

# IMPROVING SCIENTIFIC SW DEVELOPMENT PRODUCTIVITY

Management Plan, Proposed Future Work

# A Confluence of Trends

2

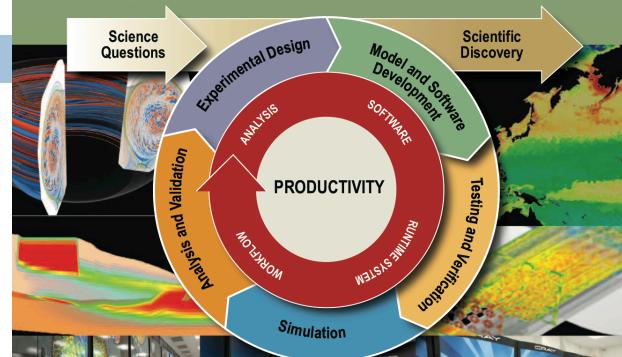
- Fundamental trends:
  - Disruptive HW changes: Requires thorough alg/code refactoring.
  - Demands for coupling: Multiphysics, multiscale.
- Challenges:
  - Need 2 refactorings:  $1 + \varepsilon$ , not  $2 - \varepsilon$ . Really: Continuous change.
  - Modest app development funding: No monolithic 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.

# Productivity Emphasis

3

- Scientific Productivity.
- Many design choices ahead.
- Productivity emphasis:
  - Metrics.
  - Design choice process.
- Software ecosystems: Rational option
  - Not enough time to build monolithic.
  - Too many requirements.
  - Not enough funding.
- Focus on actionable productivity metrics.
  - Optometrist model: which is better?
  - Global model: For “paradigm shifts”.

Software Productivity  
for Extreme-Scale Science  
DOE Workshop Report  
January 13-14, 2014, Rockville, MD



Extreme-Scale Scientific Application  
Software Productivity:

Harnessing the Full Capability of Extreme-Scale Computing

September 9, 2013



Hans Johansen (LBNL), David E. Bernholdt (ORNL), Bill Collins (LBNL),  
Michael Heroux (SNL), Robert Jacob (ANL), Phil Jones (LANL),  
Lois Curfman McInnes (ANL), J. David Moulton (LANL),  
Thomas Ndousse-Fetter (DOE/ASCR), Douglass Post (DOD), William Tang (PPPL)

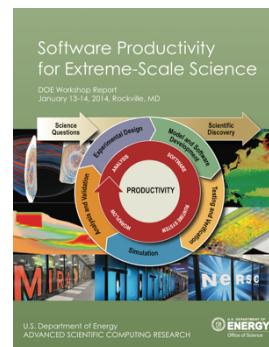
## Motivation

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

## Objectives

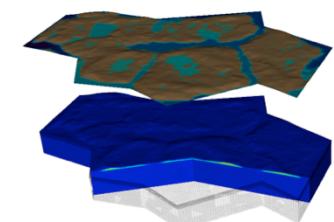
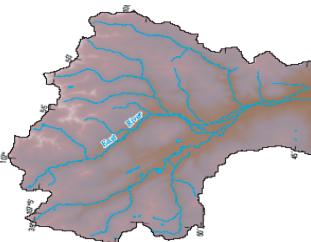
Address confluence of trends in hardware and increasing demands for predictive multiscale, multiphysics simulations.

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.

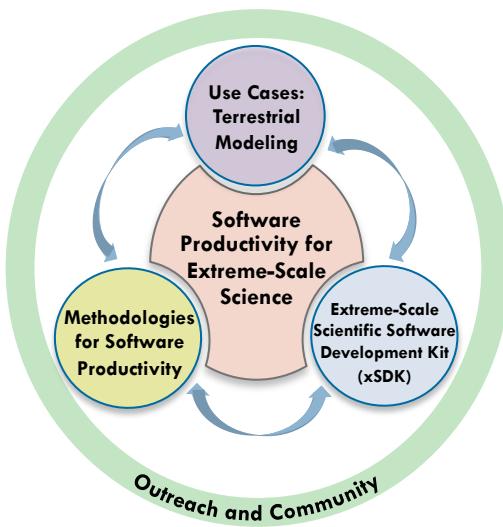
**Interdisciplinary multi-lab team** (ANL, LANL, LBNL, LLNL, ORNL, PNNL, SNL)

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.



