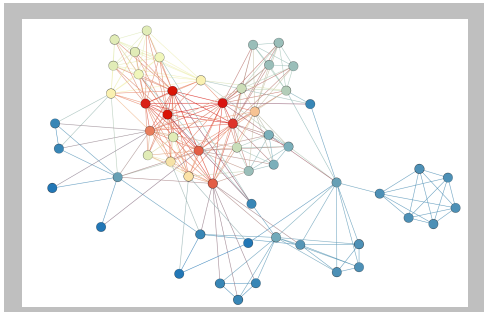
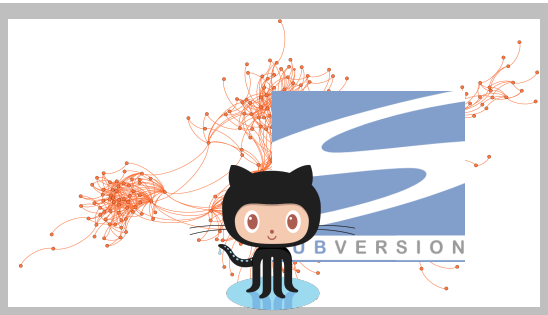
Exceptional service in the national interest

Sandia National Laboratories





Software Engineering for Science: Challenges, Opportunities, and Research Perspectives

Reed Milewicz

SAND2018-6923

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Roadmap

Sandia National Laboratories

- I will…
  - Share with you my background and what I do as a software engineering postdoc working at Sandia.
  - Talk about my team (IDEAS-ECP) and our role in the Exascale Computing Project.
  - Describe some of my ongoing research with the Trilinos team, with whom I am embedded.
  - Give you a better understanding of IDEAS-ECP's engagement and outreach efforts.

Sandia National Laboratories

Background

Who Am I?

Sandia National Laboratories

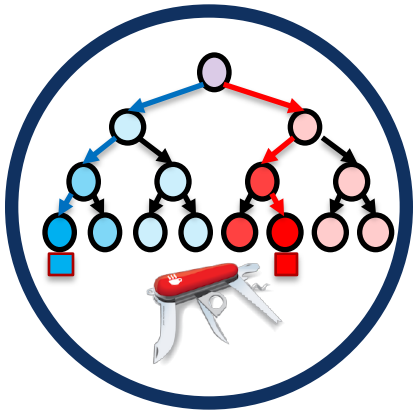
- A software engineering postdoc working at CSRI under Mike Heroux.
- My Areas of Interest:
  - Software Engineering
  - Compilers
  - Formal Verification
- My funding comes from the Interoperable Design of Extreme-scale Application Software (IDEAS) project, an arm of the Exascale Computing Project (ECP).

IDEAS productivity

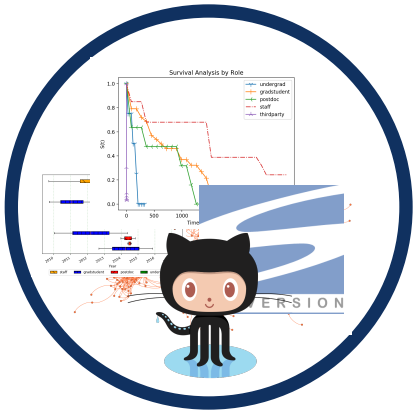
ECP EXASCALE COMPUTING PROJECT

Examples of the kinds of research I do

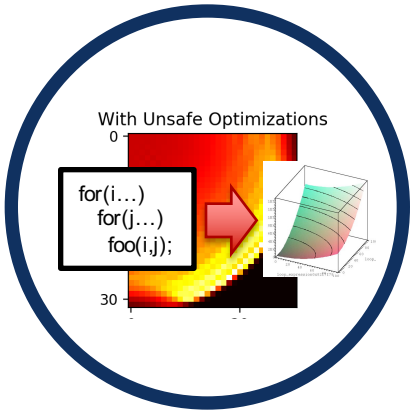
Sandia National Laboratories



Formal verification of complex concurrent programs using model checking algorithms.



Data mining of software repositories to better understand project evolution.



Analyzing error propagation in numerical applications.

Studying LGBT inclusivity in software development (and STEM more broadly).

What do I do?

Sandia National Laboratories

- The scientific software community is facing a crisis due to disruptive changes in computing architectures and new opportunities for greatly improved simulation capabilities.
- IDEAS-ECP has two overarching goals:
  - Improving **developer productivity**, increasing software quality while reducing the effort, time, and cost of development and deployment.
  - Improving **software sustainability**, to maintain and extend a software product over its intended lifespan.
- My research focuses on achieving those goals by doing research on *practices*, *processes*, and *tools* to meet those objectives.

Why?

