Towards Sustainable Scientific Software via Interoperable Design of Extreme-scale Application Software (IDEAS-ECP)

R. Milewicz, E. Raybourn, and the IDEAS team

IDEAS productivity
www.ideas-productivity.org

EXASCALE COMPUTING PROJECT

Motivation

Now more than ever, the development of software is critical to the practice of science. However, the scientific software community is facing a crisis created by the confluence of disruptive changes in computing architectures and new opportunities for greatly improved simulation capabilities. This crisis brings with it a unique opportunity to fundamentally change how scientific software is designed, developed, and supported.

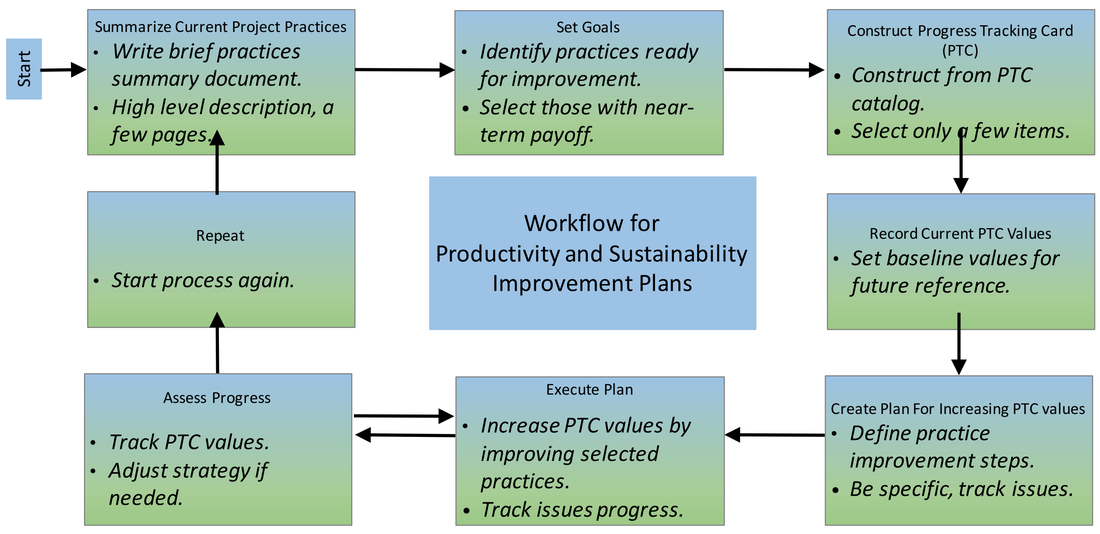
What is IDEAS?

Background. The Interoperable Design of Extreme-scale Application Software (IDEAS) project 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. The initiatives spearheaded by IDEAS all have two goals in mind:

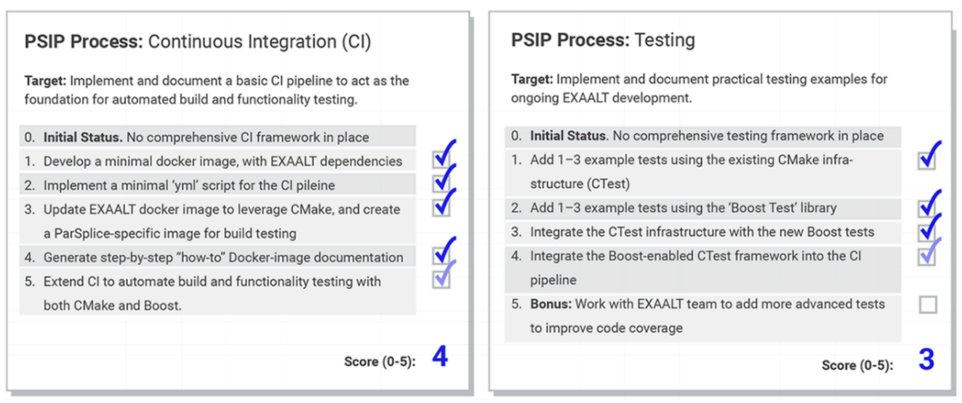
- To improve **developer productivity**, increasing software quality while reducing the effort, time, and cost of development and deployment.
- To improve **software sustainability**, the ability to maintain and extend a software product over its intended lifespan.