# Institutional Leads (Pictured)

## Full Team List

5

### Science Use Cases

J. David Moulton



Tim Scheibe

Carl Steefel

Glenn Hammond

Reed Maxwell

Scott Painter

Ethan Coon

Xiaofan Yang

Hans Johansen

Lois Curfman McInnes

Ross Bartlett

Todd Gamblin\*

Andy Salinger\*

Jason Sarich

Jim Willenbring

Pat McCormick



**Methodologies for Software Productivity**



### Project Leads

ASCR: M. Heroux and L.C. McInnes

BER: J. D. Moulton

### Extreme-Scale Scientific Software Development Kit (xSDK)



Mike Heroux

Ulrike Meier Yang

Jed Brown

Irina Demeshko

Kirsten Kleese van Dam

Sherry Li

Daniel Osei-Kuffour

Vijay Mahadevan

Barry Smith

### Outreach

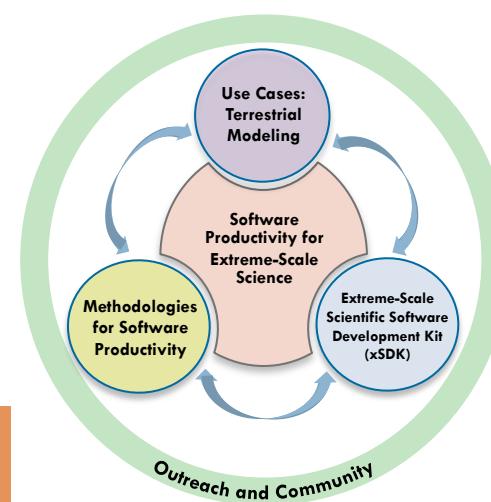
David Bernholdt



Katie Antypas\*

Lisa Childers\*

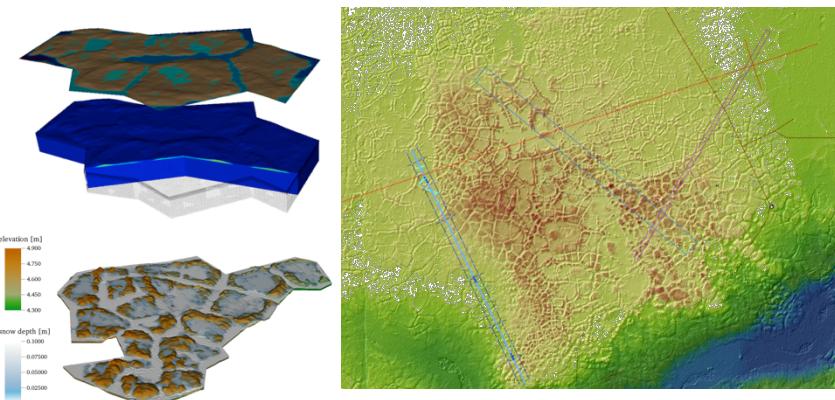
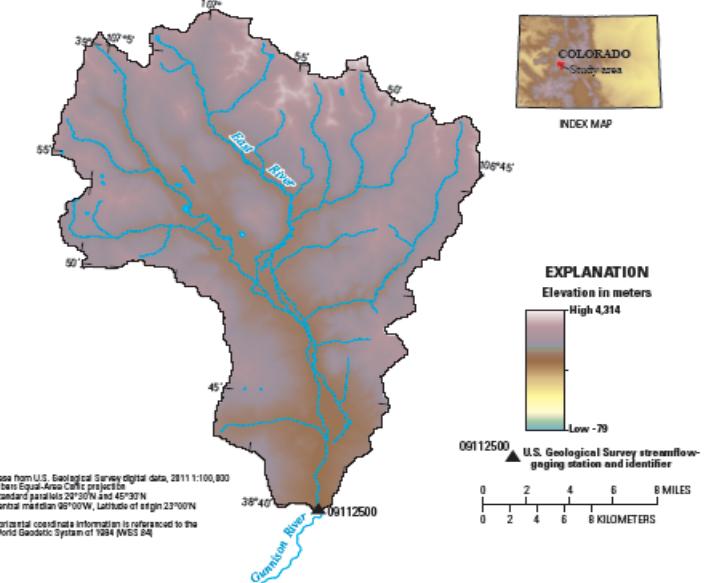
Judith Hill\*