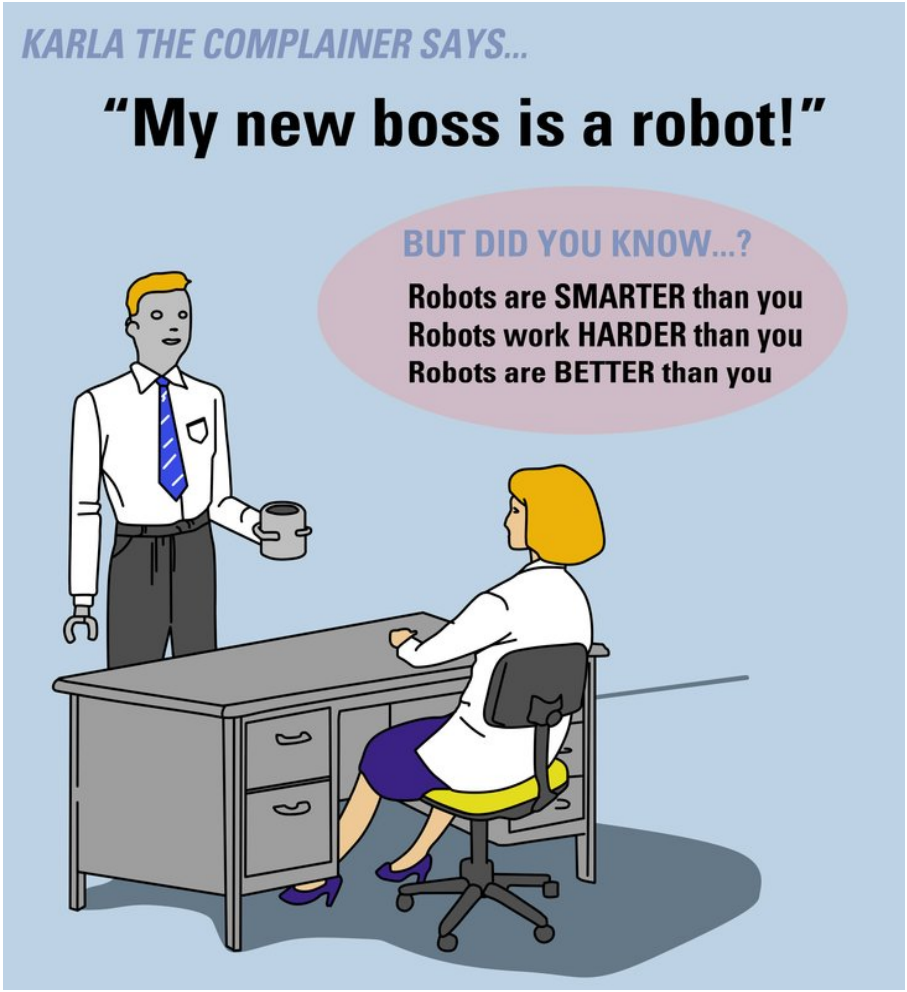
Sandia National Laboratories

- Exascale* computing systems promise to revolutionize science and will profoundly impact our everyday lives.
- However, having the hardware is only useful if we have the software, and developing that software is no easy task.
  - Scientific software projects, such as those developed at Sandia, are among the most complex, knowledge-intensive undertakings in all of human history.
  - The software is largely written by people with no formal training in software development.
  - There is a limited supply of talent, time, and resources, but also a pressing demand for innovation.

** capable of at least 10^{18} operations per second*

What works?

- I have a strong background in tools and automation.
- However, you can't build solutions without understanding what people really need. For that, you have to understand their culture, their practices, and their priorities.



Two Examples

Sandia National Laboratories

- Talk to Me: A Case Study on Coordinating Expertise in Large-Scale Scientific Software Projects
- Productivity and Sustainability Improvement Plans

Sandia National Laboratories

Talk to Me: A Case Study on Coordinating Expertise in Large-Scale Scientific Software Projects

Reed Milewicz and Elaine Raybourn

Motivation

Sandia National Laboratories

- Trilinos is a confederation of object-oriented software packages for building scalable scientific and engineering applications, written in C++.
- It provides packages for (among other things):
  - Iterative and direct solution of linear systems.
  - Parallel multilevel and algebraic preconditioning.
  - Solution of non-linear, eigenvalue and time-dependent problems.
  - Partitioning and load balancing of distributed data structures.

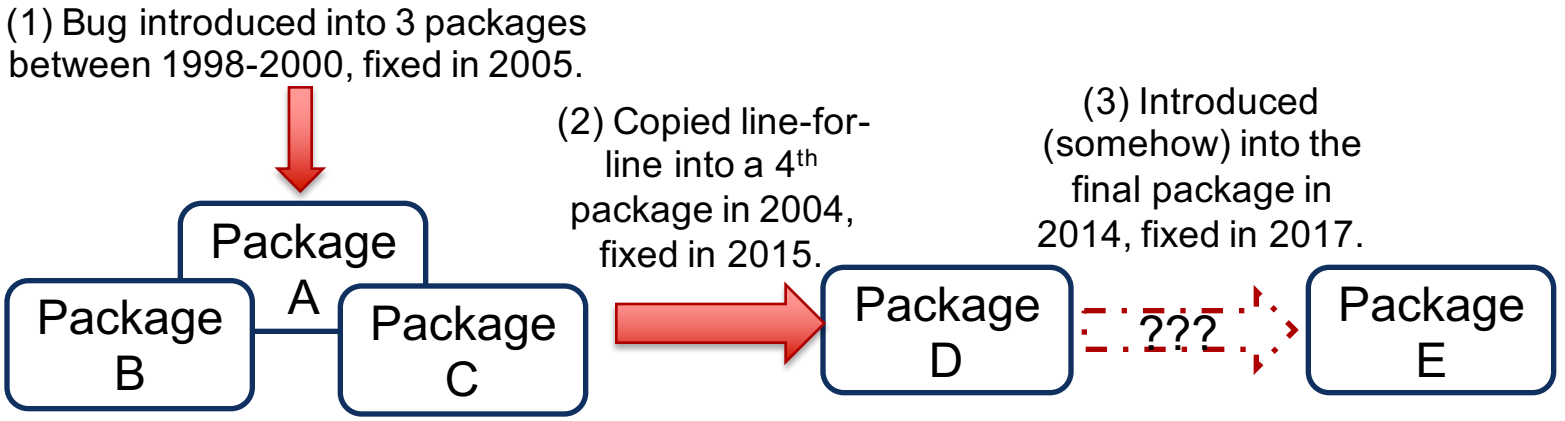
Motivation

Sandia National Laboratories

- **Problem**: A critical application using Trilinos failed a major acceptance test to an unexpected explosion in memory usage when running with more than 131072 MPI progresses.
- A team of researchers was given several months to locate the bug to no avail, but the issue was finally resolved when one heroic Trilinos scientist-developer volunteered three weeks of his time to uncover it.
- The cause of the bug?
  - A misuse of an MPI function due to a misunderstanding about its semantics.
  - An inefficient vendor implementation of that MPI function

Motivation

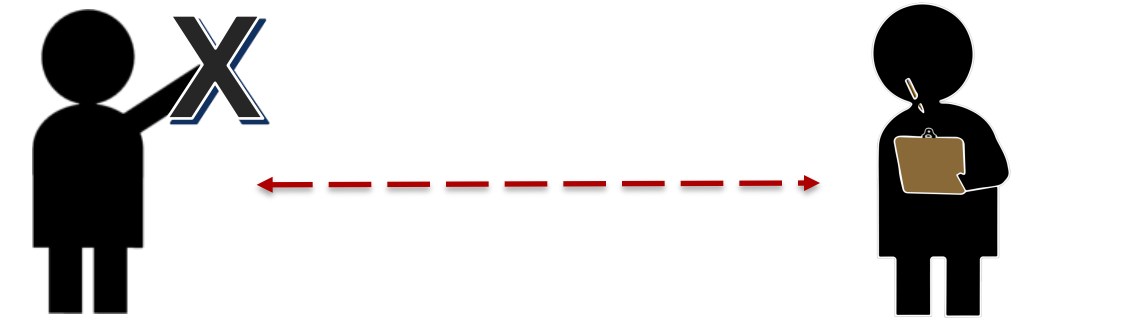
- I conducted my own investigation by interviewing different parties and analyzing source code histories.
- **The deeper mystery**: The *exact same bug* had been fixed twice before!



- In each case, the discovery and solutions were socialized, notes were made, etc. However, the information did not flow to the right parties in each subsequent incident.

Motivation

- I'm not bringing this up to criticize Trilinos. This is actually a really common occurrence in large-scale software projects.
- Large-scale → When it is impossible for everyone to know everything (this problem is amplified in scientific software development!)



- But what do we mean by knowledge?

Motivation

Sandia National Laboratories

Knowledge: A fluid mix of framed experience, values, contextual information, and expert insights that provides a framework for evaluating and incorporating new experiences and information. It originates in and is applied in the minds of knowers. In organizations, it often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices, and norms.

Davenport and Prusak 1998

- It can be tacit or explicit.
- It can be socialized, combined, internalized, or externalized.
- It can include "knowing what", "knowing who", "knowing why", etc.
- It can be lost or unable to be communicated.

