions do not have centralized registries for research software, and for those that do, there is usually not a person who understands what’s available. As the Local Research Software Support team engages members of the research community, they will have the opportunity to find, understand and catalog existing software.

This team can provide other software reuse benefits to the institution as well, because...

**Reuse extends to software not written at the institution.**

Because research software projects tend to be similar, the Team will become familiar with software components developed outside the institution (open source libraries, for example) that have general applicability to research. By using such software when appropriate, the Team can become expert users, reducing development time for the next project. Contrast this to the model where research software is developed by a grad student or postdoc in isolation. Once the project is finished, the risk is that knowledge will be lost.

This reuse paradigm can be extended through contact with other Local Research Software Support teams. One of teams funded by CANARIE in the pilot formed a “community of practice” at their institution. This community consisted of the Team members plus other interested individuals at the institution whose job function involved writing research software. The purpose is to learn from each other with a goal of furthering software reuse.

Reuse can be furthered through engagement between Local Research Software Support Teams at different institutions. Perhaps the Team at another institution has a developer who is an expert in 3D visualization, and your Team needs some 3D visualization software. Guidance from the other Team would shorten the software development effort. One of the pilot teams developed a workflow manager component that they used themselves on multiple projects. This component is now available to other pilot Teams to use in projects where it makes sense.

Of course, in order to be reusable long-term...

**Research Software must be maintained**

Even funding for large software projects often doesn’t take maintenance into account. What happens when initial development is complete and the software team has been disbanded, and a bug is
uncovered, or a new feature is required? With a Local Research Software Support Team, software developers are available on a short-term basis to do this type of work without the overhead of hiring someone. If the Team has previously encountered the software to be maintained, through either their development or cataloging activities, ramp-up times will also be shortened.

_Researchers new to software development need support too._

In disciplines where the research doesn’t traditionally rely on purpose-built software, researchers may not even be aware of how existing software can help support their work. The Local Research Software Support Team is ideally placed to help introduce such researchers to the world of built-for-research software. Similarly, for researchers who have traditionally used laptops and desktops for their research, the Team can support them as they transition to advanced research computing.
Typical Team Activities

Whomever is funding the Team may place additional restrictions on Team activities, or define others, but the following is a reasonable starting point:

1. **Software development/integration/maintenance**: First and foremost, the Team is a group of professional software developers who are available to develop, integrate, modify, and configure software to support academic research.

2. **Training researchers**: In disciplines where the research doesn’t traditionally rely on purpose-built software, researchers may not even be aware of how existing software can help support their work.

3. **Training software developers**: Rather than implementing large research software systems themselves, Local Research Software Support staff can provide guidance to the software developers dedicated to the project. Often, these developers are experts in the research discipline but not in software engineering best practices/techniques. Teams may assist in training members of the project team and providing consulting services on topics such as designing for reuse and setting up development environments.

4. **Cataloguing software already available at the institution**: The first step in reducing software duplication through reuse is to understand what software already exists. Team members are able to meet with researchers from different departments, understand what software is available, and catalogue it in a central repository.

5. **Recommending reuse of existing software**: By having a high-level understanding of research software that is available from either the institution, Local Support Teams at other institutions, or even third-party libraries and platforms commonly used for research, Local Research Software Support staff can advise researchers and their software developers on the reuse potential of existing solutions.
Designing the Team

Consider building the team to up to three (3) members, comprising a team lead and two other developers, within the first year. Previous experiences in Canada and in other countries indicate that the team will be chronically over-subscribed, so a plan to expand the team after the first year would be beneficial.

Fitting into the Organizational Structure

Although not always the case, Local Research Software Support Teams are typically formed as part of an existing group that already provides IT services to the research community. This would include the existing IT organization or Advanced Research Computing. Teams may also fall under the auspices of the Vice President, Research. A few international teams have been formed within specific faculties. This approach is not ideal as such groups typically end up supporting only researchers from that faculty.

Team Organization

Consider assigning a senior person within the institution (e.g. Director-level) to have overall responsibility for the Team. This person should be involved heavily in Team start-up (recruiting, defining processes) and should be in a position to promote the Team’s activities and value to the institution’s leadership.

Consider forming a Steering Committee to guide the top-level direction of the Team. This is an excellent way to involve the research community with the Team.

Resource:
Appendix 1: Sample Local Support Steering Committee Charter

Experience from the international community indicates that Local Research Software Support Teams are most effective when they are a team. This means that Team members should sit together. It also means that they should interact with each other frequently, so having a Team member be dedicated to a project full-time for a long period is not optimal. In such a case, it is advantageous to put two members of the Team on a project part-time rather than one full-time.

Team Onboarding

Consider onboarding the Team Lead first. This person can then assist with Team start-up activities and recruiting the other Team members. Teams are initially typically small, making it necessary for the Team Lead to be an active participant in supporting researchers. As the Team expands, there may be an opportunity for the Team Lead to move into more of a managerial role. Experience in other countries
suggest that having this type of career path is vital for retaining highly skilled research software developers.

**Resources:**
- Appendix 2: *Team Lead Job Description Example (1)*
- Appendix 3: *Team Lead Job Description Example (2)*
- Appendix 4: *Research Software Developer Job Description Example (1)*
- Appendix 5: *Research Software Developer Job Description Example (2)*

### Rules for Selecting Projects

The first task of each Team is to develop a set of rules that how/what projects will be selected. This should be settled before the Team is introduced to the research community. These rules should address:

- **How researchers at the institution will request the services of the Team**
  - Participants in the pilot had success with a “call for proposals” process, where the Team announces a formal call to which researchers respond

**Resources:**
- Appendix 6: *Sample Proposal Template (1)*
- Appendix 7: *Sample Proposal Template (2) - With Scoring Criteria*

- **How such requests will be evaluated/selected**
  - Formal evaluation and scoring criteria are useful here. Ultimately, the projects selected should be of value to the institution.
  - Certainly, the Team could evaluate proposals themselves, but if you’ve set up a Steering Committee, it might be beneficial to get that group involved in project selection.