# Use Cases: Multiscale, Multiphysics Representation of Watershed Dynamics

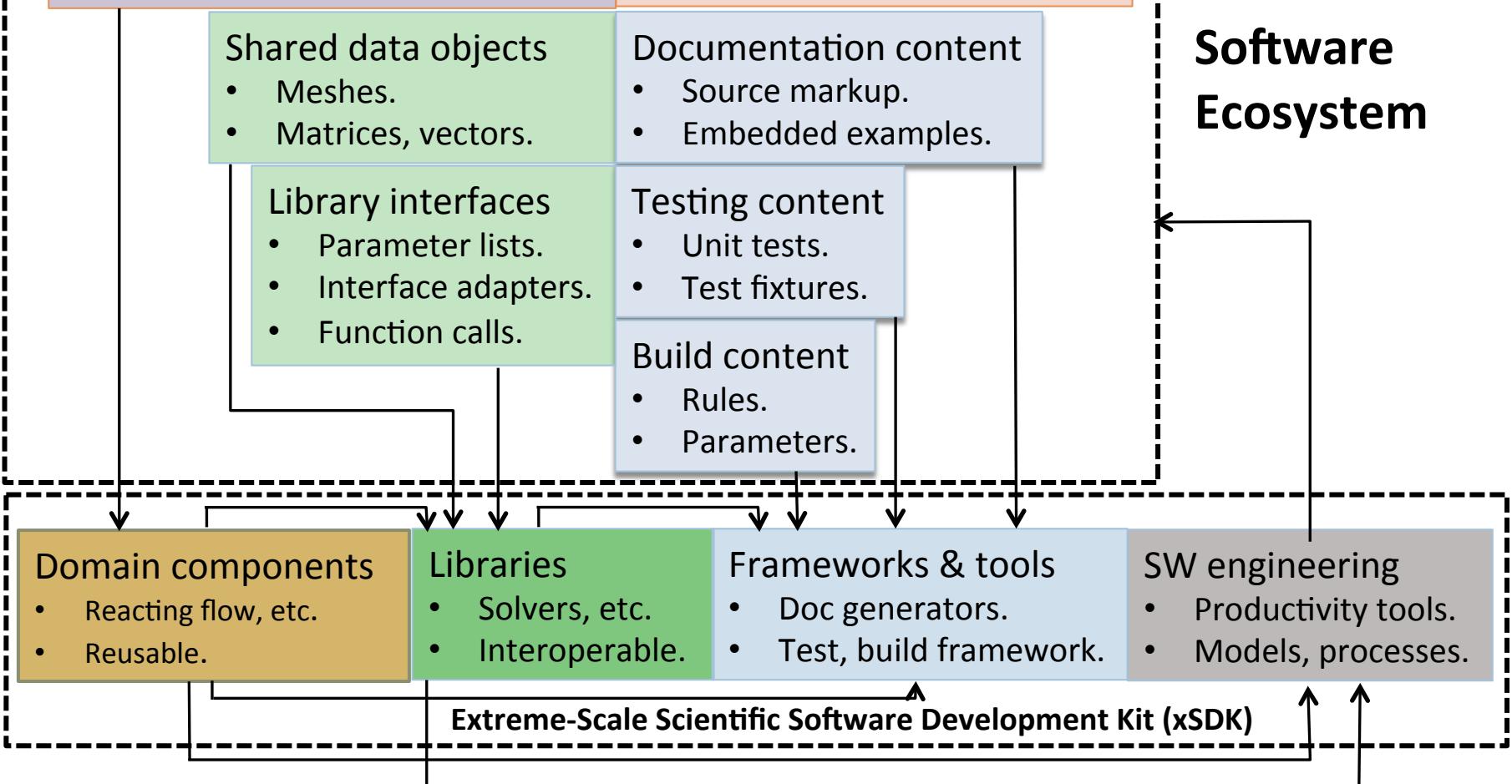
6

- **Use Case 1:** Hydrological and biogeochemical cycling in the Colorado River System.
- **Use Case 2:** Thermal hydrology and carbon cycling in tundra at the Barrow Environmental Observatory.
- **Leverage and complement existing SBR and TES programs:**
  - LBNL and PNNL SFAs.
  - NGEE Arctic and Tropics.