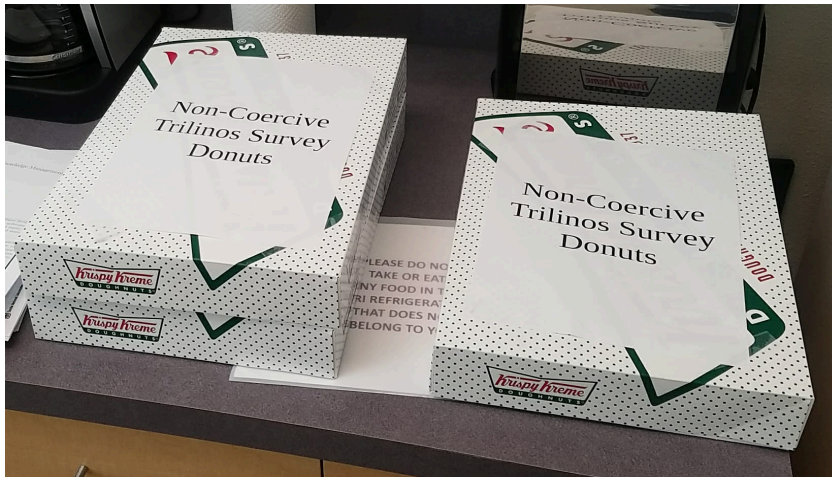
Research Questions

Sandia National Laboratories

- **RQ1**: Do scientific software developers face challenges in sharing their knowledge? If so, what are the challenges?
- **RQ2**: How does individual and organizational knowledge affect those problems?
- **RQ3**: How is that knowledge communicated?

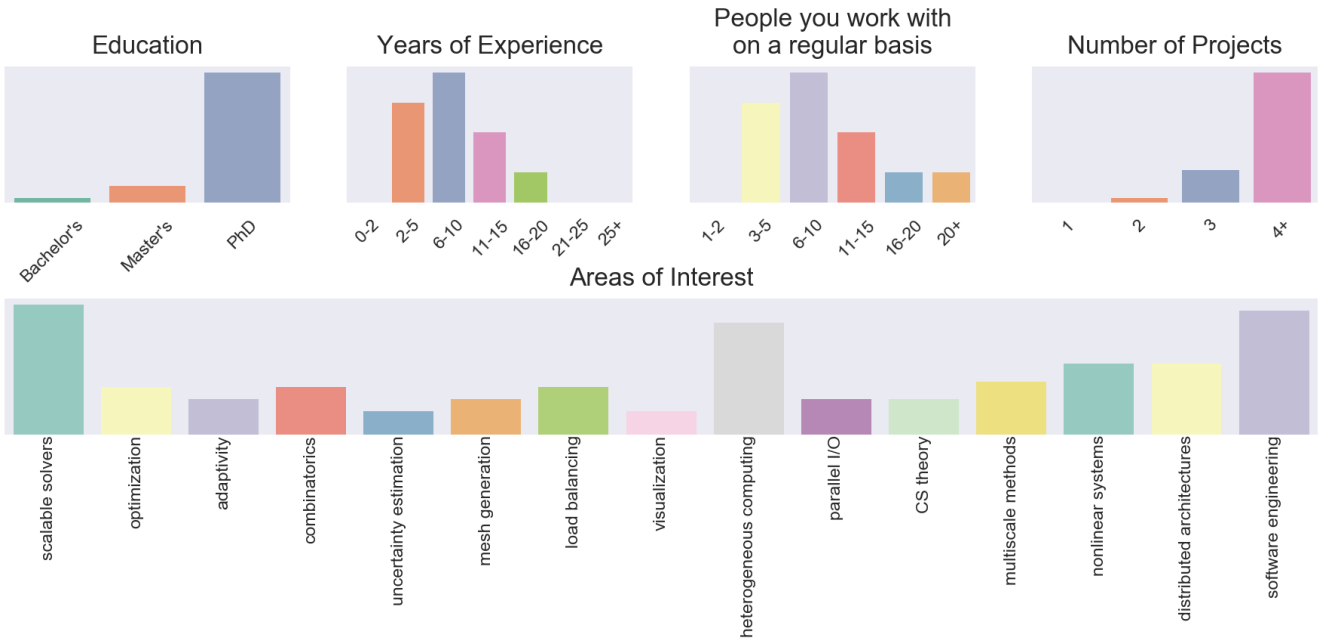
Conducting the Survey

- Survey data was collected over a period of 3 months.
- 36 developers responded, covering 95% of the "main" development group.
- Survey topics:
  - Background and demographic information
  - Career priorities
  - What and who people know
  - How they communicate
  - What problems they face



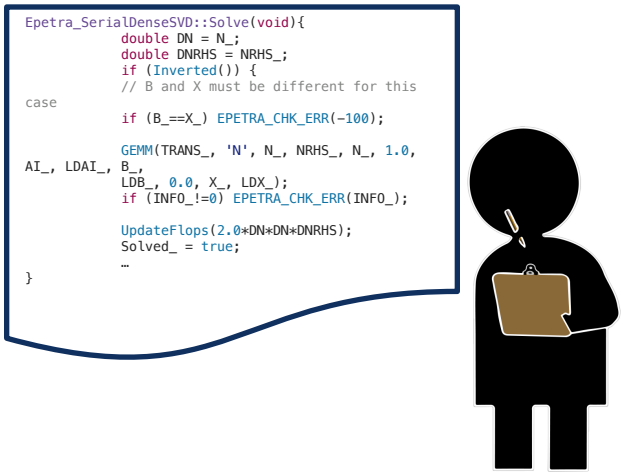
Demographics

- 86% of respondents have completed a PhD. The median respondent had between 11 and 15 years of experience.
- 73% work on 4 or more projects. Most people work regularly with 6 to 10 other people.



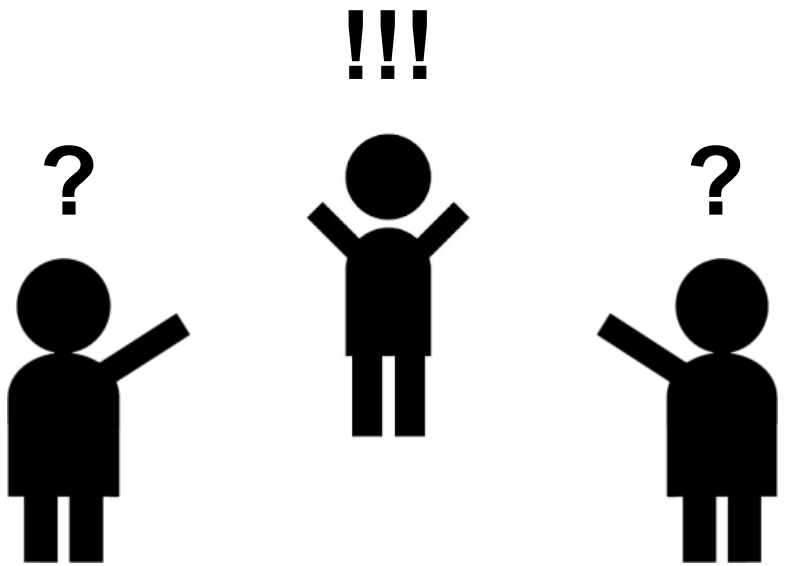
Problems: Code Understanding

Problem	This is a problem (% agree) / a difficult problem (% agree)
Understanding the rationale behind a piece of code	63.9% (13.9%)
Understanding code that someone else wrote	83.3% (22.2%)
Understanding the history of a piece of code	58.3% (8.3%)
Understanding code that I wrote awhile ago	22.2 (0.0%)