**Resources:**
- Appendix 8: *Sample Rules for Selecting Projects (1)*
- Appendix 9: *Sample Rules for Selecting Projects (2)*
- Appendix 10: *Sample Rules for Selecting Projects (3)*

- **Pitfalls to avoid**
  - Spending too much time on a single project.
  - Not acting as a team – better to put 2 people on a project part-time than one full-time.
  - Doing too many projects with the same small group of researchers.
  - Taking on work that is best done by other, existing groups. This is particularly easy to do if the Team is part of another group, but to demonstrate their value, areas of responsibility should be clearly delineated.
  - Not clearly defining when a project is “done”.
Engaging the Research Community

Once the Rules for Selecting Projects are in place, the teams will be able to announce themselves and their services to researchers at the institution. This could also be an opportunity to start identifying and cataloguing software already available locally.

An effective way to do this is by sending out a survey. A well-crafted survey will not only introduce the team to researchers, but it will also allow the team to better understand the specific needs of the local research community.

Note that there is such a thing as too much engagement. Teams are typically oversubscribed immediately and can become irrelevant if there is a perception that no one can ever get access to their services. One of the teams in the pilot only announced calls for proposals during the summer and at other times where there were fewer researchers on campus.

Resources:

Appendix 11: Sample Researcher Survey (1)
Appendix 12: Sample Researcher Engagement Plan (1)
Appendix 13: Sample Researcher Engagement Plan (2)
Demonstrating Value to the Institution

In order to continue to be funded by the institution, Local Research Software Support Teams must demonstrate their value. Unfortunately, the international community did not initially concentrate on metrics and evidence, so this is somewhat new ground.

As part of a survey to introduce the Team to researchers, consider trying to determine how many people at your institution are already involved in writing software to support research. A large community suggests that there could be economies in coordination.

Most teams will require institutional funding at some point, and experience in other countries has shown that this funding is much easier to obtain if the Team has “champions” within the institution’s research community. Consider identifying such champions early and adding them to the Steering Committee.

The participants in the Local Research Software Support Pilot are collecting the metrics below. Note that the metrics that demonstrate the value of the Team to the institution are generally longer-term metrics which will require the participation of researchers the team has previously worked with to collect, possibly over a period of a few years. Consider adding the collection of such metrics to agreements with researchers.

**Project Metrics**

<table>
<thead>
<tr>
<th>Metric: Projects Completed</th>
<th>Description: A list of projects the Team has successfully completed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection Method:</td>
<td>Simple enumeration</td>
</tr>
<tr>
<td>Purpose:</td>
<td>Demonstrate impact of Team on university research community</td>
</tr>
<tr>
<td>Notes:</td>
<td>It will be up to each Team to define when a project is “complete”.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric: Projects Declined</th>
<th>Description: A list of projects requested by researchers that the Team has declined to work on.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection Method:</td>
<td>Simple enumeration</td>
</tr>
</tbody>
</table>
| Purpose:                  | - Fine tune communications with researchers  
|                           | - Demonstrate need for team  
|                           | - Team management                                                                                  |
| Notes:                    | It would be useful to indicate why each project was declined. For example, receiving requests for projects that are not appropriate may indicate that the messaging to researchers needs to be adjusted. Similarly, declining many projects while completing relatively few may indicate that the team is taking on projects that are too large and may be better off consulting with project staff. |
### Projects Abandoned

<table>
<thead>
<tr>
<th>Metric</th>
<th>Projects Abandoned</th>
<th>Description:</th>
<th>Purpose:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>A list of projects that the team started working on but stopped prior to completion.</td>
<td></td>
<td>- Fine tune communications with researchers</td>
<td>It would again be useful to track why each project was abandoned, to assist in continuous improvement activities.</td>
</tr>
<tr>
<td>Collection Method:</td>
<td>Simple counting</td>
<td></td>
<td>- Analysis for continuous improvement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Team management</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Disciplines

<table>
<thead>
<tr>
<th>Metric</th>
<th>Disciplines</th>
<th>Description:</th>
<th>Purpose:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>A list of the research disciplines involved in the completed projects</td>
<td>Collect this information from researcher once project has been accepted. Delete if project later abandoned.</td>
<td>- Demonstrate applicability of the Team to the university research community at-large.</td>
<td>We ask researchers to use the NSERC code table when reporting their discipline, but many find that it is not fine-grained enough and make up their own.</td>
</tr>
</tbody>
</table>

### Reuse Metrics

These metrics track the time savings realized by reusing existing software. Institutions may also choose to track the associated cost savings. This is really one metric, but as there are different aspects to software reuse, separate entries are listed.

