Introduction to the IDEAS Software Productivity Project

Mike Heroux (SNL), Lois Curfman McInnes (ANL), J. David Moulton (LANL), David Bernholdt (ORNL), Hans Johansen (LBNL) And all IDEAS project members

November 2015



Confluence of trends

2

Fundamental trends:

- Disruptive HW changes: Requires thorough algorithm/code refactoring
- Demands for coupling: Multiphysics, multiscale

Challenges:

- Need refactorings: Really, continuous change
- Modest app development funding: No monolithic apps
- Requirements are unfolding, evolving, not fully known a priori

Opportunities:

- Better design and SW practices & tools are available
- Better SW architectures: Toolkits, libraries, frameworks
- Basic strategy: Focus on productivity



DEAS Interoperable Design of Extreme-scale productivity Application Software (IDEAS)

Motivation

Enable *increased scientific productivity*, realizing the potential of extremescale computing, through a new *interdisciplinary and agile approach to the scientific software ecosystem*.

Objectives

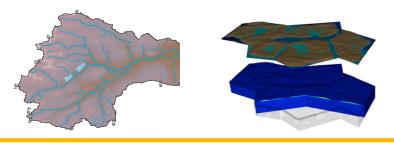
Address confluence of trends in hardware and increasing demands for predictive multiscale, multiphysics simulations.

Respond to trend of continuous refactoring with efficient agile software engineering methodologies and improved software design.



Impact on Applications & Programs

Terrestrial ecosystem **use cases tie IDEAS to modeling and simulation goals** in two Science Focus Area (SFA) programs and both Next Generation Ecosystem Experiment (NGEE) programs in DOE Biologic and Environmental Research (BER).



Approach

ASCR/BER partnership ensures delivery of both crosscutting methodologies and metrics with impact on real application and programs.

Interdisciplinary multi-lab team (ANL, LANL, LBNL, LLNL, ORNL, PNNL, SNL)

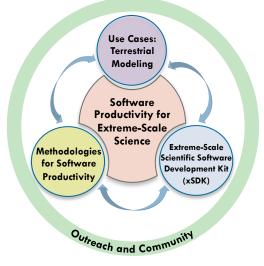
ASCR Co-Leads: Mike Heroux (SNL) and Lois Curfman McInnes (ANL) **BER Lead:** David Moulton (LANL)

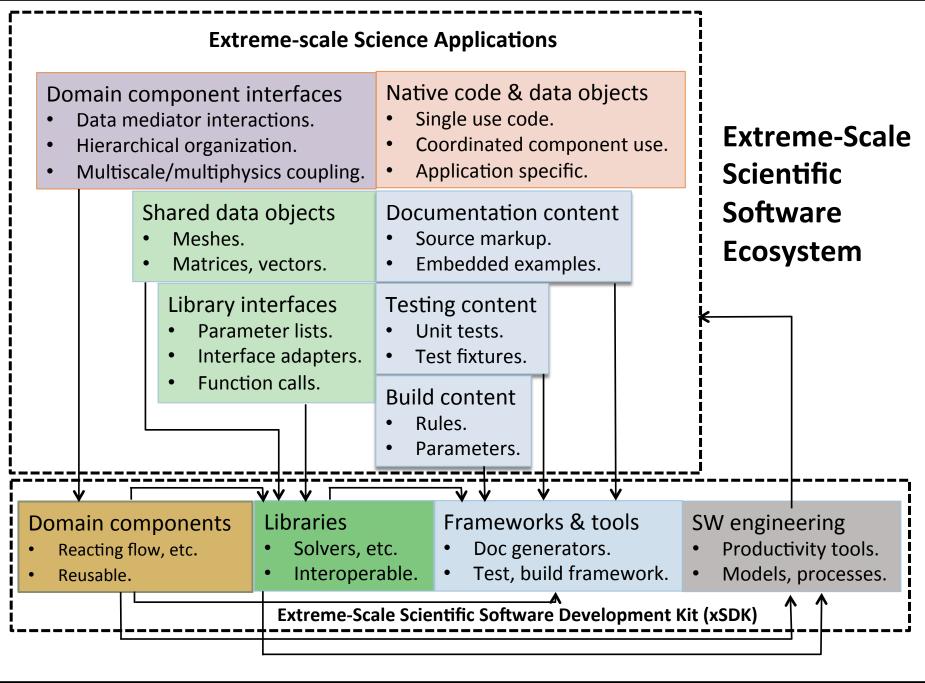
Topic Leads: David Bernholdt (ORNL) and Hans Johansen (LBNL)

Integration and synergistic advances in three communities deliver scientific productivity; outreach establishes a new holistic perspective for the broader scientific community.

www.ideas-productivity.org







IDEAS 'What is' and 'How to' docs

- Motivation: Scientific software teams have a wide range of levels of maturity in software engineering practices
 - Baseline survey of xSDK and BER Use Case teams
- **Approach**:
 - 'What Is' docs: 2-page characterizations of important software project topics
 - **How To' docs**: brief sketch of best practices
 - Emphasis on ``bite-sized" topics enables CSE software teams to consider improvements at a small but impactful scale

Initial emphasis:

- What is CSE Software Productivity?
- What are Software Testing Practices?
- How to Add and Improve Testing in Your CSE Software Project