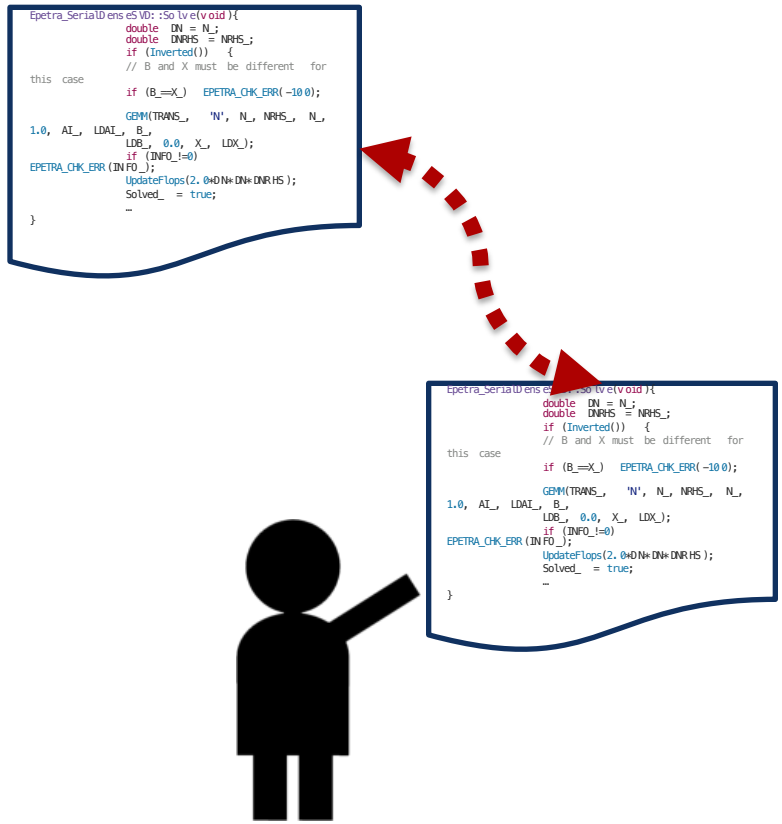
Problems: Task Switching

Problem	This is a problem (% agree) / a difficult problem (% agree)
Having to switch tasks often because of requests from my teammates or manager	75.0% (38.9%)
Having to switch tasks because my current task gets blocked	55.6% (11.1%)
Having to divide my attention between many different projects	94.4% (58.3%)



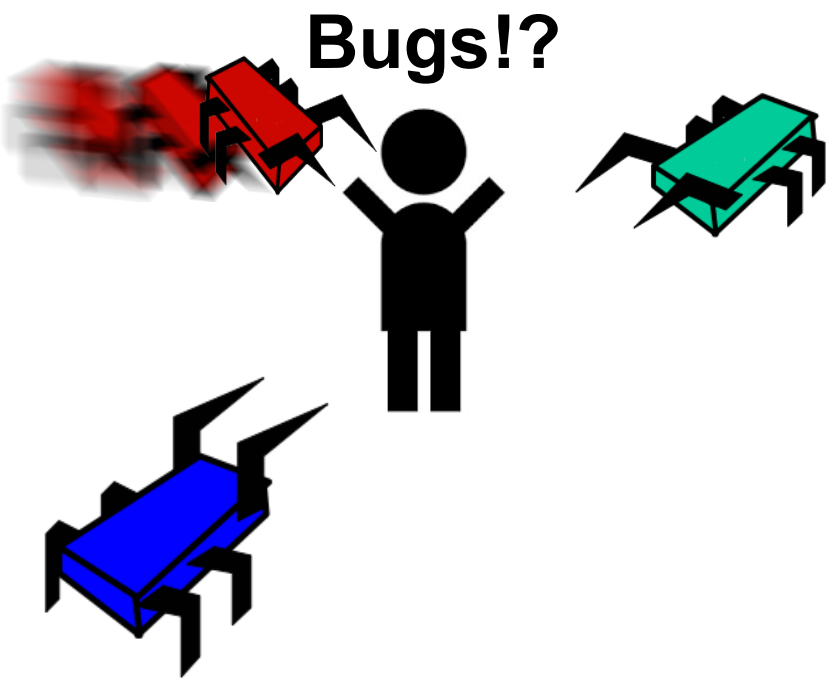
Problems: Modularity

Problem	This is a problem (% agree) / a difficult problem (% agree)
Being aware of changes to code elsewhere that impact my code	58.3% (11.1%)
Understanding the impact of changes that I make on code elsewhere	61.1% (2.8%)



Problems: Links Between Artifacts

Problem	This is a problem (% agree) / a difficult problem (% agree)
Finding all the places code has been duplicated	58.3% (2.8%)
Understanding who “owns” a piece of code	38.9% (0.0%)
Finding the bugs related to a piece of code	75.0% (8.3%)
Finding code related to a bug	83.3% (11.1%)
Finding out who is currently modifying a piece of code	33.3% (0.0%



Problems: Team

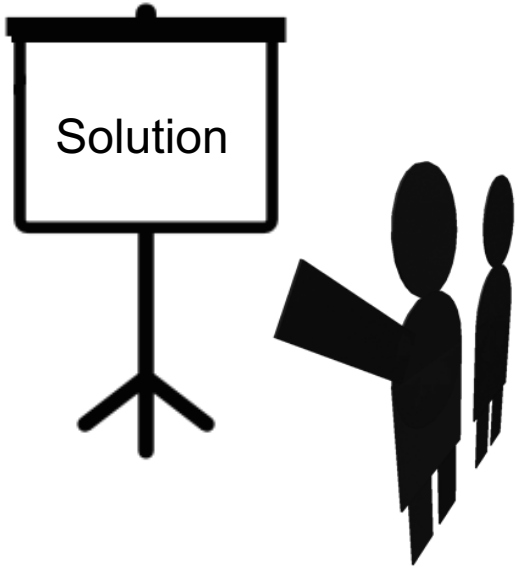
Sandia National Laboratories

Problem	This is a problem (% agree) / a difficult problem (% agree)
Convincing managers that I should spend time architecting, refactoring, or rewriting code	41.7% (25.0%)
Convincing developers to make changes to code that I depend upon	61.1% (16.7%)

