CW15 Discussion session 1 outcomes

What are the management challenges of interdisciplinary research? How do we structure goals and rewards to involve the whole project?

Name of Topic Owner: Farah Ahmed

Name of Topic Reporter: Janneke van der Zwaan

What are the five most important things learnt during this discussion:

1. Having incidental collaborations is not going to work

2. We need central hubs that connect areas and have common software and common vocabulary

3. We need to identify hubs that are there, and sustain them

4. We need to design incentives for better collaborations

5. People don't understand the possibilities of eScience, we need to help them to expand their knowlegde

What are the problems, and are there solutions?

- There are not enough incentives for (real) interdisciplinary work
- Who takes the credit for interdisciplinary work?
- There are no incentives to use the tools that are out there
- Finding the people to collaborate with -> we need a hub to connect people/groups (solutions)
- Reviewing interdisciplinary papers (you need interdisciplinary reviewers or collaborating reviewers)

What further work could be done, and who should do it (make a pledge)?

- Tweet about what you do (with the higher goal of connecting people) also aimed at people outside your field
- We need a central body to help with this (might be the SSI)
- Organize workshops/tutorials/conferences

Are there any useful resources that people should know about?

- Elements system at the National History Museum (system to show your CV)

Should we consider Software Engineering as a discipline within interdisciplinary research?

Name of Topic Owner: Raquel Alegre

Name of Topic Reporter: Raquel Alegre

What are the five most important things learnt during this discussion:

1. Is software engineering a discipline?

2. Value of software engineering is not recognised properly

3. There is a lack of career path for software engineering

4. To strengthen the importance of software engineering we should consider as a discipline with equal weight

What are the problems, and are there solutions?

It is not recognised as an equal footing. There needs to be a career path and recognition for software engineers.

What further work could be done, and who should do it (make a pledge)?

Funding bodies and publishers should encourage research output based on software engineering.

Are there any useful resources that people should know about?

N/A

Can we crowd source materials for teaching Software Engineering?

Name of Topic Owner: Robert Haines

Name of Topic Reporter: Philip Fowler

What are the five most important things learnt during this discussion:

1. It is a nightmare using other people's materials to teach; you always change them. Accept that when you teach you will change any materials.

2. Key concept is the "atom" of teaching: can't share slides but can share (short) videocasts or code/markdown (through a GitHub)

3. Another concept is having courses labelled with dependencies so you can create pathways through a web of material. (apt-get install courseX)

4. Software Carpentry is an exemplar (but it isn't well suited to software engineering). Also CodeAcademy and CodeSchool (but commercial).

5. A Material Creation Manifesto: "I will make my course freely available on GitHub; I will write in Markdown; I will use other people's material before I write new material."

What are the problems, and are there solutions?

Well below the tipping point - easy once there is traction.

Universities may resist due to IP issues.

What further work could be done, and who should do it (make a pledge)?

Rob pledges to do this for his MSc course on Software Engineering at Manchester. He also has 3 months funding for a Software Engineer to work on this.

Are there any useful resources that people should know about?

What software do you use to help you collaborate, and how?

*** Chair

Robert Patton

*** Name
Mark Basham

*** 5 questions
- Think/decide first and be consistent
  Understanding the project needs
- Huge variety of software, what do we need? we need a linked guide (a resource).
- Be aware of other's limitations so everyone can participate equally
- Avoid monolithic solutions to solve the whole collaboration
-

*** Problems and solutions
Problem: large range of software
Solution: index/archive them ordered by properties

*** What further work could be done, and who should do it?
SSI should do it and keep it updated
(wiki or flossmanuals?)
with properties of where have been used and whether has been successful (curateion process)

*** Are there any resources that people should know about?
Lots! that SSI will compile.

What software do you use to collaborate and why?

Name of Topic Owner: Robyn Grant

Name of Topic Reporter: Russell Garwood

What are the five most important things learnt during this discussion:

1. When setting up a collaboration one of the most important things is to decide up front on your program or choice of tools for collaboration.

2. Interdisciplinary work can be hard - how you collaborate is not always going to be the best way to do it, and you have to be open to other ideas.

3. Data versioning is something which we are currently very weak on - a git-like system for data versioning would be very valuable in many fields.

4. Economic circumstances are something to consider when collaborating - using open source software will make collaboration across economic circumstances easier.

5. Skype doesn't replace talking to each other - collaboration in person does still have a place and importance in this era.

Collaborative tools discussed:

Wikis are a good tool for communicatng.