- **General approach:**
  - Leverage existing open source application codes.
  - Improve software development practices.
  - Targeted refactoring of interfaces, data structures, and key components to facilitate interoperability.
  - Modernize management of multiphysics integration and multiscale coupling.



## Extreme-scale Science Applications

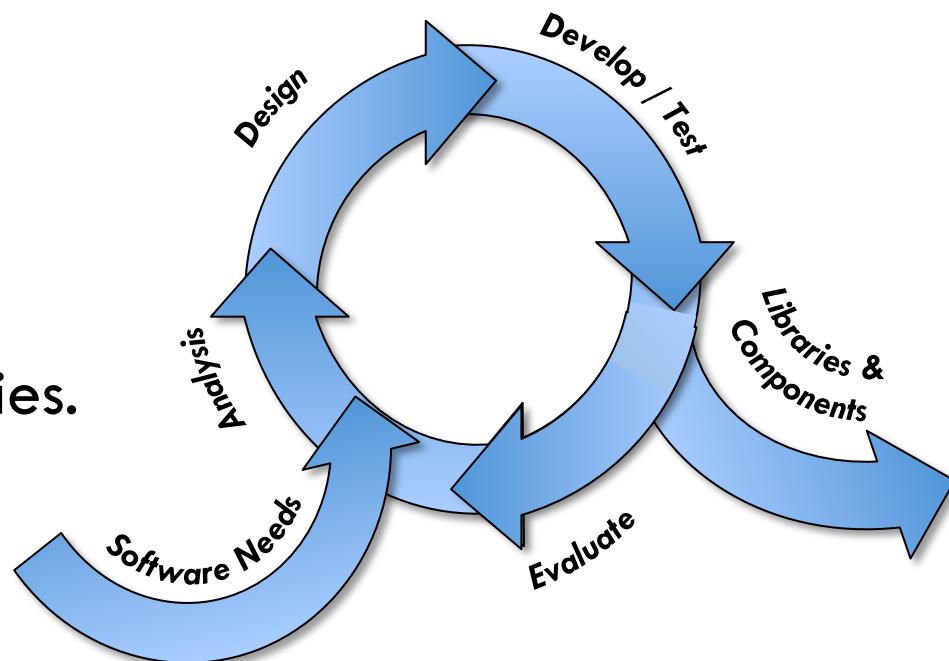
# Extreme-Scale Scientific Software Ecosystem