Problems: Expertise Finding

Sandia National Laboratories

Problem	This is a problem (% agree) / a difficult problem (% agree)
Finding the right person to talk about a piece of code	50.0% (8.3%)
Finding the right person to talk about a bug	38.8% (5.6%)
Finding the right person to review a change before a check-in	25.0% (5.6%)



What we know

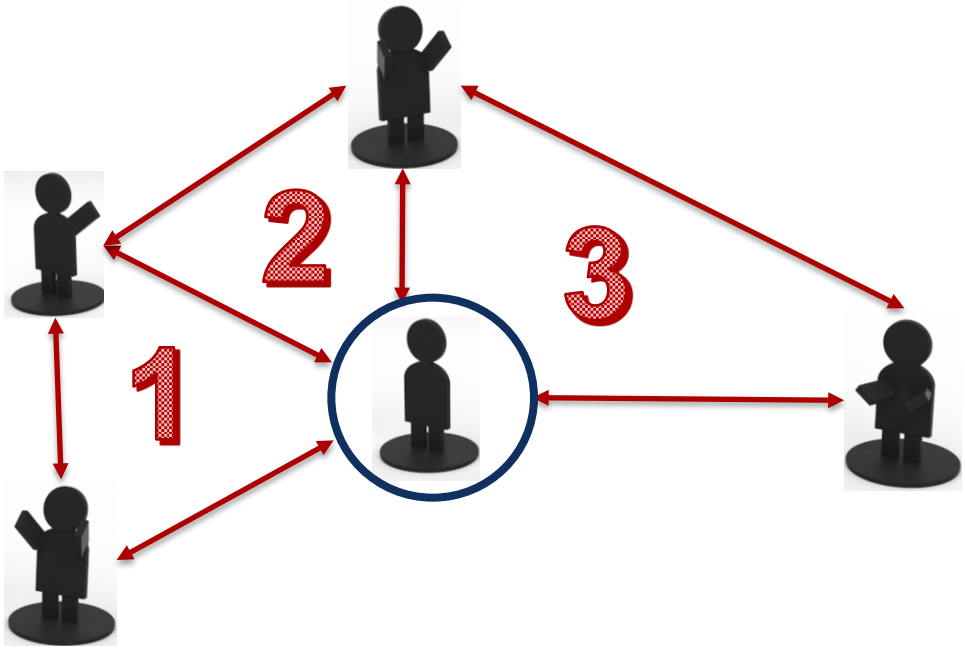
Sandia National Laboratories

- The majority of respondents reported having 13 out of 19 problems. The median respondent reported having 11 problems, 2 of which were considered serious.
- Problems within each category don't tend to correlate well with each other (as measured by Cronbach's alpha).

Category	Cronbach's alpha
Code Understanding	0.770
Task Switching	0.715
Modularity	0.474
Artifacts	0.595
Team	0.594
Expertise Finding	0.579

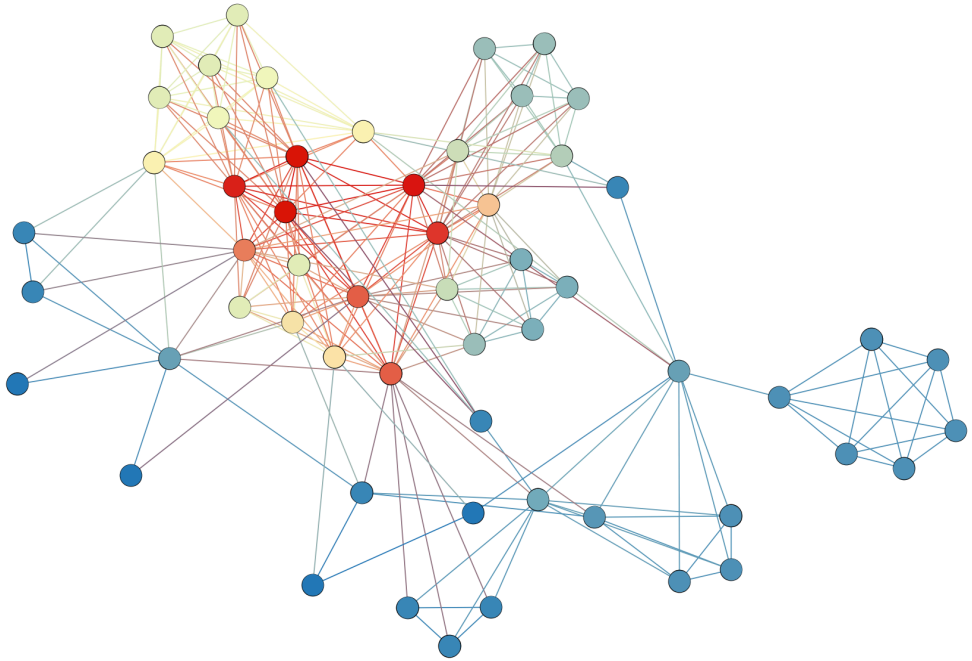
- The data suggests that these problems have multiple independent, latent causes.

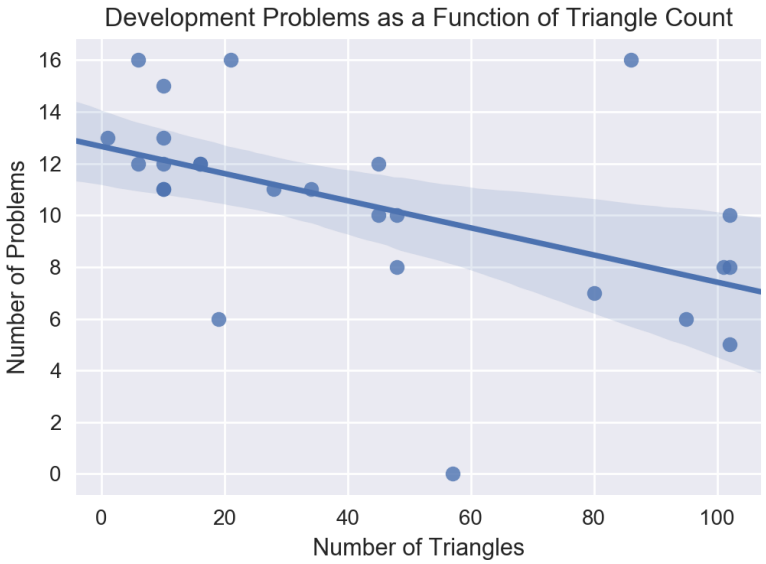
Let's try a naïve network analysis



- Triangle counting: For each vertex, how many cycles of length three can we find that include that vertex?

