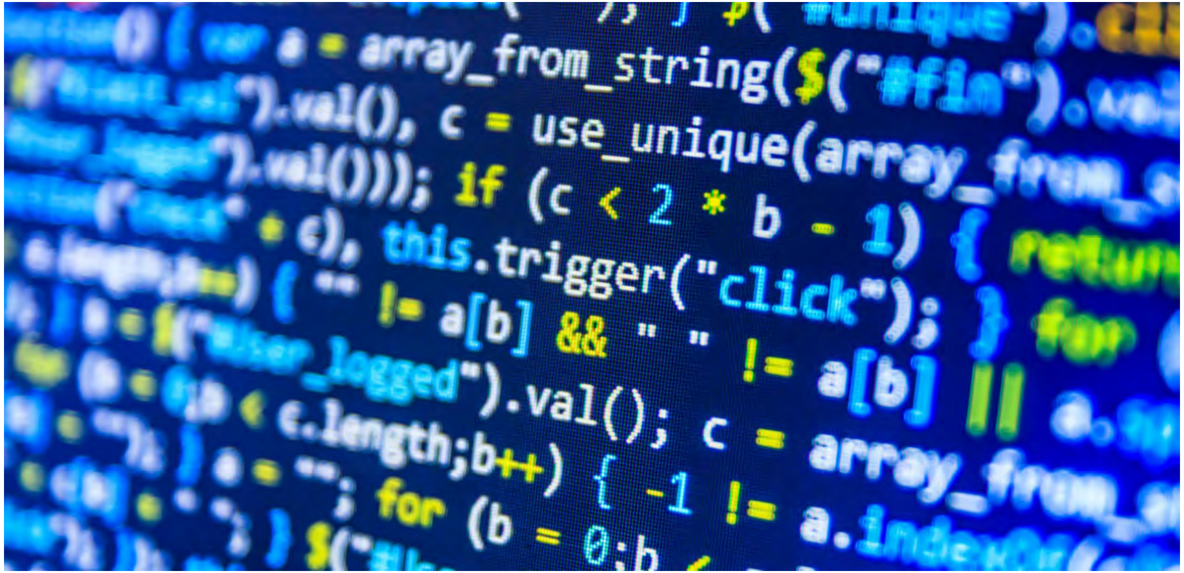


Implementing a research software team: early lessons learned

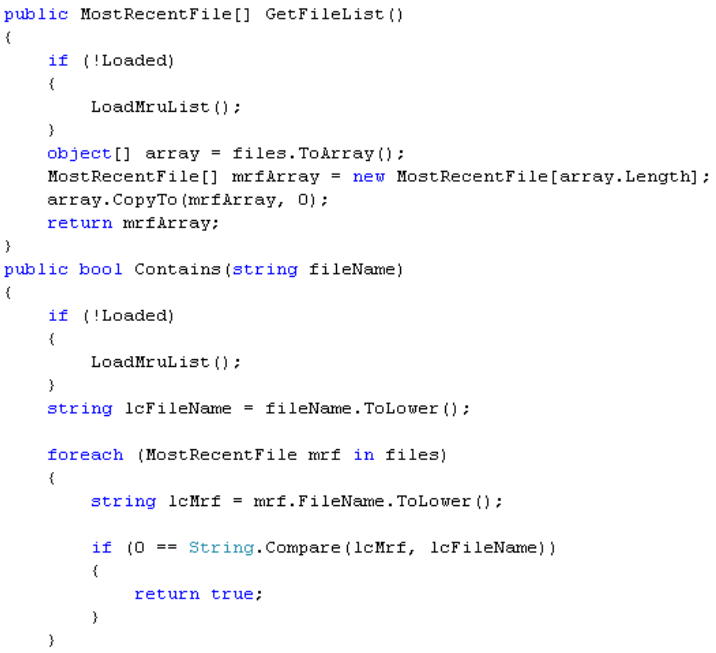
Ranil Sonnadara

About McMaster University

- Research intensive university
- Based in Hamilton with regional campuses in Burlington, Niagara, Kitchener-Waterloo
- Home to more than 70 research centres and institutes

Why do we need a research software team?

