



Metrics for RSE Contributions: An overview of community consultations

Nooriyah Lohani, Manodeep Sinha, Justin Baker, Rebecca Lange, Heidi Perrett, Nick May, Janet Stacey, Rowland Mosbergen

RSE-AUNZ Steering Committee - sc@rse-aunz.org

Software Engineering



Research

Software Engineering



Software Engineer



Research

Software Engineering

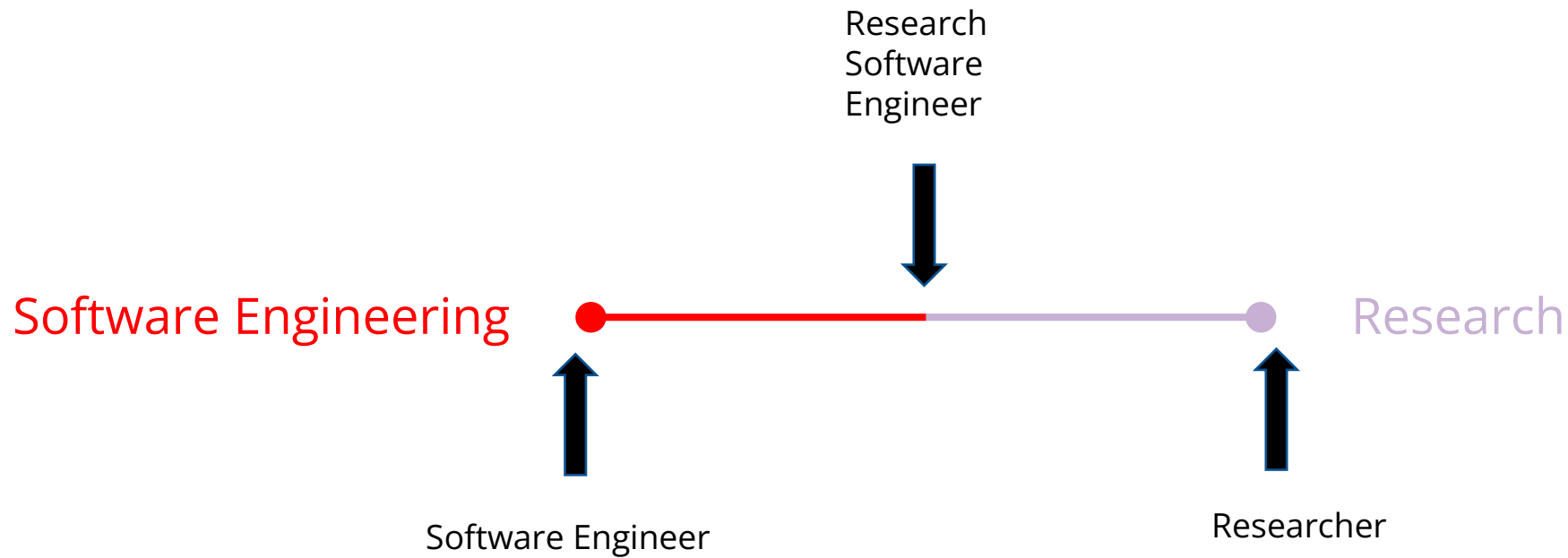


Software Engineer



Researcher

Research



The RSE Association of Australia and New Zealand

This community aims to build awareness of the diverse Research Software Engineer (RSE) roles, to connect volunteers in the RSE domain, and to help build practical solutions for the RSE community.

This community is for:

RESEARCHERS and ACADEMICS

who code

GENERALISTS

who brings communities together across the
research and technical domains

SOFTWARE ENGINEERS

who work in the research domain

SYSTEMS ADMINISTRATORS

who maintains research systems

Join the community

2021 RSE-AUNZ Steering Committee



Manodeep Sinha



Justin Baker



Heidi Perett



Nick May



Nooriyah Lohani



Rebecca Lange

2022 RSE-AUNZ Steering Committee



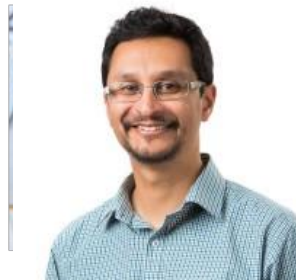
Manodeep Sinha



Justin Baker



Janet Stacy



Rowland Mosbergen

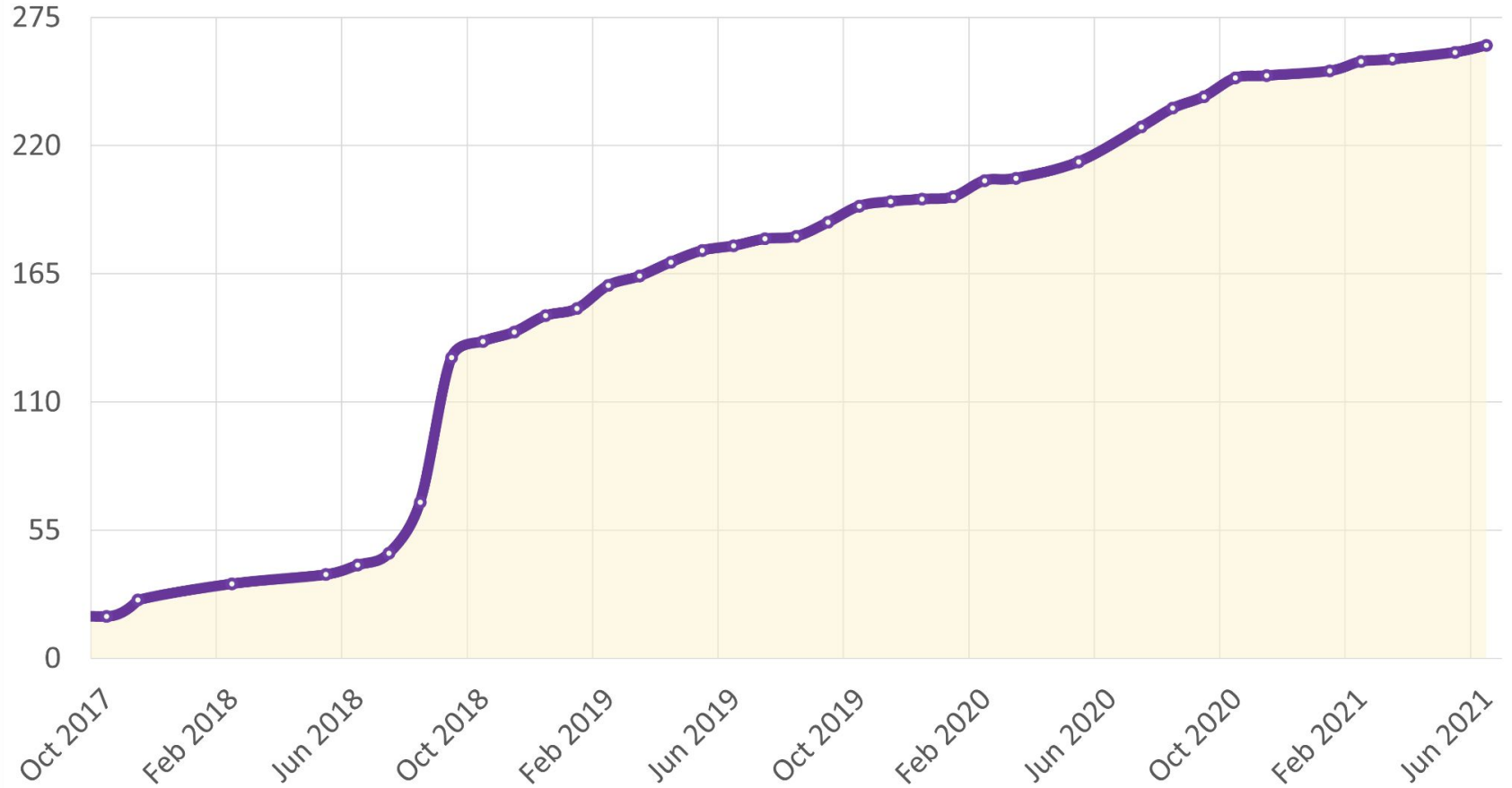


Nooriyah Lohani



Rebecca Lange

RSE-AUNZ Membership



RSE-AUNZ Goals - Australia & NZ

- Maintain an active, diverse, sustainable RSE Community.
- Define the role of RSEs within the eResearch, Academic, and Professional context.
- Form a connection between communities: people and organisations.
- Obtain recognition for RSE-AUNZ and its role.

RSEs **lack** career progression

Recognition for RSE Contributions

RSE Context

- What kinds of qualifications/experience do RSEs have?
- Where do RSE's work?
- Who is - and is not - an RSE?
- What do RSE's do?
- How are RSE's recognised for their contributions?