Google docs

-- Most of us use google docs for collaborative writing, but we sometimes get resistance from some researchers who are wary about the system

-- But due to data laws this is not allowed for civil servants, and can be problematic

Skype / google hangouts

-- These are useful for meetings

Dropbox / googledrive

-- we all use this for sharing files, data and documents

-- Bitbucket is a git repository which gives you unlimited size files for sharing

Owncloud

-- replacement for dropbox - open source

-- can be put on own server

Piratepad/etherpad

-- Good for meetings writing stuff etc.

Git

-- Git is viewed as a nice tool

-- But has technical overhead

-- Has tools for chatting (e.g. gitter)

-- Some privacy issues

-- for pipeline and code

-- non software people in group have not really used it

-- git has become more usable over the years

-- A gui would be a nice frontend for this

-- guis don't allow you to do stuff under the hood - it simplifies

Basecamp

-- project-based social network for managing big projects

http://dat-data.com/

-- git for data

-- versioning -once it is working, start building hub

-- fetch specific dataset from given time to allow for reproducibility

-- ability to keep data private? Currently not

-- authentifcation required for pushes.

-- Anyone can access it currently.

-- Currently all open

-- doesn't have to host data - registry of data

-- dat clone - binary data - metadata

-- api interfacing tool - for underlying data...

-- policy recommendations

Genbank

-- versioning of genomes/some data

-- but not metadata

disapora - social network which is distributed

collaborative work with wider community

-- wikipedia is a great place to collaborate

-- maps - people tag georeference them - always happens on wikipedia. Something about the community there

-- interesting because it's a means of collaboration emerging from a community

-- different way of collaborating..

scholarpedia

-- allows you to edit stuff as you go along

-- criedt for the page,which is reviewed and then approved

-- editors reviewers and

commentpress

-- wordpress add on - public version for writing books with versioning in

-- versioned commented book written in public - wrapped up by official publisher

How do we manage legacy code? AND What software engineering tools help separate implementation from application?

Name of Topic Owner: Alys Brett

Name of Topic Reporter: Oliver

What are the five most important things learnt during this discussion:

1. Try not to write more legacy code

2. Break legacy code down into modules, conceptual layers, input-process-output. If you can't add functional testing where you can? Do you have an exit strategy for the code? What parts of the code are actually used?

3. Senior stakeholders can understand the problems at a high level but the low level implementation is not appreciated

4. Can we sell scientific benefits (performance) of refactoring as a worthy use of money and time?

5. Clean it up when you touch it (boy scout rule),

What are the problems, and are there solutions?

The majority of developers write code without thinking that it will be used in 10 years time.

Short term contracts are a problem.

Domain specific code.

Benefits are expected during grant periods

Longer life times for software than expected

Performance and new features are often a priority not software engineering.

Software management strategies

You need an exit strategy

Understand what code is actually being used.

Code coverage.

Don't rewrite more legacy code. Standard data sets and libraries.

What further work could be done, and who should do it (make a pledge)?

Education

Are there any useful resources that people should know about?

M. Feathers - Working with Legacy Code

Fowler - Refactoring

Software Engineering

IDE tools

What are the skills required to automate a workflow which uses different components?

Name of Topic Owner: Alexander Konovolov

Name of Topic Reporter: Susheel Varma

What are the five most important things learnt during this discussion:

1. Importance of Workflow Design
2. Need for Best Practices
3. Identification of Weakest Link in a workflow pipeline
4. Move up the Research Paper to Software to Data value chain by automation
5.

What are the problems, and are there solutions?

- Formats
- Provenance and Audit Trails
- Appropriateness of reproducibility
- Compounded Archival factor

What further work could be done, and who should do it (make a pledge)?

- Workflow Community for SSI
-
    - Steve Crouch, Ian Emsley,- Best Practices in Workflow
    - Susheel Varma, Alan - Workflow Categorisation Advice
    - Best Practices for Workflows
-
    - Scientific Pipeline
    - Specification of expected input and output formats
    - Platforms typically used
    -
        - HPC, Windows,
    - Visual Programming
    - Top tips and cool tools

  - Continous Integration for RSE in
  - Provenance and Audit Trail
  -
    - Paper - Data trails
    - Software version control (
    - Testing output from software
    - uncertainty analysis
    - Compounded Archival factor for reproducibility
    - Weakest link - identification and best practice
  - Is there a benefit of engineering good workflows
  - Journals
  - Make you change your mindset
  - Problems
  -
    - Real experiments is not as rigorous as computational experiments
    - Proliferations of possibly flawed comp exper
  - Greg Wilson - Reproducibility is sharing with a future version of yourself

Are there any useful resources that people should know about?