- Good software developers are expensive
- Good software developers are hard to find
- Research software is a bit different (generally)
- Most researchers are not software experts
- Software enables research but is not always the focus
- Most research cannot use unmodified “off the shelf” software



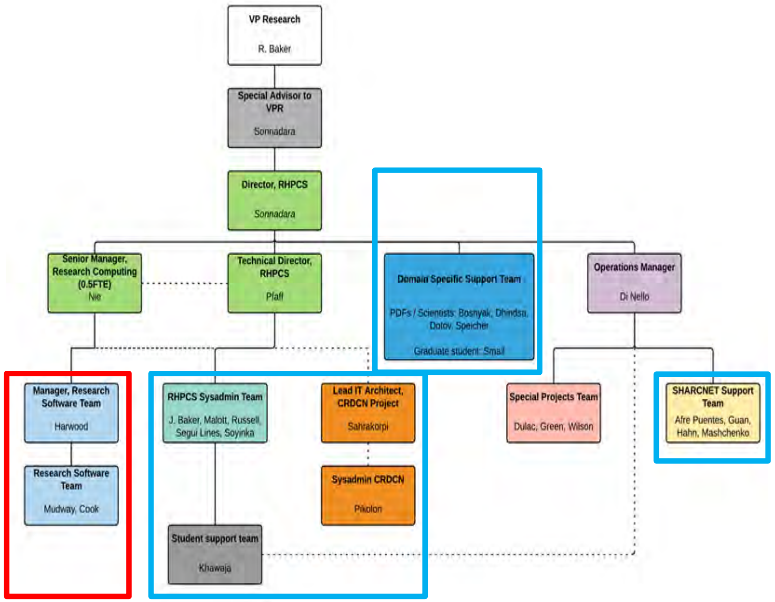
```
public MostRecentFile[] GetFileList()
{
    if (!Loaded)
    {
        LoadMruList();
    }
    object[] array = files.ToArray();
    MostRecentFile[] mrfArray = new MostRecentFile[array.Length];
    array.CopyTo(mrfArray, 0);
    return mrfArray;
}
public bool Contains(string fileName)
{
    if (!Loaded)
    {
        LoadMruList();
    }
    string lcFileName = fileName.ToLower();

    foreach (MostRecentFile mrf in files)
    {
        string lcMrf = mrf.FileName.ToLower();

        if (0 == String.Compare(lcMrf, lcFileName))
        {
            return true;
        }
    }
```

Why do we need a research software team?

- Graduate students are often a poor substitute
- Long term sustainability and flexibility is important
- Many labs are developing the same software tools

Research Computing at McMaster

- Dedicated central research computing team since 2001
- Direct reporting lines to VPR
- Strong ties to central IT, Faculty and Department computing support staff
- Strong integration with Compute Canada, SHARCNET



VP Research
R. Baker

Special Advisor to VPR
Sonnadara

Director, RHPCS
Sonnadara

Senior Manager, Research Computing (0.5FTE)
Nie

Technical Director, RHPCS
Pfaff

Domain Specific Support Team
PDFs / Scientists: Bosnyak, Dhindsa, Dotov, Speicher
Graduate student: Smail

Operations Manager
Di Nello

Manager, Research Software Team
Harwood

Research Software Team
Mudway, Cook

RHPCS Sysadmin Team
J. Baker, Malott, Russell, Segui Lines, Soyinka

Lead IT Architect, CRDCN Project
Sahrakorpi

Sysadmin CRDCN
Pikolon

Student support team
Khawaja

Special Projects Team
Dulac, Green, Wilson

SHARCNET Support Team
Afre Puentes, Guan, Hahn, Mashchenko

Canarie Research Software Team Pilot Project

- Filled an immediate need
- Fits into existing infrastructure
- Supplements existing skills
- Allows us to demonstrate the need for investment (we hope!)

Help researchers to develop their software tools

- Allow researchers to focus on research

- Allow graduate students to focus on developing the necessary range of skills

- Combine an understanding of research process, research computing support, software development

Share tools with the wider community

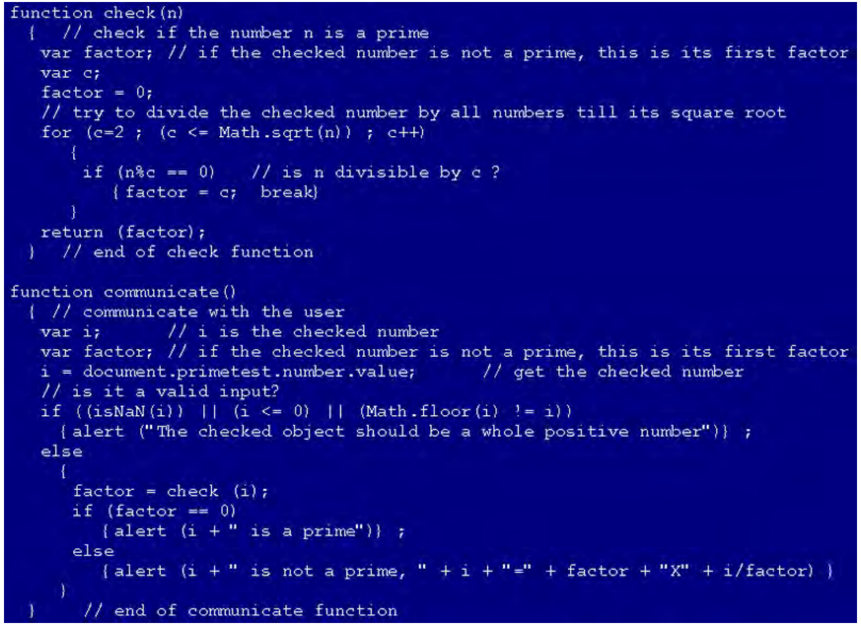
- Software developed for one research group may have value for other researchers
- There are many existing software tools of which nobody is aware
- Create tools that can easily be extended / adapted