RSEs **lack** career progression

RSEs require metrics for contributions

Facilitating community discussion on metrics (2021)

- BoF at C3DIS
- BoF at eResearch Australasia
- BoF at NZRSE conference

Metrics: Qualitative & Quantitative



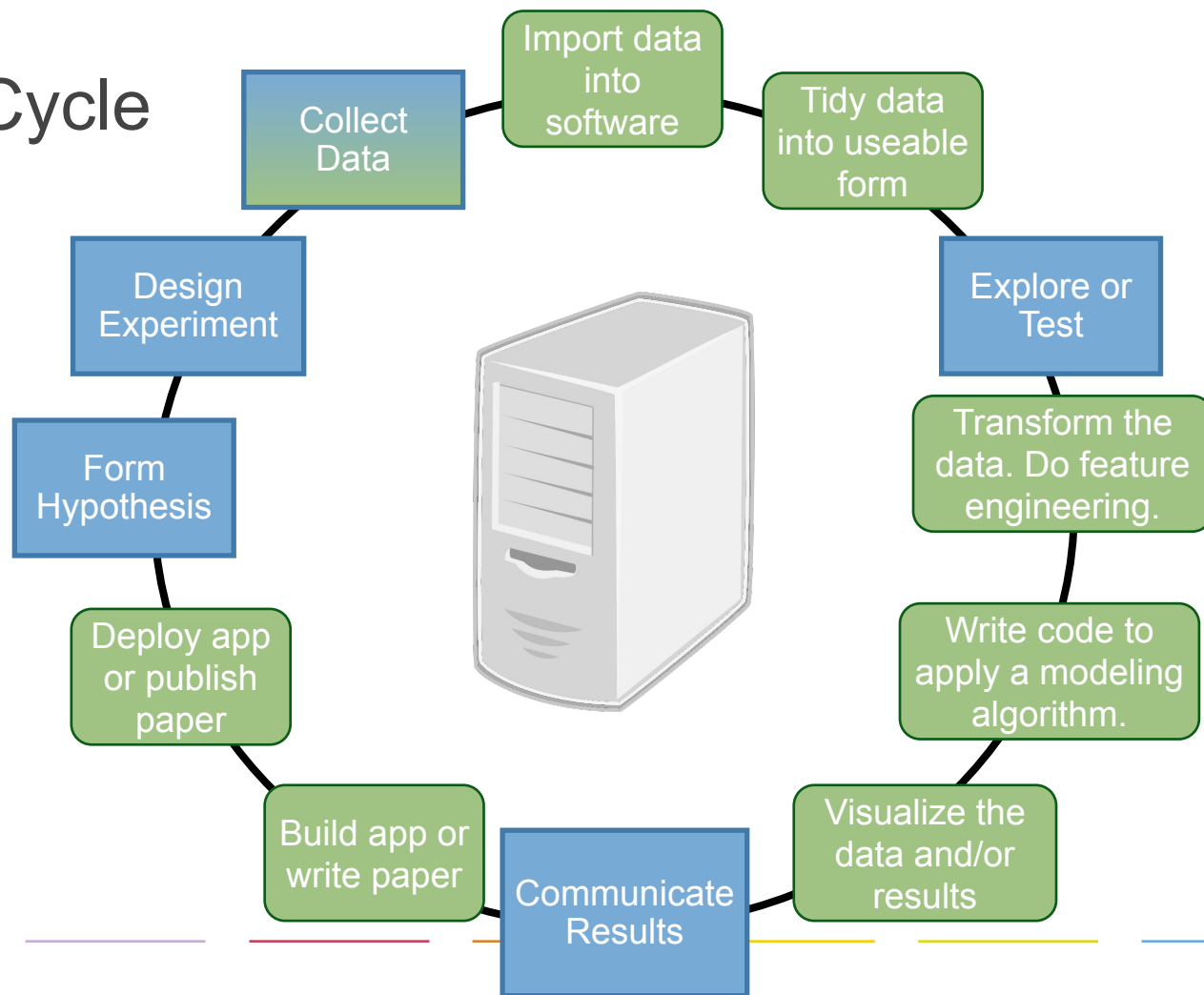
“Now we faced another giant challenge: distilling all those **quadrillions of bytes** of data down to kilobytes of actual information that would go into an image we could show the world.”

Katherine L. Bouman, IEEE Spectrum

“Now we faced another giant challenge: distilling all those **quadrillions of bytes** of data down to **kilobytes of actual information** that would go into an image we could show the world.”