Working with Teams to Improve the State of Practice

Incremental Improvement. IDEAS has partnered with numerous teams to develop **Productivity and Sustainability Improvement Plans** (PSIPs). A 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.



Through this dialogue, 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.



PSIP Process: Continuous Integration (CI)

Target: Implement and document a basic CI pipeline to act as the foundation for automated build and functionality testing.

0.	**Initial Status.** No comprehensive CI framework in place	
1.	Develop a minimal docker image, with EXAALT dependencies	✓
2.	Implement a minimal 'yml' script for the CI pileine	✓
3.	Update EXAALT docker image to leverage CMake, and create a ParSplice-specific image for build testing	✓
4.	Generate step-by-step "how-to" Docker-image documentation	✓
5.	Extend CI to automate build and functionality testing with both CMake and Boost.	✓

Score (0-5): 4

PSIP Process: Testing

Target: Implement and document practical testing examples for ongoing EXAALT development.

0.	**Initial Status.** No comprehensive testing framework in place	
1.	Add 1–3 example tests using the existing CMake infra-structure (CTest)	✓
2.	Add 1–3 example tests using the 'Boost Test' library	✓
3.	Integrate the CTest infrastructure with the new Boost tests	✓
4.	Integrate the Boost-enabled CTest framework into the CI pipeline	✓
5.	**Bonus:** Work with EXAALT team to add more advanced tests to improve code coverage	

Score (0-5): 3

Example progress tracking cards developed in collaboration with the EXAALT-ECP project.

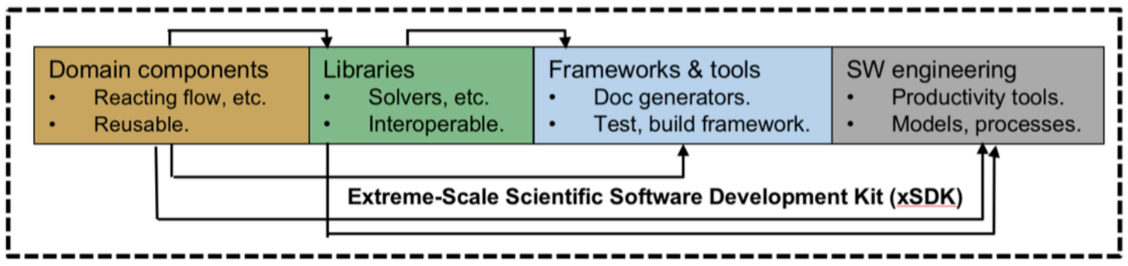
Partnerships. The IDEAS team has partnered with numerous teams to develop improvement plans. These include projects affiliated with the ECP as well as the DOE Biologic and Environmental Research (BER), such as:

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

Building A Healthy Software Ecosystem

Unity in Federation. Use of third-party software (TPLs) typically reduces the cost (time and effort) compared to developing the same capability independently, but, it also increases risk and complexity. 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.

xSDK



Community Policies. xSDK packages (available through the Spack package manager) must satisfy community guidelines to ensure interoperability and usability.

Example Guidelines

M1. Support xSDK community GNU Autoconf or Cmake options.