Develop new HQP

- Developing research software skills is a national priority

- Create a training pathway into RS

- Enhance graduate student training through partnerships



```
function check(n)
  {  // check if the number n is a prime
   var factor; // if the checked number is not a prime, this is its first factor
   var c;
   factor = 0;
   // try to divide the checked number by all numbers till its square root
   for (c=2 ; (c <= Math.sqrt(n)) ; c++)
      {
       if (n%c == 0)    // is n divisible by c ?
          { factor = c;  break}
      }
   return (factor);
  }  // end of check function

function communicate()
  { // communicate with the user
   var i;       // i is the checked number
   var factor; // if the checked number is not a prime, this is its first factor
   i = document.primetest.number.value;      // get the checked number
   // is it a valid input?
   if ((isNaN(i)) || (i <= 0) || (Math.floor(i) != i))
     { alert ("The checked object should be a whole positive number")} ;
   else
     {
      factor = check (i);
      if (factor == 0)
         { alert (i + " is a prime")} ;
      else
         { alert (i + " is not a prime, " + i + "=" + factor + "X" + i/factor) }
     }
  }    // end of communicate function
```

Explore viability of RS team at McMaster

- Determine the need for research software services
- Demonstrate a positive return
- Enhance research computing support available

Improve software sustainability

- Research software is often written by research team members that move on (e.g. PDFs, graduate students)

- Poorly documented software that uses deprecated technologies needs to be rebuilt

Challenge #1: Staffing

- The ideal research software developer:
  - Understands research
  - Understands researchers
  - Can gather explicit and implicit requirements
  - Has experience of doing software developer
  - Can work as part of a team
  - Wants to support research
  - Has modest salary expectations

Challenge #1: Staffing

- Finding the right fit is hard
- For 3 positions:
  - 67 applicants
  - 12 interviews
- Many applicants were unsuited / unqualified

Challenge #2: Letting people know we are here

- Start slow to figure out what we're doing
- Build our own capacity
- Demonstrate competence / value
- Generate interest
- Fill the immediate need

Challenge #2: Letting people know we are here

- Started off by approaching individual research groups

- Once full team was in place, formal call for proposals

Challenge #3: Finding the right projects

- Developed an acceptance process that considered:
  - Fit with skills of team
  - Viable timelines
  - Realistic goals
  - Willingness to share
  - Research focused
  - Strong engagement from researchers

Challenge #3: Finding the right projects

- Used an existing scientific committee for research computing support as review panel
- Included members of the RS team who could provide technical review

Challenge #3: Finding the right projects

- Some projects were rejected because:
  - Licensing issues (commercial software)
  - Existing tools would be more suitable for their projects (e.g. database or survey applications)
  - Lack of clear research goals (e.g. clinical or education)

Challenge #3: Finding the right projects

- Rejected proposals were given full explanations of how projects did not meet criteria and were offered recommendations for moving forwards

Challenge #4: Researcher engagement

- Research teams need to have skin in the game

- Mechanisms adapted to suit the research group
  - Writing code
  - Involved with software development process
  - Requirements providing / verification / validation
  - Providing scientific expertise
  - Testing

Challenge #5: Managing expectations

YOU'RE FIRED!
GAAA!!!
www.dilbert.com scottadams@aol.com
NOT REALLY.
BUT NOW THIS 2% RAISE WON'T SEEM SO BAD.
4-8-08 © 2008 Scott Adams, Inc./Dist. by UFS, Inc.
THIS JOB IS ALL ABOUT MANAGING EXPECTATIONS.