<table>
<thead>
<tr>
<th>Metric:</th>
<th>Time saved by reusing team-developed software</th>
<th>Description:</th>
<th>Collection Method:</th>
<th>Purpose:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric:</td>
<td></td>
<td>This metric measures the time savings realized when software developed/enhanced by the team for one project is reused in other projects. Each time such a piece of software is reused (after initial development/enhancement), the time saved is incremented by the original development time.</td>
<td>Teams will track this using internal controls and may also choose to track cost savings.</td>
<td>Demonstrate the value of the Team to the institution</td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collection Method:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purpose:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric:</th>
<th>Time saved by reusing software the team became familiar with but did not create</th>
<th>Description:</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric:</td>
<td></td>
<td>This metric measures the time savings realized when a member of the Team learns a new software package for one project and that software is used again on subsequent projects. In this case, the cost avoidance is the ramp-up time.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Collection Method: Teams will track this using their internal controls and may also choose to track cost savings.

Purpose: Demonstrate the value of the Team to the institution

Notes: This metric should apply only to software components that most research software developers would not be familiar with. So, for example, the Python interpreter and common libraries wouldn’t qualify.

Research Outputs

Metric: Papers

Description: A list of published papers based on the work of the Team. This includes papers published by the Team themselves as well as papers published by Researchers whose projects the Team contributed to.

Collection Method: Teams will require periodic follow-up with alumni as this is a long-term metric.

Purpose: - Demonstrate the value of the Team to the institution
        - Publicity for the institution

Notes: Suggest that whenever possible, software developed by the Team be assigned a DOI in order to aid in citation.

Metric: Citations

Description: A list of published works that cite the papers described above. If possible, it would be useful to track citations of citations as well, to demonstrate the reach of the Team’s work.

Collection Method: When collecting the “Papers” metric, Teams should note the “Cite As” information and periodically search for citations.

Purpose: - Demonstrate the value of the Team to the institution
        - Publicity for the institution

Notes:

Institutional Metrics

Metric: Publicity for the institution

Description: A list of press releases, blog posts, articles, etc. that reference the Team and the home institution.

Collection Method: Teams will require periodic follow-up with alumni and the institution’s Marketing department.

Purpose: Demonstrates the value of the team to the institution

Notes:

Grants

Metric: Grants

Description: Grants received as follow on to work the Team had already done. Both number of grants and dollar amount would be useful here.
**Collection Method:** Teams will require periodic follow-up with alumni as this is a long-term metric.

**Purpose:** Demonstrate value of the Team to the research community

**Notes:**

**Qualitative Metrics**

These are stories and quotes from the researchers who have worked with the Team to showcase their perception of the Team’s value.

<table>
<thead>
<tr>
<th>Metric:</th>
<th>Short quotes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>Quotes from researchers the Team worked</td>
</tr>
<tr>
<td><strong>Collection Method:</strong></td>
<td>Team to ask researchers for a quote as part of the project completion process</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Demonstrate value of the Team to the research community</td>
</tr>
<tr>
<td><strong>Notes:</strong></td>
<td>Archived internally by the team. As such quotes will have value to others, researchers should be asked for permission to share their quotes more publicly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric:</th>
<th>Case studies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>A long-form paper describing the full process involved in the Team assisting a researcher and the impact of the Team’s work on the research. This would not be done for all projects, just projects with significant team involvement and a big research impact.</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>- Demonstrate value of the Team to the research community - Publicity for the institution</td>
</tr>
<tr>
<td><strong>Notes:</strong></td>
<td>Team to ask researchers to participate as part of the project completion process</td>
</tr>
</tbody>
</table>
Working with Existing Research Computing Staff

The philosophy of eliminating duplication should extend to the duplication of human effort as well. Some institutions have staff dedicated to supporting research computing on local compute resources. Researchers who require access to high performance computing (HPC) may also be assisted by Compute Canada regional support personnel. The Local Research Software Support Team should leverage such staff as they support researchers, rather than duplicate their work. The goal is to augment these groups, not to supplant them.

For researchers who are just moving into computational research: Local Research Software Support teams can play an important role by introducing them to large-scale application development.

For researchers who are transitioning from desktop or cloud to high performance computing (HPC): Local Research Software Support Teams can translate between the language of research and the language of high-performance computing.

A fully staffed Team may itself require assistance or training from HPC or Compute Canada support personnel. To avoid overloading those personnel, it may be necessary for Teams to strategically select which projects they can take on at a given time.

The same approach should apply to training. In some situations, it may be beneficial for Local Research Software Support staff to offer researchers software development training. In circumstances where suitable training is already available through Compute Canada or its regional partners, researchers should be directed to those groups.

Working with IT

Sometimes the research community is not aware of services provided by the IT department. Perhaps IT supports a team communications platform that would be useful to researchers looking to share information with outside collaborators. The Team is ideally placed to recognize such opportunities for internal reuse and standardization. The Team will also be able to help implement institutional cybersecurity policies in purpose-built research software.
Other Resources

- In Europe and elsewhere, Local Research Software Support Teams are known as Research Software Engineering (RSE) Groups. This page, hosted by the UK Society of Research Software Engineering, provides links and resources for several international groups:
  https://rse.ac.uk/community/research-software-groups-rsgs

- A good summary describing why research needs software support:
  https://rse.ac.uk/about/why-research-needs-rses

- Case studies on setting up Local Support Teams:
  https://rse.ac.uk/resources/how-to-start-an-rse-group

- Sample Local Research Software Support Team websites:
  https://www.uregina.ca/is/about/research-software-support/index.html
  http://www.itservices.manchester.ac.uk/research/services/software/
  https://rsgsoton.net

- Research Software Engineering Conferences:
  https://researchsoftware.ca
  https://rse.ac.uk/conf2019
Appendix 1: Research Software Steering Committee
Charter

Background
The Local Research Software Support Team is a group of professional software developers within <institution> that provide development and consulting services for software that supports research activities. The services of this group are available to all researchers, regardless of discipline or faculty.