Katherine L. Bouman, IEEE Spectrum

Research Cycle



Some considerations

- Qualitative story or quantitative metric?
- Is the metric measuring more of the research side or the software engineering side?
- How does the metric compare between different domains?
- How does the metric evolve with time?
- How does the metric relate to underlying structural inequalities (e.g., men are cited more frequently than women)?

Areas where RSE's contribute

- Core software development
- Software quality
- Tools & Techniques
- Reproducibility
- Training and Outreach
- Communication
- Innovation and Discovery

NeSI and NIWA already have KPIs for measuring RSE contributions [\[1\]](#)

Core software development

- Creating new software
- Improving software efficiency (e.g., runtime, memory/disk/power usage)
 - Porting software to different hardware (e.g, gpus, FPGAs)
- Knowledge of existing software including domain-specific ones
 - Time required to perform a task by an RSE vs a pure researcher/software engineer

Software Engg./Quality Metrics?

- Would usual software engineering metrics work - may be not
 - CSIRO has adopted a qualitative metric (shout out to Eric Bastholm)
- Adding testing or continuous integration improves reliability,
- Usability can be improved by adding helpful error messages within code, and extensive documentation.
- Maintainability is a bit more fuzzy - but existing metrics like cyclomatic complexity could be used here, but can also extend to the dependencies and the effort needed to maintain a working build as the dependencies get updated.
- Advanced options could be UX - number of users/new/repeat users might be an indicator

Core software development

- Creating new software
- Designing new algorithms
- Improving software efficiency (e.g., runtime, memory/disk/power usage)
- Porting software to different hardware (e.g, gpus, FPGAs)
- Knowledge of existing software including domain-specific ones
- Version control
- Documentation
- Testing code (e.g., continuous integration)
- Ensure software engineering practices

Tools and Techniques

- Extending knowledge to multiple disciplines, APIs (say image-processing) might allow to use for a wide range of research topics.
 - E.g., Galaxy (origin in bioinformatics) -> used in linguistics for NLP
- RSEs working with a researcher to make the research software scalable / reusable
- Versioning, continuous integration, containerisation and managing (dependencies) libraries
- Metrics: View counts, download/forks, number of users, number of repeat users, test coverage, badges

Tools and Techniques: Metrics

- Metrics: View counts, download/forks, number of users, number of repeat users, test coverage, badges
- Would usual software engineering metrics work - may be not
 - CSIRO has adopted a qualitative metric (shout out to Eric Bastholm)
- Usability/maintainability is a bit more fuzzy - but existing metrics like cyclomatic complexity could be used here, but can also extend to the dependencies
- Advanced options could be UX - number of users/new/repeat users might be an indicator

Training and Outreach

- Training the next generation of researchers
- Running conferences/events
- Time spent preparing new training materials (domain specific/ targeted training), improving existing material and delivering training at scale
- Metrics from the training community
 - Number of people trained
 - Impact stories

Reproducibility

- RSEs could verify the tool does what the researcher claims
 - e.g., peer review, code integrity
- Test-code validates if code behaviour is reproducible
- Direct contribution to science (research side) and software behaviour (s/w engg. side)
- Reimplementing code in another language, or moving to another platform
 - Containerization, virtual environments

Communication

- Acting as a glue between software eng. and researchers
- Could include software release as well as software papers (JOSS, JORS) + contribution to regular academic articles
- Talks at conferences
- Visualisation
 - Creating dashboards to present the results
- How do we measure impact?

Innovation and Discovery

The novel application of advanced software/hardware technology to facilitate new insights into the research process

- Define novel - new technology solution or new use to the application
- What is the impact? - faster, automation, more robust, insight
- How is impact measured? What is the baseline that is being improved on?
- Are there graduations in measuring how advanced the technology that is being used is?

Next steps

- Community consultations and contributions to identify potential metrics
 - What would be the best ways to highlight RSE contributions to senior management/executive?
 - What kind of metrics have worked for your team?
- What other ways could you demonstrate RSE value/impact?

Let's stay
connected

Website: <https://rse-aunz.org>

Contact the Steering Committee - sc@rse-aunz.org

Join the RSE-AUNZ mailing list - <https://bit.ly/join-rse-aunz>

Metrics - <https://bit.ly/rse-aunz-metrics>

Impact stories - <https://bit.ly/rse-aunz-impact-stories>