- Software Carpentry
- Workflow Engine documentations
- WorkflowPatterns.com
- SpiffWorkflow
- Common Workflow Language - Google Group
- WFDesc standard
- W3C Prov standard

Communication in interdisciplinary research - problems and solutions

Name of Topic Reporter: Tom Pollard

## What are the problems?

### 1. papers and authorship
- some disciplines, solo authorship is required.
- too much emphasis on first authorship.
- lots of variation on what constitutes authorship

### 2. understanding
- language barriers
- we use different toolsets (e.g. git is magic to most clinicians)

### 3. research structure
- support. if you end up in a lab outside your field pet scientist. clued up lab head, then get pet bioinformatician. forget that the person is a researcher. basically get used. career gets ruined.
- trust. often interdisciplinary fields are often novel. less trust networks. will the other group

commit as much effort as you?
- funding bodies need to facilitate collaboration. make it poitive.

### 4. creating team units
- management skills lacking [can lead to wasteful practice]
- openness in terms of collaborative mindset
- need to be more personality selection [too much focus on papers]

## What are the solutions?

### 1. authorship:
- agreeing ground rules on authorship (e.g. nature paper)
- develop long term strategy for publishing. 'publishing rota!'.
- publish under team names? randomised authorship lists? contribution percentages.

### 2. understanding:
- "each of us is a member of public for another discipline"
- establish the common ground first. make links.
- interdisciplinary researchers have to be able to ask stupid questions!
- face-to-face meetings
- software carpentry
- alongside papers, provide accessible outputs (links to blogs, gists?)

### 3. research structure
- "you can't choose your family, but you can choose your friends". interdisciplinary research is an opportunity to choose people!

4. ### creating team units
- training
- bring established project managers into teams

What would a good funding scheme for interdisciplinary research which uses software look like?

Name of Topic Owner: Peter Burlinson (BBSRC) & Miriam Dowle (EPSRC)

Name of Topic Reporter: (in writing: Joe Parker & Yannick Wurm; verbally: Peter Burlinson).

What are the five most important things learnt during this discussion:

Need a RCUK Hackday!!

1. How new funding schemes are worked up

- sandpit / ideas lab (sprint type session). Problems are accessibility, group composition
- haven’t yet done hackathons.

2. What broad types of RCUK funding are available

- small grants e.g. TRDF can be difficult to assess, need wide panel. also open to fewer researchers than amounts suggest

3. When cross-funding between councils, there must be one lead concil. Call them up if unsure (they'll include reviewers from elsewhere)

4. How to support RSEs for e.g. project maintenance: Pathways To Impact can be used to request/justify resources to support this

- PIs and RCs should have dialogue with their parent HEIs, perhaps core-funded RSEs to support this? recognition not applicable to all projects, REF incentive?
- Software are very ref returnable (easy to quantify, have industry usrs etc…) → Universities should get RSEs.
- Lot of variation in examples or software outputs as impact case studies in REF submissions. Bioscience not too bad.

5. Neil: SSI have some data on amount of funding used for RS activities. No data yet on user base arising from this, but plans to sample

- Peter: BBSRC have some info on users, including data from e.g. Tom Freeman (Edinburgh) using stuff like download/update stats to monitor user counts.
- ? page impressions on downloads / project sites?
- ResearchFish, interface for impact/return data

Neil: "If you were to be judged on just 1 or 2 criteria (for trickle-funding of successful research software), what would they be?"

- Yannick: mentions. Citations are used but this has problems for software. but you can monitor other types of activity, mentions in the full-text, or on discussion/help fora etc
- Laurent: 'does the software follow best-practice?' -create a set of minimal criteria "to get in"
  - It should be possible to formalize what are the criteria to determine whether or not something is "good software" that should be supported.
- how do you identify the software that should get funding? (metrics)
  - github counts? "mentions" in social media. citations (or mentions in articles)?

Laurent: you are more likely to do truly interdisc. res. when you are an early career researcher. so incentivise this by e.g. saying 'only young ppl. and how about >2 , from different labs'? perhaps 1yr projects, kickstart

- old profs too busy & too much technical debt to explore new technologies approaches
- Miriam: need to make sure both/all disciplines get something from collabs
- Joe: academy structure type thing? residential / week type course/ conference, proposals at the end
- Peter: that's fine but at some point it becomes competitive
- Laurent: we're fine with competition but we want it to be fair

What are the problems, and are there solutions?

What further work could be done, and who should do it (make a pledge)?

Are there any useful resources that people should know about?

other ideas we came up with included:

- trickle-funding for ongoing/successful projects, tied to e.g. user base
- promoting multidiscp. research by small project (12months) scheme for early career researchers, where applications had to be from >2 ECRs in different disciplines

Present:

Peter Burlinson (BBSRC), Miriam Dowle (EPSRC), Yannick Wurm (QMUL), Joe Parker (RBG Kew), Neil Chue Hong (SSI), Rita Hendricusdottir (ELIXIR), Laurent Gatto (Cambridge)