Team Network vs. Problems



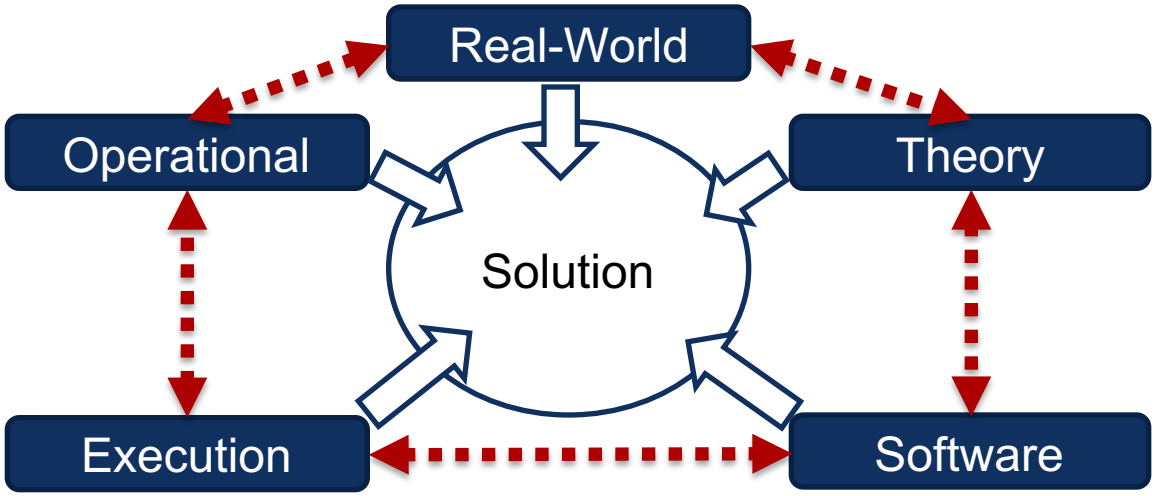


People on the organizational periphery tend to report more problems than people closer to the core. But why?

Is there a simple explanation?

- *Is this a matter of experience?*
  - **No**, only 3/19 problems can be correlated with experience.
- *Is this a matter of the number of people that respondents work with or the number of projects they work on?*
  - **No**, people don't tend to report more or fewer problems based on the number of contacts or projects they have. 2/19 problems can be correlated with the number of projects people have.

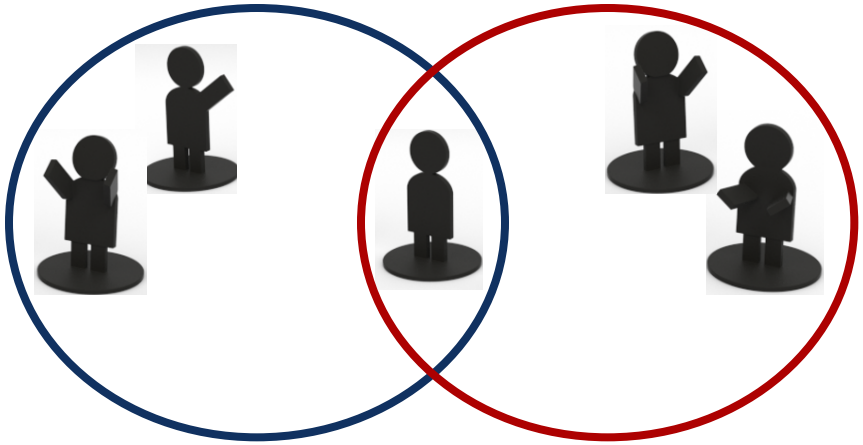
What/who do they know?



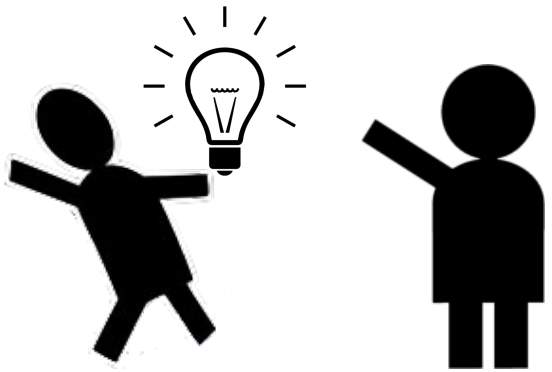
- We used a five factor knowledge model of scientific software development (Kelly 2015). From that, we created a list of topics and had respondents rank their expertise in those areas.
- For each topic, we also asked respondents to indicate whether they knew someone they could "turn to for help" for it.

D. Kelly, "Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software," *Journal of Systems and Software*, vol. 109, pp. 50–61, 2015.

What/who do they know?



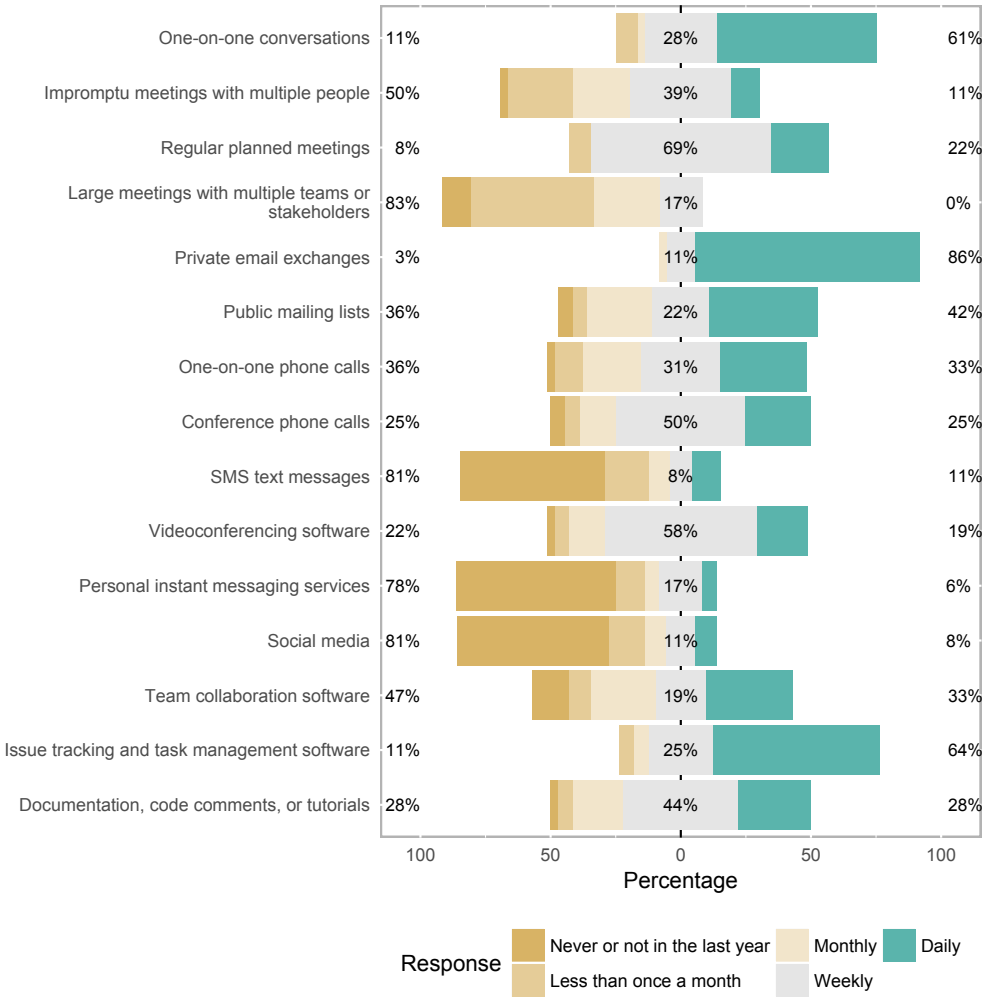
The most useful forms of expertise are those that allow respondents to position themselves between domains of activity.