- Helping researchers understand what is and is not feasible
  - Time constraints
  - Expertise
  - Available resources
  - Sometimes, things are difficult

McMaster's RS team

- Ron Harwood (RS Team Manager)
  - Software Dev/Sys Admin/Electronics Hacker/Maker (building a motion simulator for his part-time masters project - 17 years at McMaster)

- Thomas Mudway (Research Software Developer) *
  - Physics and Computational Mathematics (spent his masters cursing the stars while working on ChaNGa (Charm N-body GrAvity solver) cosmological simulation software)

- Oliver Cook (Research Software Developer) *
  - Computer Science (spent time in Rigolet, NL - interviewing locals and performing qualitative analysis as part of his masters)

- *Wayde Nie (Senior Manager)*
- *Angela Di Nello (Operations Manager)*
- *Dobri Dotov, Kiret Dhindsa (PDFs)*
- *Areeb Khawaja (Student staff)*

- **Oliver and Thomas are both recent graduates of masters programs*
- *Ron says he'll be done next month…*

RS steering committee

- Drawn from existing Research Computing Advisory Committee
  - 4 Faculty members (Physics, Electrical Engineering, Kinesiology, Economics)
  - Special Advisor to VPR (Surgery, PNB, SOTA)
  - Research Software Team Manager
  - Senior Manager, Research Computing

- Meet as needed (most communication via email)

Call for proposals

- Sent to McMaster’s research community through web, news announcements, and via ADRs
- Submissions collected via webform
- Distributed electronically to Steering Committee for review
- Two stage application: Paper and then in-person presentation

Proposal review

- Treated like any other grant review
- Top proposals were invited to do a brief in-person presentation and Q&A
- Steering committee drew in other expertise as required (e.g. PDFs, staff) and then made decisions by consensus

Metrics

- Impact
  - What is the expected breadth of usage (across disciplines, institutions)?
- Novelty
  - Does the software provide utility that is not available using existing (open source) tools?
- Engagement
  - Will the researchers be available to participate in development (programming, testing, providing knowledge as required)

Metrics

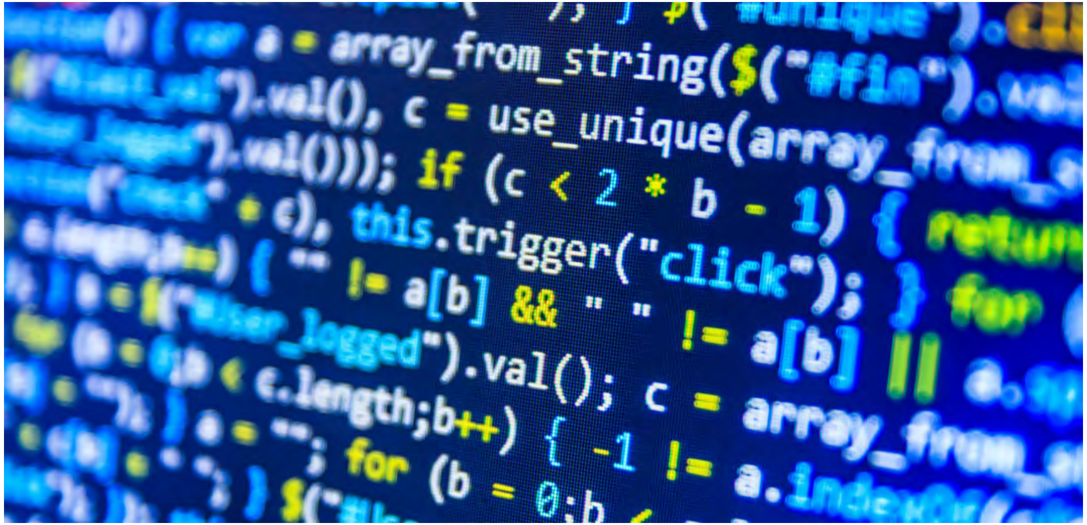
- Feasibility
  - Will the expected timeline of the project fit into a 3 – 6 month window of (0.5FTE) development time?

- Scope
  - Does the team have the necessary expertise
  - Is the complexity of the project reasonable

