Case Studies of Research Software Development & Management in Universities

Daniel S. Katz
d.katz@ieee.org, http://danielskatz.org, @danielskatz

NITRD MAGIC, 2 September 2020

Credit to Rob Haines for the original slides for our paper (10.1109/SE4Science.2019.00009 & https://arxiv.org/abs/1903.00732)
Research Software and Research Software Engineers
Why do we care about research software?
Why do we care about research software?

• Funding
  – ~20% of NSF projects over 11 years topically discuss software in their abstracts ($10b)
  – 2 of 3 main ECP areas are research software (~$4b)
Why do we care about research software?

• Funding
  – ~20% of NSF projects over 11 years topically discuss software in their abstracts ($10b)
  – 2 of 3 main ECP areas are research software (~$4b)

• Publications
  – Software intensive projects are a majority of current publications
  – Most-cited papers are methods and software
Why do we care about research software?

• Funding
  – ~20% of NSF projects over 11 years topically discuss software in their abstracts ($10b)
  – 2 of 3 main ECP areas are research software (~$4b)

• Publications
  – Software intensive projects are a majority of current publications
  – Most-cited papers are methods and software

• Researchers
  – >90% of US/UK researchers use research software
  – ~65% would not be able to do their research without it
  – ~50% develop software as part of their research
Software as a strategic advantage
Software as a strategic advantage

• Both direct (software itself) and indirect (things enabled by software and by professional software developers)
Software as a strategic advantage

• Both direct (software itself) and indirect (things enabled by software and by professional software developers)
• Competition between institutions
  – E.g. for faculty who want to go where they can be successful and are supported by resources including start up funding, facilities (computing, data, software, etc.) and staff
  – E.g. for projects that build or rely on software
Software as a strategic advantage

• Both direct (software itself) and indirect (things enabled by software and by professional software developers)
• Competition between institutions
  – E.g. for faculty who want to go where they can be successful and are supported by resources including start up funding, facilities (computing, data, software, etc.) and staff
  – E.g. for projects that build or rely on software
• UK Research Excellence Framework (REF)
  – System for assessing quality of research in UK higher education institutions, tied to university funding, includes software as an output and research that relies on software
How do we develop better research software?
How do we develop better research software?

There are two hard problems in Software Engineering:
How do we develop better research software?

There are two hard problems in Software Engineering:

1. People
How do we develop better research software?

There are two hard problems in Software Engineering:

1. People
2. Convincing others that “people” is a hard problem

https://martinfowler.com/bliki/TwoHardThings.html;
https://twitter.com/holman/status/77629183336979456
How do we develop better research software?

There are two hard problems in Software Engineering:

1. People
2. Convincing others that “people” is a hard problem

So, let's talk about people...

https://martinfowler.com/bliki/TwoHardThings.html;
https://twitter.com/holman/status/776291833336979456
How do we develop better research software?

There are two hard problems in Software Engineering:

1. People
2. Convincing others that “people” is a hard problem

So, lets talk about people...

Really:

https://martinfowler.com/bliki/TwoHardThings.html;
https://twitter.com/holman/status/77629183336979456
How do we develop better research software?

There are two hard problems in Software Engineering:

1. People
2. Convincing others that “people” is a hard problem

So, let's talk about people...

Really:

• Culture problems (hardest)

https://martinfowler.com/bliki/TwoHardThings.html;
https://twitter.com/holman/status/77629183336979456
How do we develop better research software?

There are two hard problems in Software Engineering:

1. People
2. Convincing others that “people” is a hard problem

So, let’s talk about people…

Really:

• Culture problems (hardest)
• Management problems

https://martinfowler.com/bliki/TwoHardThings.html;
https://twitter.com/holman/status/776291833336979456
How do we develop better research software?

There are two hard problems in Software Engineering:

1. People
2. Convincing others that “people” is a hard problem

So, let's talk about people...

