

F-Role of Software Engineering Research

Lead: Anshu Dubey

Presenter: Ethan Coon

Writer: Ethan Coon

Participants -- Also please add your email address

Anshu Dubey adubey@anl.gov

Devin Matthews

Tom Clune

David Tarboton

Dan Ibanez

Cindy Rubio-Gonzalez

Colin C. Venters

Ethan Coon

Walid Keyrouz

Ernie Lucier

Michael Mascagni

Tien N. Nguyen

Hridesh Rajan

Sol Greenspan

Charles Ferenbaugh cferenba@lanl.gov

Matthew Dwyer

Vikram Adve

Karl Rupp

Grady Campbell

John McGregor

Mike Glass

How can SW engineering knowledge from the broader software community be leveraged in CSE and customized for it?

Broader SE vs SE for CSE

Is there really a difference between broader SE and SE for CSE? Can we quantify this difference?

- Design studies to understand this? E.g. mining of repositories?
- This can help understand the difference between broad SE and SE for CSE, where projects face problems
- This can help understand what adaptations for CSE are working.

Fundamental principles of SE apply to CSE, but methods of applying them may differ or need to be tailored to specifics of CSE.

- For example there may be lateral interactions between code units, or modularity may come at the cost of performance, there may be many other trade-offs

How can SW engineering knowledge from the broader software community be leveraged in CSE and customized for it?

The community needs convincing of using tools and best practices. Ways to fix this:

- tools have to be easy, very apparent that the user will gain something (time?, credibility, acceptance, reproducibility, reduction of technical debt)
- convince people they are incurring technical debt by not using best practices
- ***If you don't use these tools/best practices, should all of your results be suspect?***

Need to be aware of small vs large teams.

Lifecycle plans and code maturity -- when moving prototype (research) code to production code -- SE has much to offer here.

What are the areas in SE that need to develop within CSE because they are underrepresented within SE?

- performance portability -- performance engineering
- interdisciplinary interactions -- separation of concerns in software design -- how to utilize abstractions in maintaining separation of concerns
- rapidly changing requirements -- research code -- rapidity and magnitude of changes differ with maturity (though maybe the startup world has thoughts on this?)
- design and integrity (correctness in face of bitwise non-reproducibility)
- multiphysics software, refactoring legacy codes (balance between evolution and revolution)
- ***Design of verification regime***
 - testing and reproducibility
 - The design of tests should reflect the production requirements instead of focus on one or other type of testing.
 - Code coverage is more challenging to verify.

What role can funding agencies play in fostering efforts?

FOAs that require touching on SE for CSE strategy

- Modeled after data needs
- Requires a conversation between PIs and SE for CSE experts
- Requires someplace/document to touch on what these strategies are.
- Care to not be too specific
 - need to encourage, not squash creativity in how to adapt SE for CSE
 - need to avoid onerous requirements that can't reasonably be expected of small teams

How can SW engineering improvements be incorporated and sustained on small CSE teams with few dedicated SW development resources?

Evangelize, effect cultural change, convince people that they should use them

Regression testing can generally be introduced in a lightweight manner, becoming more sophisticated over time.

Containers with SE tools ? Decreasing learning curve and administration, such as repository, automated testing tools etc.

Software carpentry like canned tutorials that teams can download and use internally -- communication from peers who have adopted -- ties into the quantification from studies of team productivity through repository mining (spread the word)

Howtos in a well publicised web location, workshop

What role can funding agencies play in fostering efforts?

Explicitly ask for SE & CSE teams to propose together for some collaborations

Provision for sabbatical visits from SE community to CSE groups

Design call for proposals around the topics highlighted in earlier slides, require software management and release plan in the submission, and SE practices should be called out as an important criterion for proposal review, strong emphasis on comprehensive code verification

Spread of provenance and reproducible results requirement would give the incentive

Incentive for the SE community to provide tools tailored for small teams in the CSE community (pay attention to FORTRAN also, it still is the de-facto program language for many CSE efforts)