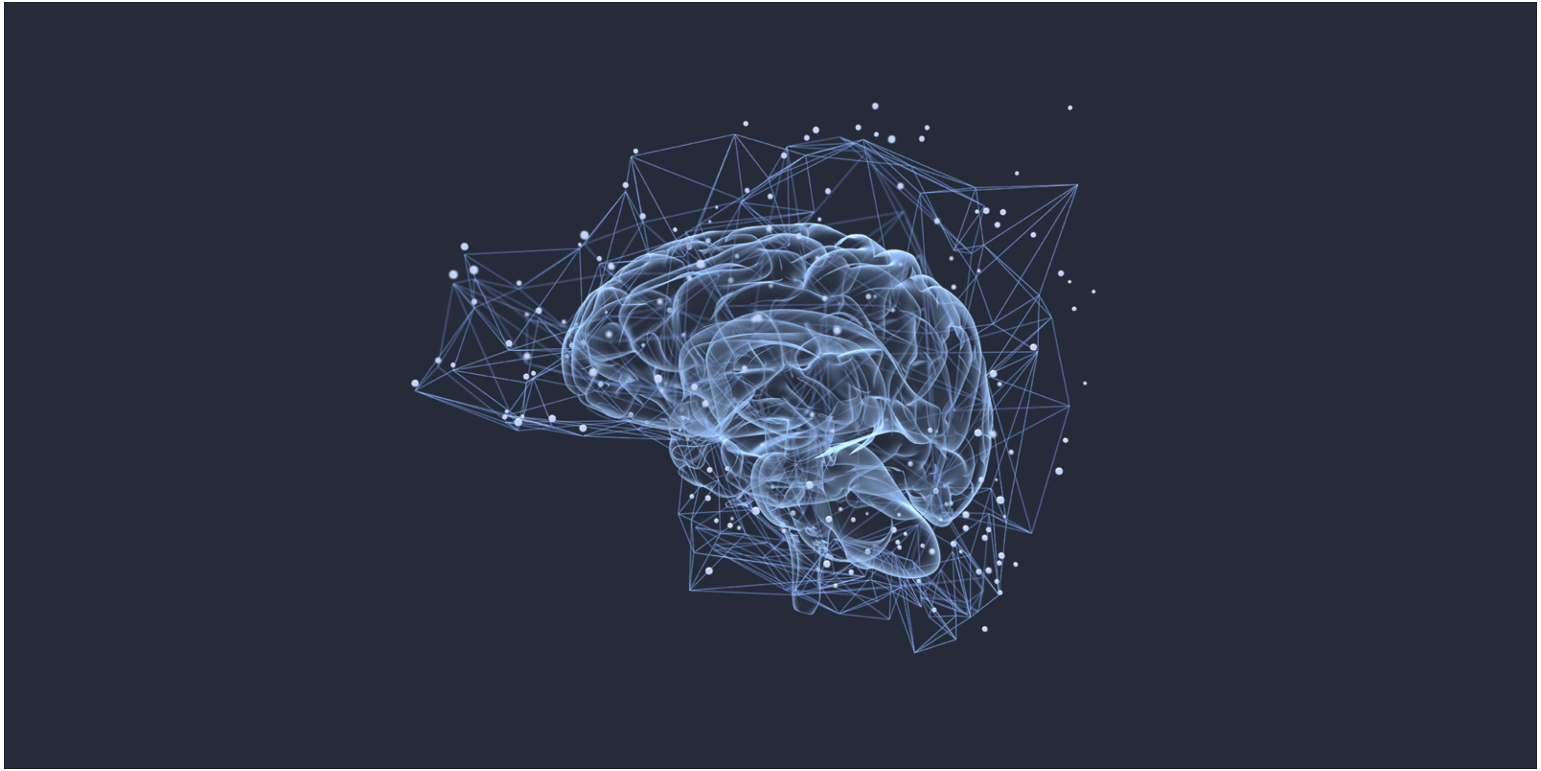
Outcomes from first CFP

- We had 10 formal responses and 21 other informal responses

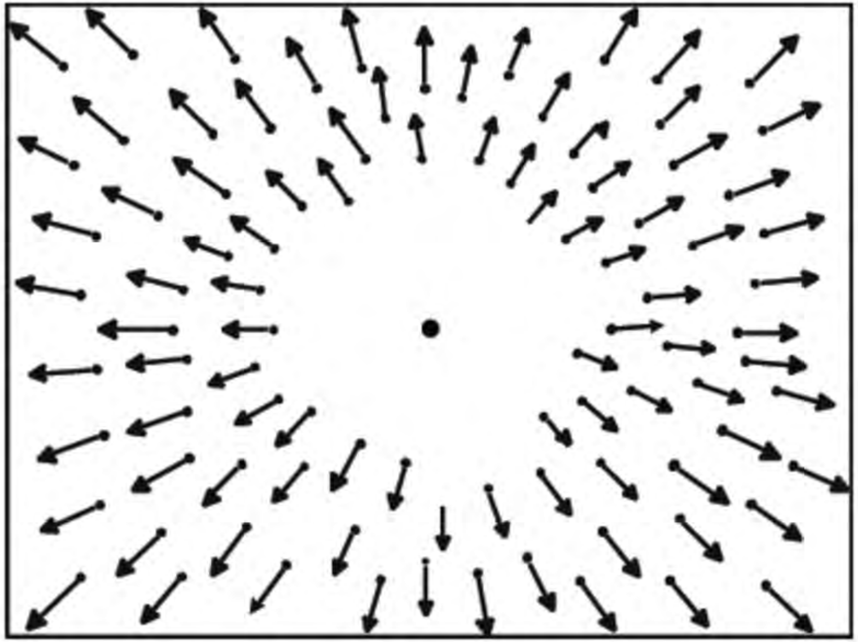