Role
The role of the Steering Committee is to ensure that the activities of the Team are providing maximum benefit to <institution>.

Committee Membership

Ex-Officio:
1. Local Support Project Authority (PA) (Co-Chair)
2. Local Support Team Lead

Members at Large
Ideally, the committee will include the following types of members from the institution.

1. Researchers from a variety of faculties with varying software knowledge. Such members are selected by the Associate Deans of Research to represent the interests of their faculties and the university as a whole.
2. At least one research software developer from the institution who is not part of the Local Support Team.
3. A member from Research administration.

Terms of Reference
The Committee has the duties, responsibilities and authorities as set out below:

1. Setting the strategic direction of the Team.
2. Monitoring Team activities to ensure value to the institution.
3. Reviewing metrics collected by the Team and suggesting changes to the Team’s mandate as required.
4. Promoting and advocating the benefits of the Team and its services within the institution.
5. Reviewing and selecting applications submitted by researchers requesting the Team’s services.
6. Assisting Team leadership with funding renewal activities, including providing guidance on alternative funding models.

**Procedures**

1. Committee membership shall be at the invitation of the Local Support Project Authority (PA).
2. Members at large shall have an xx-year term.
3. The Committee shall meet at least quarterly, and when required to review applications.
4. The agenda for Committee meetings will include standing items for:
   a. Discussion of Team processes and procedures.
   b. Review of Team activities since the last meeting.
   c. Review of Team metrics.
Senior Research Software Developer

Reporting to the Manager of Research Computing Services (RCS), the Senior Research Software Developer will lead the Research Software Development team to support research application development at <your institution>. The RCS unit (part of Information Technology Services) is responsible to provide computational support, assistance and resources to researchers on campus.

The Research Software Development team is funded through a project from CANARIE to provide researchers at selected institutions with local, centrally deployed full-time research software development support. The goal of this pilot is to assist researchers with software development expertise and knowledge of existing tools to reduce development time, minimize duplication and increase research output.

The incumbent is an experienced software developer having supported or been involved in research projects at an academic institution or in industry. This role will involve working with a varied group of clients including faculty members, postdoctoral fellows, graduate students and other IT professionals on campus. The incumbent will be required to liaise with the research community and other stakeholders to ensure needs are being met and to evaluate gaps in provided services.

**Duties and Responsibilities:**

- Lead a team of qualified developers with the assistance of research computing specialists and other IT professionals to provide research software solutions for researchers
- Lead and participate in software-related activities to enhance or re-use existing software tools for research.
- Manage ongoing research software projects, produce metrics, reports and other documentation as required.
- Provide expertise to researchers on existing software, development environments, languages, libraries, methodologies and team recruitment.
- Survey existing research software and tools
- Work with the Office for Research Initiatives and Services to assist researchers in proposal-writing for software components of research proposals.
- Interface with similar groups at other institutions
- Develop and track metrics to evaluate and report the effectiveness and value of the team to researchers on- and off-campus.
- Be involved in outreach and information sessions to engage researchers to ensure the success of the pilot project.
**Education and Experience Requirements:**

**Education:** Graduate degree in a computationally intensive field or relevant experience. Undergraduate degree in a computationally intensive field with research experience will be considered.

**Experience:** A minimum of five (5) years of software development and research computing experience in a research focused environment with supervisory experience. 3-5 years progressive experience in an IT management position is required.

**Required Skill Sets:**

- Excellent knowledge of programming, development and project management in a complex environment.
- Excellent knowledge of software development methods, frameworks and environments.
- Experience with Advanced Research Computing (ARC) resources and awareness of digital research infrastructure organizations in Canada (e.g. CANARIE, Compute Canada).
- Excellent project management skills and use of tools to ensure projects are completed on time and within established budgets.
- Excellent negotiation, interpersonal skills and ability to explain difficult concepts.
- Research experience.
Appendix 3: Team Lead Job Description (2)

Team Lead Research Support

Position Summary:
As a member of the Communications and Infrastructure Services Group, and reporting to the Manager, Research Server Support, the incumbent acts as an IT technical resource leader and proponent for research software support. As the lead of the Research Software Support Team, the position will lead and direct other programmer analysts in the support of researchers in the discovery, use, re-use, and adaptation of research software. This support will take place through creating awareness, education, software investigation, installation, operation, coding, and debugging. As well, the Team Lead will liaise with other universities and research software entities, report on team activities, assign and assess the work of Programmer Analysts, negotiate agreements with researchers for work activities based on scope, time and cost, and determine which assignments to undertake and potentially provide for ongoing software support. The Team Lead will be an expert advisor and hands-on mentor to the team members with respect to all of their duties.

Duties:
The successful candidate will have significant experience in providing leadership and directing a software development team. You will have superior knowledge of, and ability in, the creation, installation, configuration, management and maintenance of research-related software. The duties include:

- Recruitment and supervision of programmer analysts including assigning and tracking work assignments, work progress, and reporting on team activities.
- Investigating research software applications (e.g. open source, discipline specific repositories, commercial offerings, etc.).
- Meeting with researchers to analyze and understand if a research software request can be undertaken, and, if the request is accepted, match the research needs with a software solution.
- Cooperatively and successfully negotiate agreements with researchers for project outcomes, fees and schedules.
- Working with researchers to install agreed-to applications and provide operating assistance and potentially ongoing support.
- Creating brand-new, or extending, modifying, or integrating existing software as necessary, to meet research needs.
- Conducting seminars, building relationships with researchers, and acting as an advocate for the research software support model.
- Collaborating and cooperating with research support staff from other institutions and the worldwide research software engineering community, including Canada’s CANARIE.
- Contributing work back to the research community as appropriate.
• Contributing to and reviewing grant material related to research software that may include informing budgets, schedules, plans, helping document services available in support of the grant, contributing text, proofing and editing.
• Consulting with researchers and teams on possible research programs/ideas.
• Writing documentation or directing researchers to existing application documentation.
• Consulting with other research developers as needed.
• As appropriate, investigating and using new development tools and environments.
• Promoting the use of existing code libraries, proven software platforms, and good software development practices.
• Ensuring the financial sustainability of the group by defining and building a sound business model and through rigorous financial budgeting and management.
• Other related duties as assigned.