"Knowing who" is instrumental for maintaining awareness as well as negotiating and coordinating with others.

- Operational and execution domain knowledge have a positive effect on four of the nineteen problems.
- Real-world, theory, and software domain knowledge provided **no** measurable benefit.
- Ten of the nineteen problems are influenced by seeking help from others.

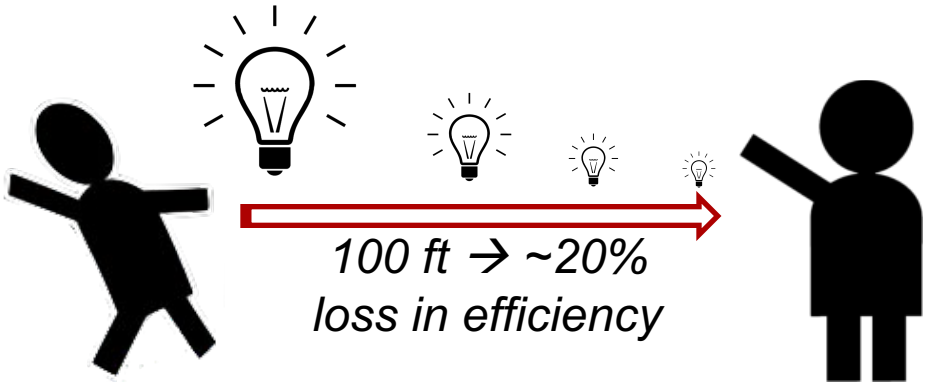
How do they communicate?



- Knowledge scores appear to mirror communication scores; those who communicate more, know more.
- Face-to-face communications enable expertise-finding activities, affecting three of the nineteen problems.
- Digital communication strategies are useful for protecting modularity and understanding the links between artifacts, but the communication overhead also introduces new challenges (e.g. divided attention).

How do they communicate?

Face-to-face communication is important for collaborative problem solving. It's more than just knowing who to talk to, it's about cultivating close relationships with those people.



In fact, there was a recent study of an R&D organization that found a drop-off in collaboration frequency and success after about 100 feet of distance between offices (Kabo et al 2014) .

F. W. Kabo, N. Cotton-Nessler, Y. Hwang, M. C. Levenstein, and J. Owen-Smith, "Proximity effects on the dynamics and outcomes of scientific collaborations," *Research Policy*, vol. 43, no. 9, pp. 1469– 1485, 2014.

So, what can we say about the 19 problems?

Sandia National Laboratories

- They are not cured by time or experience.
- We found no proof to suggest that additional domain or software development training will make them go away.
- However, they are, in some sense, a function of a person's "embeddedness" in the team. More specifically, we found several factors that appear to influence the occurrence of problems:
  - Knowing what
  - Knowing who
  - Communication strategies

Problem Area	Backgrounc	What They Know	Knowing Who Knows	How They Commu-nicate
Code Understanding	(2/4)	(1/4)	(3/4)	
Switching Tasks		(2/3)	(3/3)	(1/3)
Modularity	(1/2)		(1/2)	(1/2)
Links Between Artifacts	(2/5)	(3/5)	(2/5)	(2/5)
Team			(1/2)	(1/2)
Expertise Finding	(1/3)	(1/3)		(3/3)

Recommendations; Future Work

Sandia National Laboratories

- What might help?
  - Empowering knowledge brokers
    - 33% of our respondents knew no one they could turn to for help in any of the knowledge areas while having an average of 5 different problems that could potentially be mitigated by having useful contacts
    - Giving formal recognition and power to the people who know people
  - Cultivating organizational awareness
    - The value of serendipitous encounters
    - Interdepartmental seminars and luncheons
  - Encouraging integrative work (more expensive)
    - 36% of respondents reported having no daily face-to-face interactions with other coworkers
    - Occasional (temporary) team rotations
    - Pair programming for production code

Future Work

Sandia National Laboratories

- Interviews!
- Surveying more teams!
- Look into tool solutions for different classes of problems!