Really:

- Culture problems (hardest)
- Management problems
- Software problems (easiest)

https://martinfowler.com/bliki/TwoHardThings.html
https://twitter.com/holman/status/77629183336979456
The Craftsperson and the Scholar
The Craftsperson and the Scholar

• Scholar: archetypical researcher driven to understand things to their fullest capability
  – Find intellectually-demanding problems
  – Curiosity-driven, work on a topic until understanding has been acquired, pass on that understanding through teaching
The Craftsperson and the Scholar

- Scholar: archetypical researcher driven to understand things to their fullest capability
  - Find intellectually-demanding problems
  - Curiosity-driven, work on a topic until understanding has been acquired, pass on that understanding through teaching

- Craftsperson: driven to create and leave behind an artifact that reifies their efforts in a field
  - Feels pain when things they make are fragile or ugly
  - Prefer to make things that explain themselves
  - Work requires patience, and pride in doing a job well

http://www.software.ac.uk/blog/2012-11-09-craftsperson-and-scholar
The Craftsperson and the Scholar

• Scholar: archetypical researcher driven to understand things to their fullest capability
  – Find intellectually-demanding problems
  – Curiosity-driven, work on a topic until understanding has been acquired, pass on that understanding through teaching

• Craftsperson: driven to create and leave behind an artifact that reifies their efforts in a field
  – Feels pain when things they make are fragile or ugly
  – Prefer to make things that explain themselves
  – Work requires patience, and pride in doing a job well

• Scientific software requires individuals who combine the best of both roles.

http://www.software.ac.uk/blog/2012-11-09-craftsperson-and-scholar
What is a Research Software Engineer?
What is a Research Software Engineer?

Research

Software Engineering

Software Engineer

Researcher
What is a Research Software Engineer?
What is a Research Software Engineer?
What is a Research Software Engineer?

Society of Research Software Engineering

An independent organization for the RSE movement
- Membership
- Voting rights
- International

UK Registered Charity Number 1182455
- Same model as Royal Society of Chemistry and Institute of Physics, etc.

• [https://www.society-rse.org/](https://www.society-rse.org/)
US RSE Association

- Created with inspiration and support from Society of Research Software Engineering
- Focused on US members and US issues
  - Networking, jobs, careers, events, ...
  - In academia, labs, industry, ...
- Started in 2018, publicized in 2019
- BOF at PEARC 2019, panels & BOF at SC19, workshops at PEARC 2020 & SC20
- About 500 members
- https://us-rse.org
Career paths in US universities

• Campus Research Computing Consortium (CaRCC)’s CI Workforce Development/Professionalization Committee held 2018 CI Professionalization Workshop

• One outcome: draft Research Computing and Data Professionals Job Elements and Career Guide

• Organized around “four facings” roles: researcher-facing, system-facing, software/data-facing, sponsor/stakeholder-facing

• For RSEs and data scientists (software/data facing), similar to the NCSA descriptions

• Now publicizing, aim to get universities to support and implement
Research Software Group Models
Illinois NCSA: Software Directorate

- **Organizational context**
  - Research institute hosted at a university, outside of academic departments

- **Team size**
  - ~40

- **Remit**
  - Support individual needs; generalize those needs across projects; build software frameworks in response

- **Funding model**
  - Research grants (soft funding)

- **Job security; career progression**
  - Staff hired on indefinite contracts
  - Five grades of research programmer (RP): assistant, RP, senior, lead, principal
Notre Dame: Center for Research Computing

- **Organizational context**
  - Part of the broader Notre Dame Research organization at the university

- **Team size**
  - 24

- **Remit**
  - Provide software development support and services to researchers

- **Funding model**
  - Grants, contracts, and collaborations (soft funding)

- **Job security; career progression**
  - Staff hired on fixed term contracts
  - Three grades of Research Programmer, roughly equivalent to first three NCSA grades
Manchester: Research Software and Data Science

- Organizational context
  - IT Services, external to the Faculties
- Team size
  - 25
- Remit
  - Application support; training; short projects (weeks); research projects (months-years)
- Funding model
  - Application support and training: baseline funded (hard funding)
  - Projects: cost recovery from grants (soft funding), but underwritten by IT Services
- Job security; career progression
  - Staff hired on permanent contracts
  - Three grades of RSE: ~ graduate, postdoc, lecturer
Agility

• Notre Dame is the most “agile”
  – Group divided into 4 teams, each with research programmers, a product owner, and a scrum master (shared across each two teams)
  – A team works on 1-6 projects, generally using 2-week sprints

• Illinois uses some agile methods for overall group
  – Staff work on multiple projects and in multiple teams, so twice weekly team standups help keep everyone focused on tasks and share knowledge
  – Projects can adapt some agile methods while recognizing staff are shared

• Manchester uses some agile processes within faculty-focused teams
  – Staff work on multiple projects in multiple teams; informal sprint model with regular feedback from PIs
  – Coordination between the faculty teams is via a weekly leaders' meeting
Line management & mentoring

- **Manchester**
  - Head & 6 area leads line-manage & mentor staff
  - 0.5 FTE institutional support for each