Requirements:

• A Bachelor's degree in Computer Science or Software Engineering plus a minimum of 5-8 years of directly related work experience in performing the above duties. Equivalent education and experience may be considered.
• Proven experience in successfully developing and supporting all aspects of modern software development in a higher-education research environment is key. Experience with modern software development tools and environments is essential.
• Experience with leading, supporting, developing, and coaching individuals in a team setting is a requirement. This experience includes being able to demonstrate employee assessment and performance management skills.
• The incumbent will have a proven, proactive commitment to on-going technical learning and skills development and be able to establish a working environment for team members to do the same.
• Demonstrated ability to apply and use the services of Canada's Advanced Research Computing (ARC) environments. As well, the successful candidate will be able to demonstrate significant knowledge of the use of software libraries and repositories. In particular the use of CANARIE's software is necessary.
• The incumbent must have the demonstrated ability to:
  o Work collaboratively and cooperatively to provide exemplary service and advice to the research community on research software;
  o Understand research needs to match and link software development efforts to the researcher's research program;
  o Be able to communicate clearly, both verbally and in writing;
  o Analyze the needs and requests of the researcher to define, propose, negotiate and deliver successful solutions;
  o Reach set and agreed-to goals while working independently with limited supervision;
  o Analyze, decipher and understand technical problems, and to find, develop, and implement software solutions where there may be little local expertise or existing material;
  o Establish and maintain excellent and effective working relationships;
- Demonstrated knowledge of financial budgeting and management to achieve financial sustainability;
- Demonstrate the ability to define and build a sustainable business model for a software development team.
Appendix 4: Research Software Developer Job Description (1)

As a member of the Research Software Group and reporting to the Team Lead, Research Software Support, the incumbent acts as a technical resource and consultant in the support of University researchers through creating awareness, education, software application discovery, software installation and adaptation, operation, coding, and debugging. The directly supported constituents include faculty members, and research institutes and programs. The successful candidate will be required to become the subject-matter expert for applications to which they have been given responsibility.

Duties

The successful candidate will have significant experience and superior knowledge and ability in the creation, installation, configuration, management and maintenance of research-related software. The duties include:

- Investigating research software applications (e.g. open source, discipline specific repositories, commercial offerings, etc.).
- Meeting with researchers to understand and match research needs with known software solutions.
- Working with researchers to install needed applications and provide operating and ongoing support.
- Creating brand-new; or extending, modifying, or integrating existing software as necessary to meet research needs.
- Contributing work back to the research community as appropriate.
- Conducting seminars, meeting with researchers, and acting as an advocate for the support model.
- Collaborating and cooperating with research support staff from other institutions and the worldwide research software engineering community.
- Contributing to and reviewing grant material related to research software.
- Consulting with researcher’s and teams on possible research programs/ideas.
- Writing documentation or directing researcher’s to existing application documentation.
- Consulting with other research developers as needed.
- As appropriate, investigating and using new development tools and environments.
- Promoting the use of existing code libraries, proven software platforms, and good software development practices.
- Other related duties as assigned.
Requirements

- A Bachelor's degree in Computer Science plus a minimum of 4 years of directly related work experience in performing the above duties. Equivalent education and experience may be considered.
- Proven experience in successfully developing and supporting all aspects of modern software development in a higher-education research environment is key. Experience with modern software development tools and environments is essential.
- An established, proactive commitment to on-going technical learning and skills development.
- Demonstrated ability to apply and use the services of Canada's Advanced Research Computing (ARC) environments
- The incumbent must have the demonstrated ability to:
  - Work collaboratively and cooperatively to provide exemplary service and advice to the research community on research software;
  - Understand research needs to match and link software development efforts to the researcher's research program;
  - Be able to communicate clearly, both verbally and in writing;
  - Analyze the needs and requests of the researcher to define, propose, negotiate and deliver successful solutions;
  - Reach set and agreed-to goals while working independently with limited supervision;
  - Analyze, decipher and understand technical problems, and find, develop, and implement software solutions where there may be little local expertise or existing material;
  - Establish and maintain effective working relationships.
Appendix 5: Research Software Developer Job Description (2)

Job Summary

We are recruiting an enthusiastic developer to join <institution’s> Research Software Team as a Research Software Developer. Working with experienced members of the Team, you will apply software engineering, cloud and high-performance computing, and data science to research across multiple disciplines. Your contributions will enable world-leading researchers to accelerate their discoveries by increasing the quality of their outputs.

Our recent projects range <sample projects/disciplines list> and have employed a broad range of development tools and programming languages. In addition to working on such projects you will also be responsible for promoting excellence in software engineering. This will involve demonstrating and advocating best practices through code and documentation, developing and delivering training material, and performing outreach activities inside the university.