Sandia National Laboratories

Productivity and Sustainability Improvement Plans

The IDEAS Team

Motivation

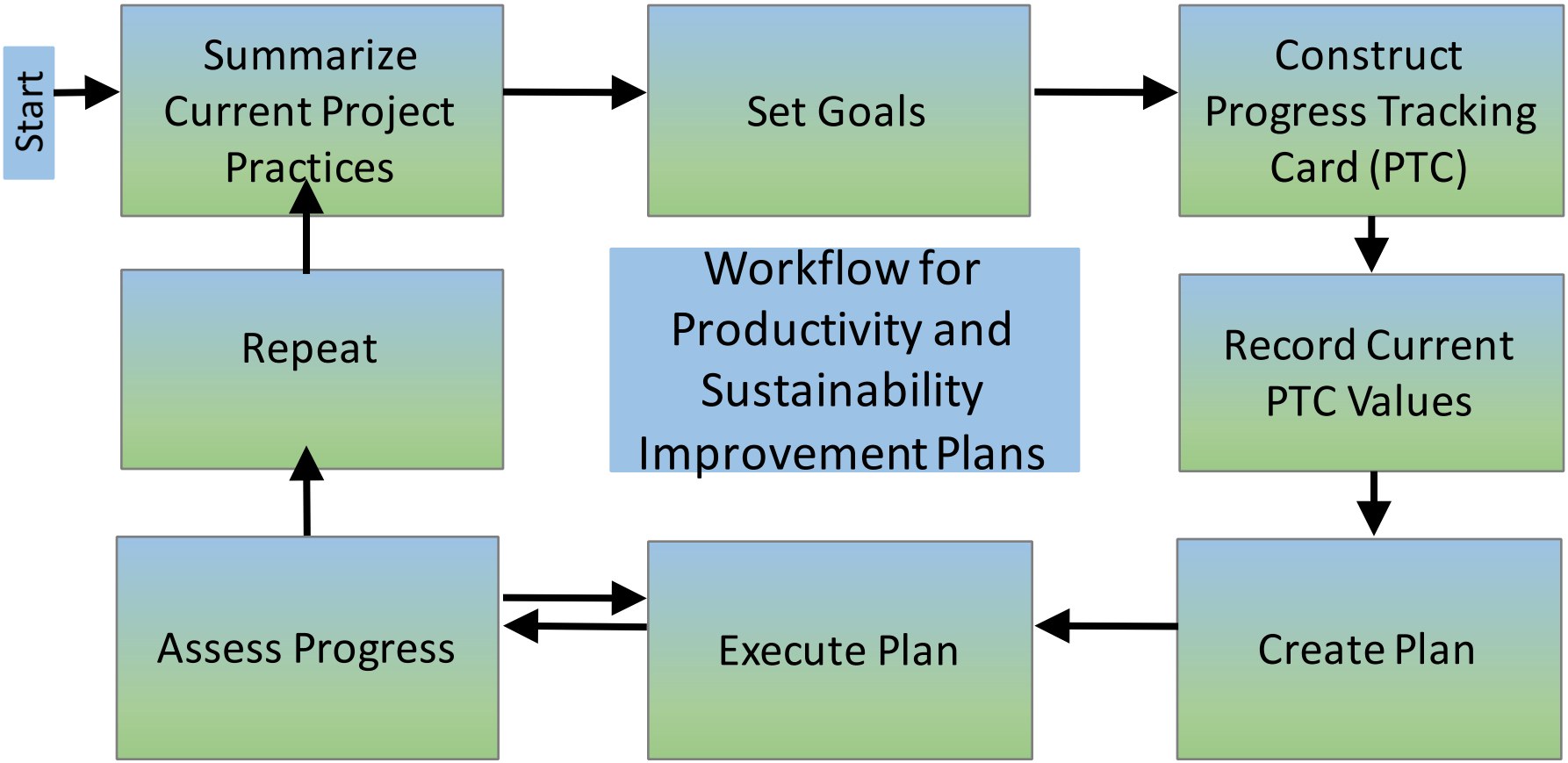
Sandia National Laboratories

- IDEAS-ECP is an interdisciplinary coalition of domain experts, software engineers, and social scientists. Its aim is to ***improve scientific productivity*** through innovative practices, processes, and tools targeting all phases of the software development lifecycle.

What is a PSIP?

- A Productivity and Sustainability Improvement Plan (PSIP) is a **living document** that is a planning and communication tool for capturing and conveying the practices, processes, policies and tools of a given software project.
- The PSIP process helps IDEAS-ECP collect data about current practices as well as giving us the opportunity to help shape the state of practice.

The PSIP Process at a Glance



Tracking Progress

Sandia National Laboratories

- Both near term and long term targets for improvement are naturally identified during the development of the Current Project Practices living document.
- These targets are expressed and recorded using Progress Tracking Cards, usually selected from our growing catalogue.

Practice: Test Coverage		Score (0 – 5):	
Score Descriptions			
0	Little or no independent testing. Functional testing via users.		
1	Independent functional testing of primary capabilities.		
2	Primary functional testing, some unit test coverage.		
3	Comprehensive unit testing, primary functional testing.		
4	Comprehensive unit testing, functional testing for documented use cases.		
5	Comprehensive unit, use case functional testing; test coverage commitment.		

Comments:

1. **Functional testing:** Testing capabilities from user's perspective. Many functions can be called. Good for usability assurance. Insufficient to protect against some regressions. Difficult to isolate regressions. Can require extensive test execution times.
2. **Unit testing:** Isolated, independent testing of functions and methods. Enable test-driven development, rapid test execution, fault isolation. Insufficient to ensure functional correctness.
3. **Comprehensive:** Does not mean 100% line coverage, but sufficient coverage to detect most errors. Experts suggest various metrics such has 80% or more line coverage, or some similar high percentage of function point coverage.
4. **Commitment:** Team is committed to writing comprehensive tests concurrent with functionality.

You're Not Alone!

Sandia National Laboratories

Onboarding

What we lack is a systematic approach to documentation for developers, which adds to the cost of getting a new developer up to speed. […] There is no well-developed process for introducing new developers to code.

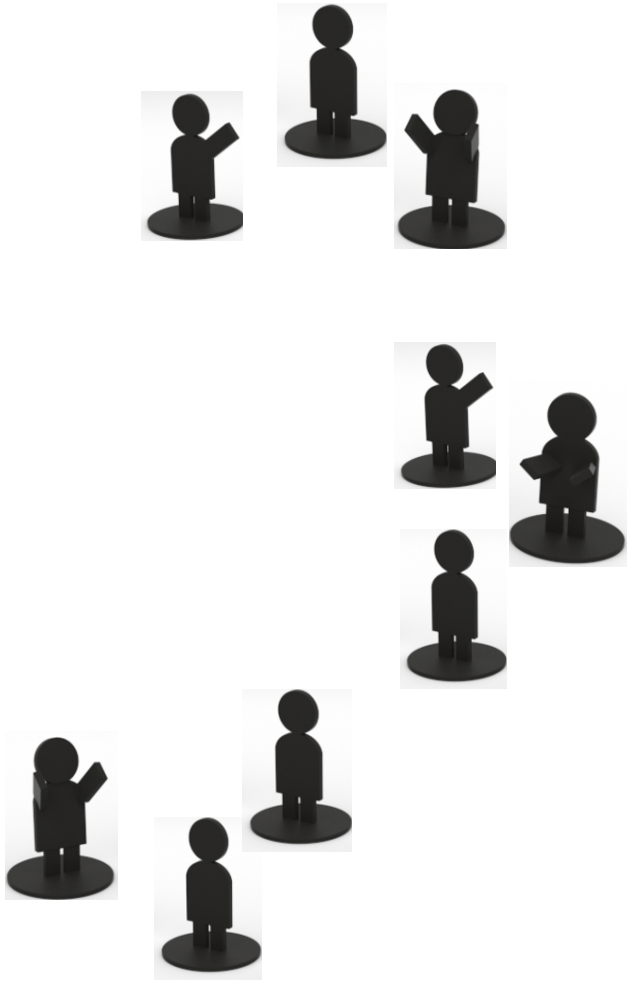
Our project tasks students and postdocs with formulating their own tutorial for how to build and run the code (explain the code, the input files, etc.). As a result, [we have] a "gold standard" for components of this tutorial.

- For every problem one team faces, another team has often found a solution.

IDEAS-ECP Outreach

- IDEAS-ECP has been reaching out to many teams to help, including:

- Alquimia
- Amanzi
- ATS
- CrunchFlow
- ParFlow
- PFLORTRAN
- Trilinos
- MPICH
- VeloC
- Astro
- UnifyCR
- EXAALT
- NWChemX
- QMCPACK
- CANDLE
- MARBL
- SPARC



Related IDEAS-ECP Projects

Sandia National Laboratories

IDEAS aims to socialize best practices through the Better Scientific Software site (https://bssw.io), a central hub for sharing information on practices, techniques, experiences, and tools to improve scientific software productivity, quality, and sustainability.

IDEAS is making significant progress helping teams manage TPLs through its Software Development Kit (xSDK) (https://xsdk.info), which aims to be a foundation of an extensible scientific software ecosystem.

My Thoughts

Sandia National Laboratories

- Scientific software development is **hard**, but there are ways that we can achieve incremental improvements.
- Better practices mean better science. We can reduce costs, increase productivity, and get more science per unit time.
- “You can’t do it alone, value everyone’s strengths” – Allen Robinson (on Tuesday)

Sandia National Laboratories

Questions?