- **Illinois**
  - 2 co-leads and 6 group leaders manage and mentor staff
  - 0.2 FTE institutional support for each
  - Each project also has a senior developer, supported ~0.1 FTE on that project to mentor staff in that project

- **Notre Dame**
  - Line management provided by dual Scrum Master/Manager role within teams (~8:1 dev to manager ratio, 2:1 team to manager), then above, Assoc. Director of Cyberinfrastructure (head of software group)
  - Mentoring through technical leadership team composed of senior developers, with representation from each scrum team
Supporting Research Software Development
High-level comparison

Institutional memory spanning projects, domains, time
Flexible workforce with flexible skills
Can support varying levels of effort, in particular portions of staff members
Supports mentoring/coaching
Reduced bus factor with regards to project core knowledge
Enables scalable growth to more rapidly take on new/large efforts
Fosters reuse and sustainability of built software
Costlier staff, however, better more maintainable code
Perpetual precarious staffing allocations when solely reliant on grants

<table>
<thead>
<tr>
<th>Effect</th>
<th>Manchester</th>
<th>Illinois</th>
<th>Notre Dame</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>+</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>+</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>+</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>±</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>−</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>−</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>−</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>−</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Overcoming varying finite duration funding streams
Overcoming varying finite duration funding streams

• We want to retain professional software development staff
Overcoming varying finite duration funding streams

• We want to retain professional software development staff
• Grant funding
  – Often less than 1 RSE FTE
    • And at NCSA and Manchester, a project’s 1 RSE FTE should be split across 2 staff members to lower project risk and reduce project capture of staff
  – Shorter duration than staff careers
Overcoming varying finite duration funding streams

• We want to retain professional software development staff
• Grant funding
  – Often less than 1 RSE FTE
    • And at NCSA and Manchester, a project’s 1 RSE FTE should be split across 2 staff members to lower project risk and reduce project capture of staff
    – Shorter duration than staff careers
• RSE groups tend towards a form of “Matrix Management”
  – Principle Investigator axis: changes over time
  – RSE group manager axis: fixed
Overcoming varying finite duration funding streams

- We want to retain professional software development staff
- Grant funding
  - Often less than 1 RSE FTE
    - And at NCSA and Manchester, a project’s 1 RSE FTE should be split across 2 staff members to lower project risk and reduce project capture of staff
    - Shorter duration than staff careers
- RSE groups tend towards a form of “Matrix Management”
  - Principle Investigator axis: changes over time
    - Funded by research projects
  - RSE group manager axis: fixed
Overcoming varying finite duration funding streams

• We want to retain professional software development staff
• Grant funding
  – Often less than 1 RSE FTE
    • And at NCSA and Manchester, a project’s 1 RSE FTE should be split across 2 staff members to lower project risk and reduce project capture of staff
  – Shorter duration than staff careers
• RSE groups tend towards a form of “Matrix Management”
  – Principle Investigator axis: changes over time
    • Funded by research projects
  – RSE group manager axis: fixed
    • Funded by ... ? Overhead? Tax on projects? Institution?
Overcoming varying finite duration funding streams

• We want to retain professional software development staff
• Grant funding
  – Often less than 1 RSE FTE
    • And at NCSA and Manchester, a project’s 1 RSE FTE should be split across 2 staff members to lower project risk and reduce project capture of staff
  – Shorter duration than staff careers
• RSE groups tend towards a form of “Matrix Management”
  – Principle Investigator axis: changes over time
    • Funded by research projects
  – RSE group manager axis: fixed
    • Funded by ... ? Overhead? Tax on projects? Institution?
• How do we fund RSE groups to set aside time for growth?
Overcoming varying finite duration funding streams

• We want to retain professional software development staff
• Grant funding
  – Often less than 1 RSE FTE
    • And at NCSA and Manchester, a project’s 1 RSE FTE should be split across 2 staff members to lower project risk and reduce project capture of staff
    – Shorter duration than staff careers
• RSE groups tend towards a form of “Matrix Management”
  – Principle Investigator axis: changes over time
    • Funded by research projects
  – RSE group manager axis: fixed
    • Funded by ...? Overhead? Tax on projects? Institution?
• How do we fund RSE groups to set aside time for growth?
  – E.g., skill development, technology exploration, grant writing
### Career paths (NCSA)