This is a great opportunity to join a dedicated team addressing meaningful challenges. You will be working with world-leading researchers on diverse projects across many scientific disciplines, employing technology to develop innovative solutions. Experience of working in research would be beneficial but is not required – this is a non-academic role and we welcome applicants regardless of background.

Duties and Responsibilities

- Liaising closely with researchers to understand their research and software requirements.
- Contributing to the design, development and maintenance of research software.
- Promoting the activities of the Team where appropriate, including publishing code, contributing to journal articles and giving presentations.
- Participating in activities within the research software development community, both internally and externally.

Requirements

- Bachelor’s degree/college diploma in a relevant subject, or equivalent professional experience.
- Experience of analyzing and solving complex programming problems.
- Full-cycle software development including design, implementation and deployment.
- Significant involvement in at least one substantial software.
- Demonstrable knowledge of one or more languages commonly used for research computing: Python, C++, Fortran, R or MATLAB.
• Experience using Unix-based operating systems, tools and utilities.
• Knowledge of best-practice techniques for software development including automated testing, build automation and continuous deployment, and the use of containers.
• Excellent written and verbal communication skills including the ability to effectively present complex or technical information to a range of audiences.
• Interest in learning about new disciplines.
• Serious commitment to software quality and a very strong attention to detail.
Appendix 6: Sample Proposal Template (1)

If you have any questions or issues regarding the software project proposal application, please email: researchsoftware@yourinstitution.ca.

Section A: Applicant Information

A1. What is your full name? (first and last names)
A2. What is your e-mail address?
A3. What faculty/department are you associated with?

Section B: Software Project Description

B1. Please describe, briefly, the software for which you are seeking support, e.g. What does it do? By whom was it created? In what language is it written? How many people/research groups currently use the software? How is it different from other similar packages that are available if applicable?

B2. Please describe, briefly, the help you are seeking from the Research Software Development Team. E.g., What changes are needed to the software? Who will be working with the team from your research group, and in what capacity?

B3. Please provide a realistic timeline for your proposal, e.g. When will it start? When do you expect it will end? What are the critical milestones?

B4. Please describe what you hope to achieve through this application, e.g. What are markers of success? What are the deliverables?

Thank you for your proposal application.
Notification of initial application outcomes will be delivered on dd/mm/yyyy.
Appendix 7: Sample Proposal Template (2) - With Scoring Criteria

Research Software Development Project – Call for Proposals

Research & High-Performance Computing Support (RHPCS) is pleased to announce the launch of a new Research Software Development Project. The goal of this project is to help researchers refine existing software tools to improve usability and robustness, to disseminate research software beyond the lab in which it is created, and to enhance existing functionality. The Research Software Development Project will provide access to a small team of professional software developers who will work on a part-time (up to 0.4 FTE) basis alongside research groups for 2 to 6 months. One intended outcome of this pilot project is a number of research software packages that can be disseminated for use by other research groups.

Call for Proposals

We are inviting research groups to apply for support from the Research Software Development Team, as outlined below. We will initially be accepting 2 to 3 applications and expect to launch future calls for applications in mm/yy and mm/yy as resources permit.

Interested researchers can apply by completing a short application form that that:

- Describes the problem requiring support from the Research Software Development Team;
- Describes the scope of the required software, how it is intended to be used, and how it might be used by other groups;
- Explains the current status of the software to be developed, including an estimate of the number of current users and an estimate of the number and range of intended users after working with the Research Software Development Team;
- Proposes a strategy for engaging with and supporting the Research Software Development Team to ensure a successful outcome within the intended timeframe.

Applications will be assessed by a selection committee made up of researchers from the RHPCS Advisory and Research Technology Committees using the criteria outlined below. Applicants are invited to discuss their application with the Research Software Development Team prior to submission. The three to five most highly ranked applications will be formally invited to discuss their proposals in more detail with the selection committee who will then pick the two projects they consider to be the best fit for the program.

Timeline

Applications are due at hh:mm on dd/mm/yy.

Notification of application outcomes will be delivered by dd/mm/yy. Those being invited to discuss their
proposals with the selection committee will be expected to do so before the end of mm. Projects will begin after dd/mm/yy.

Questions should be addressed to <email address>

Question 1:

Please describe, briefly, the software for which you are seeking support, e.g., What does it do? By whom was it created? In what language is it written? How many people/research groups currently use the software? How is it different from other similar packages that are available if applicable? (Max 250 words)

Assessment Criteria: Impact; Novelty (10 points)

Scoring:

Impact

- **Used by a single research group:** 1 point
- **Used by a number of research groups across a variety of fields:** 5 points

Novelty

- **Replicates functionality found in other packages:** 1 point
- **Offers unique functionality not available in other packages:** 5 points

Question 2:

Please describe, briefly, the help you are seeking from the Research Software Development Team. E.g., What changes are needed to the software? Who will be working with the team from your research group, and in what capacity? (Max 250 words)

Assessment Criteria: Engagement of research group with team; Fit with skills and capacity of team (10 points)

Scoring:

Engagement

- Little to no detail about how research group will support team: 1 point
- Clear plan for partnership between research group and team that will enable success: 5 points

Fit

- Project requires technical skills not present in team or will take more time than team has available: 1 point
- Team has all the skills necessary to effectively support the project and enough capacity to take on and complete this work according to the proposed timeline: 5 points
Question 3:
Please provide a realistic timeline for your proposal. When will it start? When do you expect it will end? What are the critical milestones?