- We selected five projects



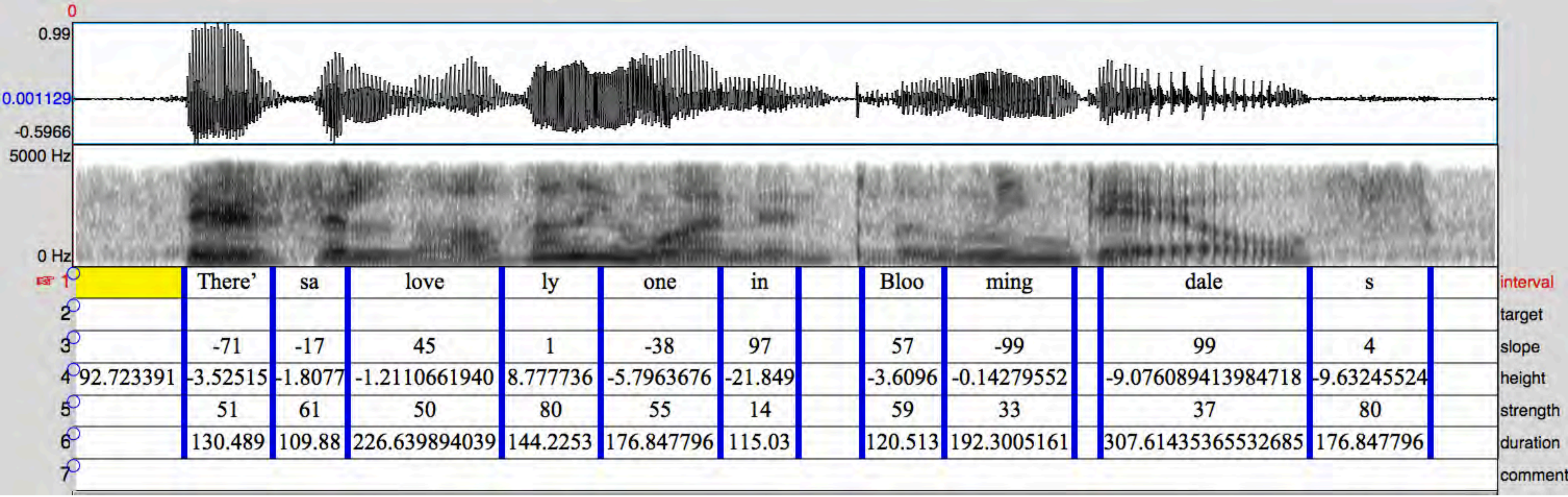
EEG Pipeline automation using MNE (1/5)



Motion tracking and optic flow toolbox (2/5)