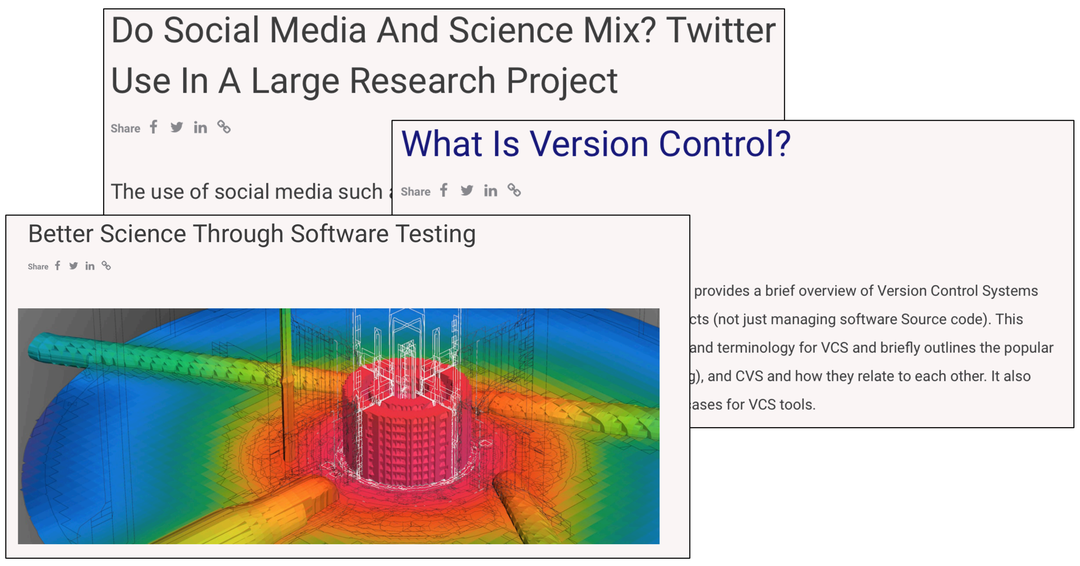
M5. Provide a documented, reliable way to contact the development team.

M11. Have no hardwired print or I/O statements

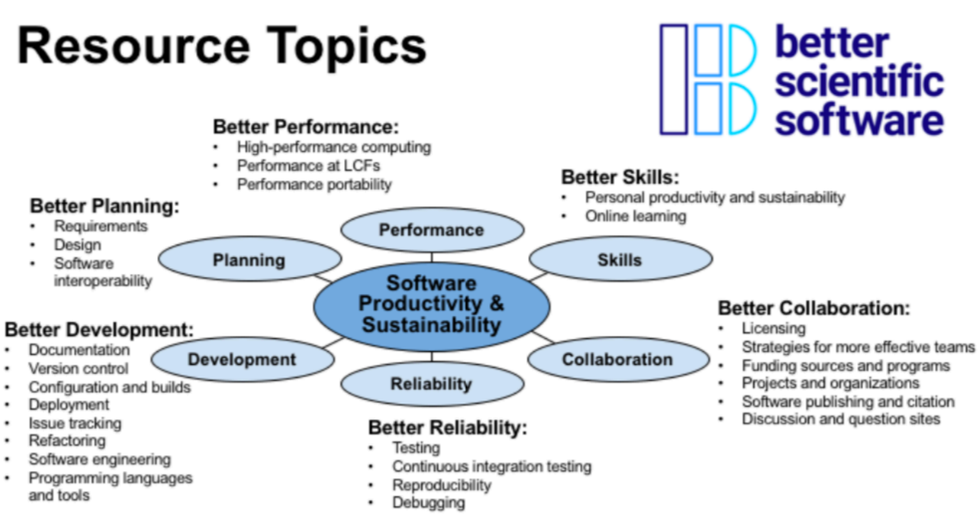
Common Challenges, Common Opportunities

You're not Alone! Our investigations have revealed that scientific software projects face many common problems. More importantly, we have seen teams strive to create innovative solutions to those problems. 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.

better scientific software



BSSW curates a growing body of content on all topics relevant to scientific software development, ranging from what-is and how-to tutorials to experience reports about best practices.



Argonne NATIONAL LABORATORY · BERKELEY LAB · Lawrence Livermore National Laboratory · Los Alamos NATIONAL LABORATORY EST. 1943 · OAK RIDGE National Laboratory · Sandia National Laboratories · UNIVERSITY OF OREGON

WSSSPE 6.1, October 2018

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. SAND2018-12174 C