<table>
<thead>
<tr>
<th>Title</th>
<th>Assistant Research Programmer</th>
<th>Research Programmer</th>
<th>Senior Research Programmer</th>
<th>Lead Research Programmer</th>
<th>Principal Research Programmer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major Function</td>
<td>Develop software and tools to address scientific and other real world problems</td>
<td>Lead development activities and oversee/mentor groups of developers as a &quot;player coach&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key Responsibilities</td>
<td>Develop novel software or contribute to existing software, both independently and/or in collaboration with team members, in support of a project's goals</td>
<td>Interact with people in a wide range of educational, scientific, and engineering disciplines to create advanced software</td>
<td>Represent the group at meetings, give presentations at conferences or other venues, and contribute to publications and grant proposals</td>
<td>Carry development and contribute significantly towards the reporting on one or more projects</td>
<td>Track technologies changes/research activity in relevant fields</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Evaluate the strengths and weaknesses between varieties of novel approaches to problems and communicate these to colleagues</td>
<td>Design new approaches and techniques in resolving project specific problems using independent judgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lead and assume responsibility for activities of a project's development team/sub-team, ensure team/project goals and deadlines are met</td>
<td>Oversee/steward design and development of several significant software code bases</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision Making Authority</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervision</td>
<td>Senior/Lead/Principal Research Programmer, Group/Division Leads, Project Leads</td>
<td>Group/Division Leads, Project Leads</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required Education, Training, and Experience</td>
<td>BA/BS in computer science. Alternative degree fields will be considered/accepted if accompanied by equivalent experience</td>
<td>Some course level soft. dev. experience 1-3 years software development experience 5+ years software development experience 10+ years software development experience</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preferred Experience</td>
<td>Subject matter experts on one or more technologies</td>
<td>Ability to establish a software development effort from the ground up (create software from scratch)</td>
<td>Proficient in 3 or more programming languages with an ability to explain/decide why one would be utilized over another</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ability to follow research publications, an ability to write, and an ability to be creative towards open ended software development</td>
<td>Programming in one or more programming languages (e.g. Java, C++, Python, Scala, FORTRAN, Ruby, Javascript)</td>
<td>Proficient in 3 or more programming languages with an ability to explain/decide why one would be utilized over another</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web development (e.g. Server side scripting, client side frameworks, HTMLS, CSS, REST)</td>
<td>Web development (e.g. Server side scripting, client side frameworks, HTMLS, CSS, REST)</td>
<td>Web development (e.g. Server side scripting, client side frameworks, HTMLS, CSS, REST)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Databases (e.g. MySQL, MongoDB, PostgreSQL)</td>
<td>Databases (e.g. MySQL, MongoDB, PostgreSQL)</td>
<td>Databases (e.g. MySQL, MongoDB, PostgreSQL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contributions towards research publications</td>
<td>Contributions towards research publications</td>
<td>Contributions towards research publications</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Experience in one or more of the following: Machine learning or data mining, Natural language processing, Geospatial data management and programming, Computer vision or graphics, HPC environments, cloud computing, and/or systems administration</td>
<td>Experience in one or more of the following: Machine learning or data mining, Natural language processing, Geospatial data management and programming, Computer vision or graphics, HPC environments, cloud computing, and/or systems administration</td>
<td>Experience in one or more of the following: Machine learning or data mining, Natural language processing, Geospatial data management and programming, Computer vision or graphics, HPC environments, cloud computing, and/or systems administration</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Best practices
Best practices

• Research is a team endeavor; larger & more complex projects need a wider range of skills
Best practices

• Research is a team endeavor; larger & more complex projects need a wider range of skills
• Professionals who write software & analyze data are essential
  – Can’t push these activities to students & post-docs, but need to work with them
Best practices

• Research is a team endeavor; larger & more complex projects need a wider range of skills
• Professionals who write software & analyze data are essential
  – Can’t push these activities to students & post-docs, but need to work with them
• RSE group members (senior developers and/or project managers) can bring more mature project management aspect/structure
  – Team organization (roles, responsibilities)
  – Work organization (WBS & timelines for deliverables)
  – Software organization (code repositories, issue trackers, wikis, messaging)
  – Coding practices (sprints, code reviews, TDD, integration testing, autobuilds)
  – → better, more sustainable code & more efficient group coordination
Best practices

• Research is a team endeavor; larger & more complex projects need a wider range of skills
• Professionals who write software & analyze data are essential
  – Can’t push these activities to students & post-docs, but need to work with them
• RSE group members (senior developers and/or project managers) can bring more mature project management aspect/structure
  – Team organization (roles, responsibilities)
  – Work organization (WBS & timelines for deliverables)
  – Software organization (code repositories, issue trackers, wikis, messaging)
  – Coding practices (sprints, code reviews, TDD, integration testing, autobuilds)
  – → better, more sustainable code & more efficient group coordination
• These can be barriers in projects that focus on science questions
Institutional memory
Institutional memory