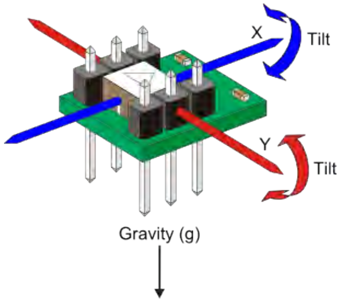


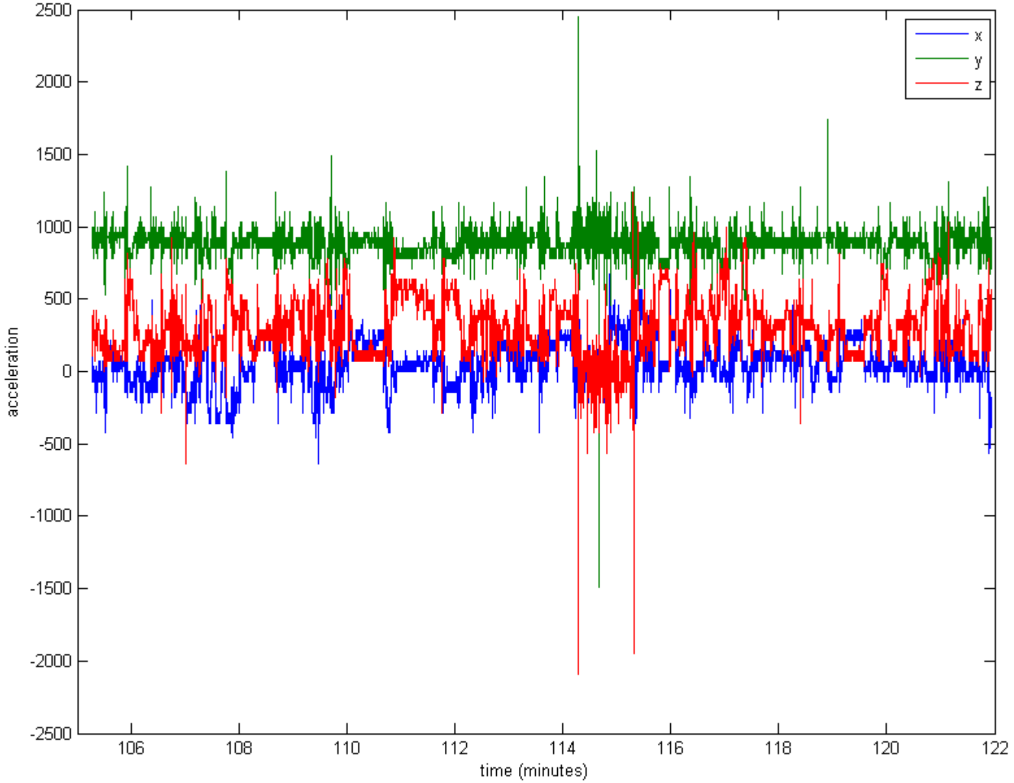
Voicelab software toolbox (3/5)



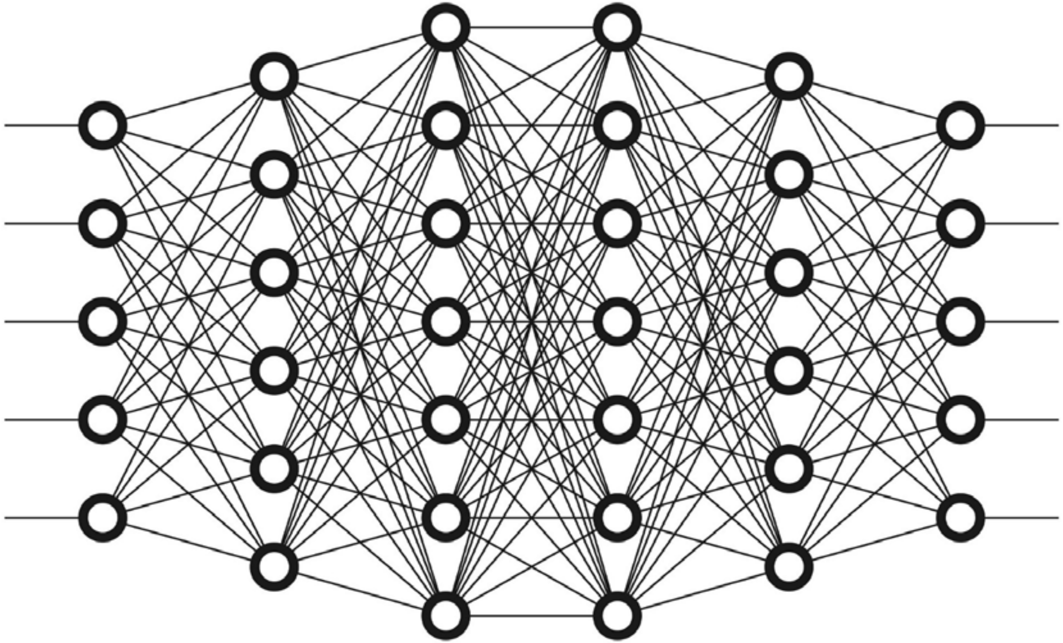
1		There’	sa	love	ly	one	in		Bloo	ming		dale	s		interval
2															target
3		-71	-17	45	1	-38	97		57	-99		99	4		slope
4	92.723391	-3.52515	-1.8077	-1.2110661940	8.777736	-5.7963676	-21.849		-3.6096	-0.14279552		-9.076089413984718	-9.63245524		height
5		51	61	50	80	55	14		59	33		37	80		strength
6		130.489	109.88	226.639894039	144.2253	176.847796	115.03		120.513	192.3005161		307.61435365532685	176.847796		duration
7															comment

Live sensor data toolbox (4/5)





Localized Feature Selection (LFS) Machine Learning Toolbox (5/5)