# Agile, Iterative, and Incremental Cycles for Extreme-Scale Science

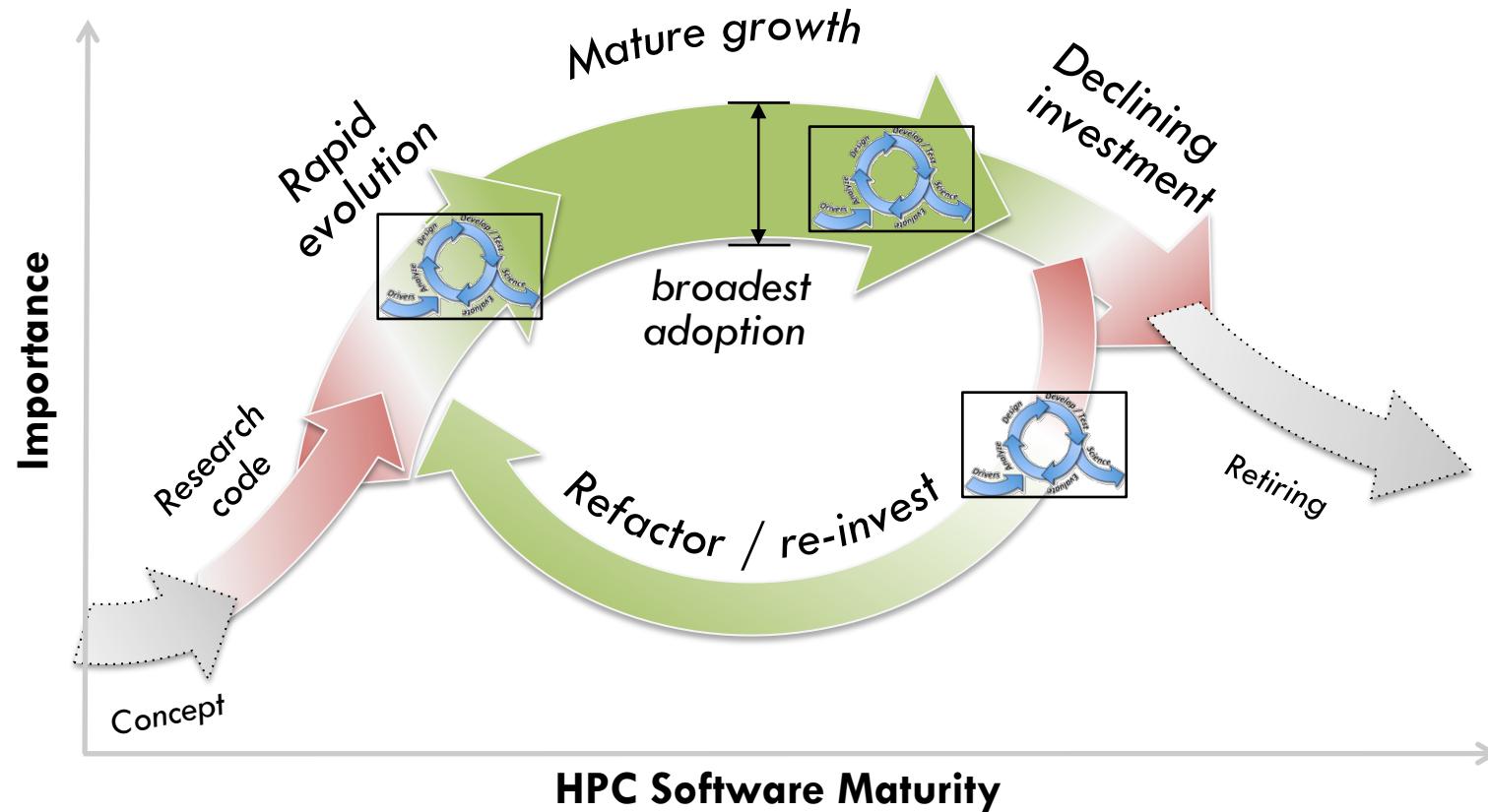
8

- IDEAS SW Engineering:
  - Extreme-scale focus.
  - Lean/agile, adapted for science.
- Metric-centric:
  - Define, use.
- Reusability:
  - Components, libraries.