Topics in progress:

- Refactoring tools and approaches
- Best practices for using interoperable libraries
- Designing for performance portability

https://ideas-productivity.org/resources/howtos

What Are Software Testing Practices? IDEAS The IDEAS Scientific Software Productivity Project productivity What www.ideas-productivity.org Motivation: Software requires regular extensive test to maintain portability to a wide variety of (ev How to Add and Improve Testing **IDE** · to allow refactoring or the addition of new fea productivity in Your CSE Software Project unknowingly introduce new errors, or reintrod to produce correct results for users. The IDEAS Scientific Software Productivity Project www.ideas-productivity.org In this document, we introduce some terminology of and general approaches to testing. Overview: Adding tests of sufficient coverage and quality improves confidence in software and makes it easier to change and extend. Tests should be added to existing code before the code is changed. Tests should be added to new code before (or while) is being written. These tests then become the foundation of a regression test suite that helps effectively drive fature development and improves borgivem soutamatibility. Types and granularities of testing: Software engin testing (see Definition and Categorization of Tests for · Verification testing: Tests that verify that the Target Audience: CSE software project leaders and developers who are facing significant relactioning efforts because of hardware architecture changes or increased demands for multiphysics and multiscale coupling, and who want to increase the quality and speed of development and reduce development and maintenance costs. No-change (often, perhaps mistakenly, call code produces the same results (to an approx Having comprehensive no-change unit tests code (refactoring) but quickly verify that the re Purpose: Show how to add quality testing to a project in order to support efficient modification of existing code or addition of new code. Show how to add tests to support adding a new feature, (2) fixing a bug. (3) improving the design and implementation of the support of In addition, three granularities of testing are recogn · Unit tests: Focus on testing individual software (4) optimizing resource usage individual classes Prerequisites: First read the document <u>What Are Software Testing Practices?</u> and browse through Definition and Categorization of Tests for CSE Software. Integration tests: Focus on testing the inter the full system level System-level tests: Focus on testing the full a Set up automated builds of the code with high warning levels and eliminate all level. For example, a system-level test of a C warnings. 2. Select test harness frameworks input files, running the full simulation code, an solutions Managing and reporting on testing: The simplest runs one or more executables, saving the output into examine. Once a package becomes too complex, thi satisfactory and requires various enhancements. Aut called test harnesses) are introduced to lower the (d outputs. b. Set up no-change or verification tests with a system-level test harness in order to pin down importar behavior. Add integrataria and writt tests (an exeluid for adding/changing code) a. Incorporate tests (1, 2) for code to be changed incorporate tests (1, 2) for code to be changed in testing change positiss for thigh change or new code device of the system of the system of the system of the system in test of the system of the system of the system of the system in test of the system of the syste adding new tests. For example, filters can be autom indicates problems (e.g., here and here), and to displ color) which build instantiations generated errors (e.g requires developers to check a website on a regular ¹ Regression -- a return to a former or less developed state. The return to a less developed state. · Cover targeted code to be changed with sufficient (characterization) This material is based upon work supported by the U.S. Dep Computing Research and Biological and En ased upon work supported by the U.S. Department of Energy Office of Science arch and Biological and Environmental Research programs. DBAFT Version 0.1 April 27, 2014

Impact: Provide baseline nomenclature and foundation for next steps in SW productivity and SW engineering for CSE teams.

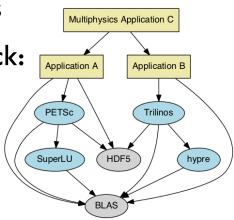


Etc.

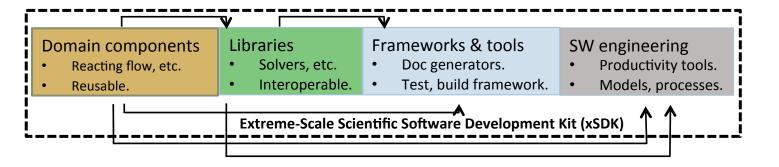
xSDK: Working toward the development of a highly effective extreme-scale scientific software ecosystem

Focus: Increasing the functionality, quality, and interoperability of important scientific libraries and development tools

- xSDK foundations: Seeking community feedback: <u>https://ideas-productivity.org/resources/xsdk-docs/</u>
 - xSDK package compliance standards
 - xSDK standard configure and CMake options
- Library interoperability
- Designing for performance portability



Standard xSDK package installation interface facilitates combined use of xSDK libraries (initially hypre, PETSc, SuperLU, Trilinos), as needed by BER use cases and other multiphysics apps.





Better <u>software productivity</u> is essential for extreme-scale CSE

Better SW productivity can give us better, faster and cheaper

- **Better:** Science, portability, robustness, composability
- **Faster:** Execution, development, dissemination
- **Cheaper:** Fewer staff hours and lines of code

IDEAS project

- Enabling production of high-quality science results, rapidly and efficiently
 - Multiscale terrestrial ecosystem science
 - Broadly: DOE extreme-scale scientific apps
- Delivering first-of-a-kind extremescale scientific software ecosystem
 - xSDK
 - SWP methodologies ("HowTo")
 - Outreach and community

Essential mechanism for progress

- In time of disruptive change
- In presence of multiple design tradeoffs