Assessment Criteria: Feasibility (10 points)

Scoring:

Feasibility

- Timeline lacks detail/milestones or is not realistic: 1 point
- Timeline has enough detail and clearly articulated milestones; is realistic: 5 points

Question 4:
Please describe what you hope to achieve through this application. What are markers of success? What are the deliverables? (Max 250 words)

Assessment Criteria: Clear, measurable indicators of impact; Novelty; Scope (15 points)

Scoring:

Impact

- There are no clearly measurable indices of impact: 1 point
- There are clear indices of impact that will allow us to report on the success of this application: 5 points

Scope

- Successful completion of this project will impact the local research group: 1 point
- Successful completion of this project will impact researchers across a number of different fields of research: 5 points

Novelty

- Completion of this project will result in minor improvements to software: 1 point
- Completion of this project will result in significant improvements to key functionality: 5 points
Appendix 8: Sample Rules for Selecting Projects (1)

We expect that the Team lead will have a key role to play in determining appropriate policies around engagement. Initial thoughts are that researchers will request services of the Team through a formal call for proposals which will occur 2 - 3 times each year (subject to capacity). Evaluation and prioritization of requests will be made by an adjudication committee. This committee will be made up of members of the HPC advisory committee (all researchers), as well as the manager of this Team. The exact scoring system will be developed in collaboration with the adjudication committee, but proposals will be rated on scientific/research merit, feasibility, potential impact, resource availability, researcher engagement, and alignment with institutional priorities by at least 2 members of the panel, working independently. Following this process, the full committee will collate results and select projects. Care will be taken to ensure some distribution of support to research areas which have traditionally struggled to fund support for software development. We will also prioritize projects which seek to enhance existing software tools. The Team will initially support a single project for no more than 6 months, though projects will be eligible to reapply for support in subsequent calls for proposals. This format (borrowed from the SHARCNET Dedicated Programming competition which has been highly successful) will ensure that the Team can be accessed broadly and can effectively and equitably allocate their resources whilst at the same time allowing them sufficient time to make meaningful progress on projects.

The exact number of simultaneous projects will be determined based on capacity, but it is envisaged that the Team will support 4 – 6 projects at a time. The intent is that no individual Team member would spend 100% of their time on a single project to ensure that knowledge and expertise across the Team would be used to support all projects. Rather, we envisage individual Team members perhaps spending 40% of their time on a single project, and probably supporting 2 major projects at any one time, as well as perhaps doing some outreach / consultation work.
Appendix 9: Sample Rules for Selecting Projects (2)

Through strong and regular engagement with the Research organization, the Team will be proactive to understand potential software/tool needs of research projects in early stages to minimize delays and maximize resource planning and research effectiveness. For approved research projects, the Team will make use of the service request/management platform for internal tracking, recording, and reporting project activities similar to, but separate from, other units in Information Services. Researchers may request services through direct contact with the Team Lead, unit Manager, or customer support staff. The Team will create an online presence as well which should include an online platform for potential clients to self-identify and seek engagement. The Team will regularly liaise with staff in the University Library and Research Office to coordinate activities and align plans.

The AVP Information Services will meet regularly with the VP Research, University Librarian, and Research Office to review the alignment of the initiative with other University activities and highlight the work of the Team.

As awareness of the Team increases it can reasonably be expected that requested work will exceed the Team capacity. A mechanism will be developed in conjunction with institutional research support groups to vet and prioritize work. Common approaches within the institution for these processes involve the use of pre-determined frameworks for categorizing workload into broad request types based on expected effort required and anticipated outcomes (e.g. small effort vs large, fix vs new, mandatory vs optional), as well as the use of stakeholder groups to make or advise on resource contention issues. These techniques would likely be applied to assign workload to the Team. The availability or prospect of external funding may be a consideration in the prioritization and assignment of requests.
As the Team is contemplated to be quite modest initially, it is unlikely that any staff member would be dedicated to a single project without compensating factors coming into play (such as a fee for service arrangement allowing for additional staff resources). It could be possible for Team members to participate in projects across institutions, although that would be a complicated undertaking. A probable Team member workload would consist of one or two longer term projects assigned along with short incremental assignments which could be dispatched quickly to deal with immediate needs.

Longer term assignments would necessarily be informed by the available Team capacity and workload. Some augmentation from pre-existing staff along with one-time University funds are possible, although the Team is expected to be largely self-sufficient over the long term.
Appendix 10: Sample Rules for Selecting Projects (3)

There are currently a few ways that researchers can request help from the RCS team which would include the new team. Researchers can contact members of RCS directly by email or by phone after visiting the RCS webpage or being referred by another faculty member or staff. This is often a precursor to a face-to-face meeting to discuss the researcher’s requirements and discuss the type of work that would be needed. Researchers can also send an email or call the ITS Service Desk to request any ITS service. A ticket is then created which is sent to the RCS team.

Projects will be evaluated in terms of resources, time commitment and priority (low, medium or high based on deadlines or impact to researchers or campus) within the members of the team. The RCS team keeps a project table, updated weekly, which contains a description, start date, expected end date and currently allocated resources. As new projects are received and evaluated, they are entered in the project table and the start time, the length of the project is discussed with the researcher. A meeting with the researcher is suggested to discuss and understand the problem before making the evaluation.

Maximum project duration will be based on availability of staff and resources. RCS currently evaluates the project and if the project is expected to take more than 2 weeks of continual work, we try to break the task into smaller milestones instead. Some projects require minimal weekly help but can take several months from start to finish.