Step 1: Requirements gathering

- Informal meetings / Q&A with research teams to collect user stories
- User stories were given priorities (Must have / Should have / Nice to have)
- User stories were converted into tasks and were assigned scope levels based on shirt sizes

Step 1: Requirements gathering

- Small: relatively simple or quick (a few days)
- Medium: more complicated (more than a week)
- Large: hard to estimate full scope but may need to be split into multiple tasks
- Extra large: split into multiple tasks)

Step 1: Requirements gathering

- Conversations were documented, and then sent back to research team for approval to make sure the RS team understood the project

- Tasks were then assigned to RS team members

- Team were empowered to adjust assignments as appropriate

Leveraging common themes

- All projects used Python (some additional Matlab)
- All projects used a flow-based programming user interface
- All projects used a pipeline-based back end
- All projects used common Python libraries (e.g. NumPy, SciPy, Matplotlib, PyQtGraph)

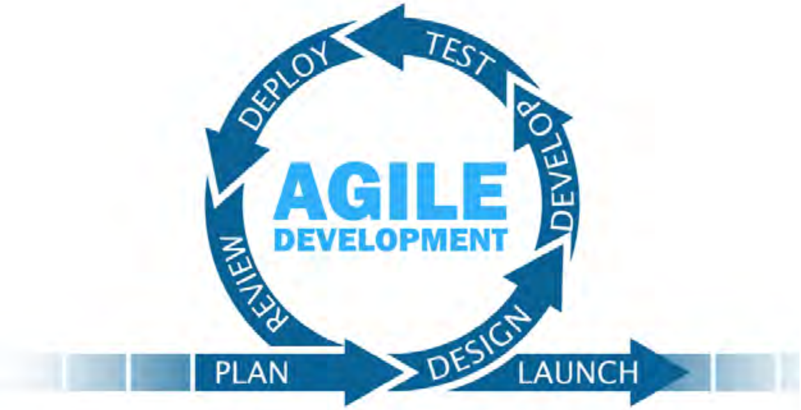
Leveraging existing infrastructure

GitLab

canarie

- Existing expertise (RHPCS, Sharcnet)
- Management / administrative support

Collaboration and Project Management



- Shared team office
- Regular meetings with research teams

The story thus far

- It is too early to say anything definitive, but thus far, all projects are progressing well….

Next steps: Understanding what RS is being used

- Being done as part of an institutional review on research computing support

- Research software survey
  - What types (commercial, open-source, custom)
  - How much money is being included in applications for funding
  - Who is doing research software development
  - What expertise is there already (languages, tools)
  - Where are people publishing

Next steps: Understanding campus needs

- We asked researchers what research computing support they needed the most
  - #1: Research software developers / Data visualisation
  - #2: Technical support (system administration)
  - #3: Storage for research data

Next steps: Understanding how to quantify impact

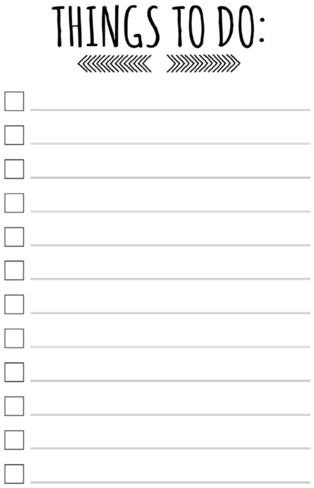
- Each software project will have a DOI to track citations
- Tracking downloads and bug / feature requests
- Publications / funding applications?
- What has the impact been on trainees within the group?
- Has participation in this project increased capacity in the RS team? On campus? More broadly?

Next steps: refining the selection process

- How well did projects fit with the team?
- Were the researchers appropriately engaged?
- Were the projected timelines accurate?
- Were there any challenges that we were not expecting?
- Was there any scope changes that were not expected?

Our “to do” list

- Finish up first round of projects
- Next call for proposals – when? any changes?
- Analysis of research software survey data
- Create a central listing of software / expertise on campus
- Link to external resources (CC, Canarie) where appropriate
- Extending funding



What has worked well

- Strong interest from research community
- Strong engagement from research groups
- Hired excellent staff
- RS team has integrated well into existing infrastructure
- Strong early signs of impact

Lessons learned (so far)

- Start up was slower than we would have hoped
- Finding the right people is hard
- Challenges come from unexpected places (office space)
- Providing additional operational support creates huge efficiencies and allowed us to hire strong software developers
- Picking projects with common themes has worked well
- We will need more time to really show impact

Acknowldgements

canarie

RESEARCH
@MCMASTER UNIVERSITY