• Research software is becoming more important, complex, and costly
Institutional memory

• Research software is becoming more important, complex, and costly
• Expertise is valuable
Institutional memory

• Research software is becoming more important, complex, and costly
• Expertise is valuable
• Students and postdocs pack up and move on
Institutional memory

- Research software is becoming more important, complex, and costly
- Expertise is valuable
- Students and postdocs pack up and move on
- RSE groups with longevity beyond any individual project, can act as “institutional memory”
  - Long-term/permanent contracts
  - Contracts not aligned to projects
Institutional memory

- Research software is becoming more important, complex, and costly
- Expertise is valuable
- Students and postdocs pack up and move on
- RSE groups with longevity beyond any individual project, can act as “institutional memory”
  - Long-term/permanent contracts
  - Contracts not aligned to projects
- RSEs are generally more mobile across domains than other research staff
  - Opportunities for translation of knowledge/artifacts across more users/communities
Changing scientific culture
Changing scientific culture

• Scientific research is about scientific discovery first and foremost
Changing scientific culture

• Scientific research is about scientific discovery first and foremost
• There’s a long and deep culture as to what this means
  – Scientific method, students, paper publications, ...
Changing scientific culture

• Scientific research is about scientific discovery first and foremost
• There’s a long and deep culture as to what this means
  – Scientific method, students, paper publications, ...
• We’re trying to change that culture
  – Or at least adapt it to include technical aspects required by modern day science
Changing scientific culture

• Scientific research is about scientific discovery first and foremost
• There’s a long and deep culture as to what this means
  – Scientific method, students, paper publications, …
• We’re trying to change that culture
  – Or at least adapt it to include technical aspects required by modern day science

• Research is increasingly a team endeavor
  – As projects become larger and more complex, a wider range of skills is required

Jay, et al; Identifying the challenges of code/theory translation; 2017; 10.3897/rio.3.e13236
Changing scientific culture

• Scientific research is about scientific discovery first and foremost
• There’s a long and deep culture as to what this means
  – Scientific method, students, paper publications, …
• We’re trying to change that culture
  – Or at least adapt it to include technical aspects required by modern day science

• Research is increasingly a team endeavor
  – As projects become larger and more complex, a wider range of skills is required
• Whether we write software or papers, we are all researchers

Jay, et al; Identifying the challenges of code/theory translation; 2017; 10.3897/rio.3.e13236
Summary
Conclusions
Conclusions

• Software is becoming recognized as an *essential* part of research
Conclusions

• Software is becoming recognized as an *essential* part of research
• Support aspects of such software are not ... yet
  – RSE staff, RSE groups
  – Models for sustaining, citing and crediting software
Conclusions

• Software is becoming recognized as an *essential* part of research
• Support aspects of such software are not ... yet
  – RSE staff, RSE groups
  – Models for sustaining, citing and crediting software
• RSE-type groups are emerging globally to address these needs
Conclusions

• Software is becoming recognized as an essential part of research
• Support aspects of such software are not ... yet
  – RSE staff, RSE groups
  – Models for sustaining, citing and crediting software
• RSE-type groups are emerging globally to address these needs
• RSEs and data scientists do not work in a vacuum
  – They are key to common research activities, such as hypothesis generation, study design, data analysis, and interpretation of results
Conclusions

• Software is becoming recognized as an *essential* part of research
• Support aspects of such software are not ... yet
  – RSE staff, RSE groups
  – Models for sustaining, citing and crediting software
• RSE-type groups are emerging globally to address these needs
• RSES and data scientists do not work in a vacuum
  – They are key to common research activities, such as hypothesis generation, study design, data analysis, and interpretation of results
• Efforts to make this well understood and accepted by the scientific community at large are ongoing
Acknowledgments
Acknowledgments

Paper: 10.1109/SE4Science.2019.00009
Acknowledgments

Paper: 10.1109/SE4Science.2019.00009
Acknowledgments

Paper: 10.1109/SE4Science.2019.00009
Thanks:
   – Rob Haines, Kenton McHenry, Caleb Reinking
Acknowledgments

Paper: 10.1109/SE4Science.2019.00009

Thanks:
– Rob Haines, Kenton McHenry, Caleb Reinking
"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development (NITRD) Program

**Mailing Address:** NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

**Physical Address:** 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: [https://www.nitrd.gov](https://www.nitrd.gov)