The number of concurrent projects will be based on the existing list of projects, the amount of resources that need to be allocated for each and the number of staff available to work on those projects. Some projects will require a consecutive number of hours while others will need time spread over several weeks. Projects will not be rejected if they fit within the eligible activities, but their start time could be
adjusted based on available resources. Currently with only one Research Computing Specialist in the RCS team, there are typically less than 10 active projects where 2-3 require daily work while others only need weekly attention or less. It could be possible for a team member to spend 100% of their time dedicated to a single project if there are no other projects waiting in the queue or if there is a time constraint on the project.
Appendix 11: Sample Researcher Survey

1. Which faculty/department are you associated with?
   - Centre Continuing Education
   - DeGroote School of Business
   - Divinity College
   - Faculty of Engineering
   - Faculty of Health Sciences
   - Faculty of Humanities
   - Faculty of Science
   - Faculty of Social Sciences
   - Other
   Other

2. To which of the following groups do you primarily belong?
   - Staff
   - Postdoc
   - Graduate Student
   - Undergraduate Student
   - Other
   Other

3. What is the extent of your involvement in research?
   - I am directly involved in performing research
   - I support those performing research
   - I am not involved in research
   - Other
   Other
4. How would you rank the following software categories with regards to its use in your research?

Please note that a given software package may fit in multiple categories. Examples include:

- Microsoft Excel and Mathworks Matlab/Simulink could be used for statistical analysis, simulations and simulations.
- Mediawiki and Microsoft Word could be used data organisation, collaboration and publication.

How you use the software is what is most important.

<table>
<thead>
<tr>
<th>Software Category</th>
<th>5 – Very Important</th>
<th>4 – Important</th>
<th>3 – Fairly Important</th>
<th>2 – Slightly Important</th>
<th>1 – Not Important</th>
<th>0 – Does not Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical or statistical analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computational models or simulations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text/data mining and content analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data visualisations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer aided design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data organization and administration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collaboration and data sharing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content rendering and publication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Is there software used in your research that does not fit into the categories presented in the previous question? If so, what is the software and what category would you place it in?


6. For the software used in your research, please check all that apply.

- Commercial
- Open Source
- Custom Made
- Other


7. For each of the following categories, if applicable, what are your most important software packages?

- Numerical or statistical analysis
- Computational models or simulations
- Text/Data Mining and content analysis
8. Have you ever included costs for software or software development in a funding proposal?
   - Yes
   - No

9. How is your current research software work funded?
   - <Institution>
   - CFI
   - NSERC
   - CIHR
   - SSHRC
   - Other
   Other

10. Do you use existing software that has been customised for your research? Examples of customised software could include extensions or modifications to commercial or open source software.
    - Yes
    - No

11. Do you write or customise software for your research or as part of your job?
    - Yes
    - No

12. What programming languages, development environments or tools do you use?
13. How do you typically test your software?
- No formal testing
- Developer tested
- Automated tests
- End user testing
- Other

Other

14. What source code management or version control software do you use for your software development?
- Open source version control software (Git, CVS, etc.)
- Proprietary version control software (Perforce, Visual Studio Team Services, etc.)
- None
- Other

Other

15. Have you released software/source code?
- No
- Only upon request
- On an internally hosted website
- On a publicly hosted service (Github, Sourceforge or the like)
- Commercially released and licensed
- Made available as a web application
- Other

Other

16. Do you use any kind of bug tracking software for your software development?
- No
- Feedback via email
- Web forums
17. Do you document your software?

- No
- Comments in the code for my own future understanding
- Extensive comments in the code to help others understand it
- Documentation outlining how the software functions and its usage
- A simple help/usage screen
- Extensive documentation regarding the code (API documentation, for example)
- Other

Other


Appendix 12: Sample Researcher Engagement Plan (1)

The Communications staff of Information Services along with members of the Communications and Marketing team of our External Relations department will develop a formal communications plan to introduce the Research Software Support function to the University research community. Consideration will be given to renaming or branding the existing Research Server Support group in support of the communications strategy.

Activities contemplated as part of the communications strategy:

- Introduction by the VPR as he/she attends various Faculty meetings throughout the University including the communication of value/benefits with this approach for both the local university research community and the broader Canadian/Global research community
- Amendments to the Information Services, University Library, and Research Office web sites
- General email communications to University employees, including faculty
- Targeted communications to specific faculty who may be identified as having research activities aligned with the goals of the proposed group
- Overviews to researchers on RSS contacts and processes to promote the existing suite of research software/tools as well as evolve functionality of this software to meet incremental research needs
- Regular reporting on activities through the University Information Technology Steering Committee (the senior IT governance committee of the University)
- Meeting with the Research Office staff to introduce the Research Software Support function and its relationship to research application support
- Ad hoc meetings as appropriate
Appendix 13: Sample Researcher Engagement Plan (2)

Researchers will be made aware of this initiative through an announcement on the research communications portal, as well as through faculty representatives who sit on the HPC advisory committee. A formal announcement of the program, including the application process for support will be circulated through the University Research Council which includes Associate Deans of Research from all Faculties at the University. Beyond this, the PA is in regular contact with the research community across campus and will be able to bring this opportunity to the attention of researchers as part of his regular interactions with the community.

The first project that the Team will support is a comprehensive survey of software currently in use by researchers on campus. In part, the purpose of this will be to help identify areas of commonality and themes that may guide support efforts of this Team over the duration of the program. Another outcome of this project will be a central listing that can be used to make researchers aware of software and expertise that exists on campus or is available through science.canarie.ca and Compute Canada. Creating such a list, which is currently lacking, has already been identified as a priority for research computing support by a comprehensive review of IT services that was undertaken last year.