# Methodology Relevant for Lifecycle Needs

9

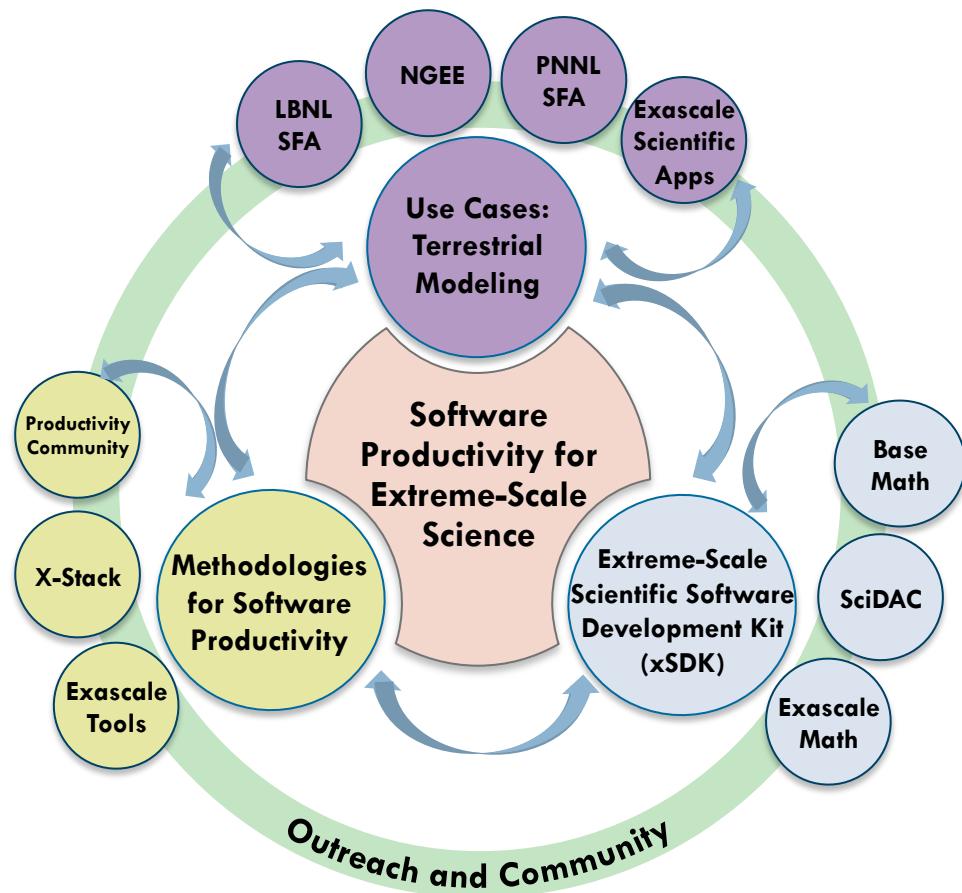


**Goal:** Put steps in place to encourage adoption and reuse of research libraries, and improve longevity of ASCR software investments through refactoring, componentization.

# IDEAS fills critical gaps and mitigates risks for DOE's long-term plans for extreme-scale science.

10

**IDEAS: Software productivity for BER terrestrial modeling and the broader science community.**



- **IDEAS leverages:**
  - ASCR math and CS research.
  - BER science programs.
  - Work on next-generation, extreme-scale algorithms and software.
  - ASCR Co-Design Centers, NNSA PSAAP2, etc.
  - Facilities partnerships (ALCF, NERSC, OLCF)
- **Focus on productivity:**
  - Enhances value of all activities.
    - Sustainable componentization
    - Methodologies and best practices
  - Essential mechanism for progress:
    - For current time of disruptive change.
    - In presence of multiple design tradeoffs.

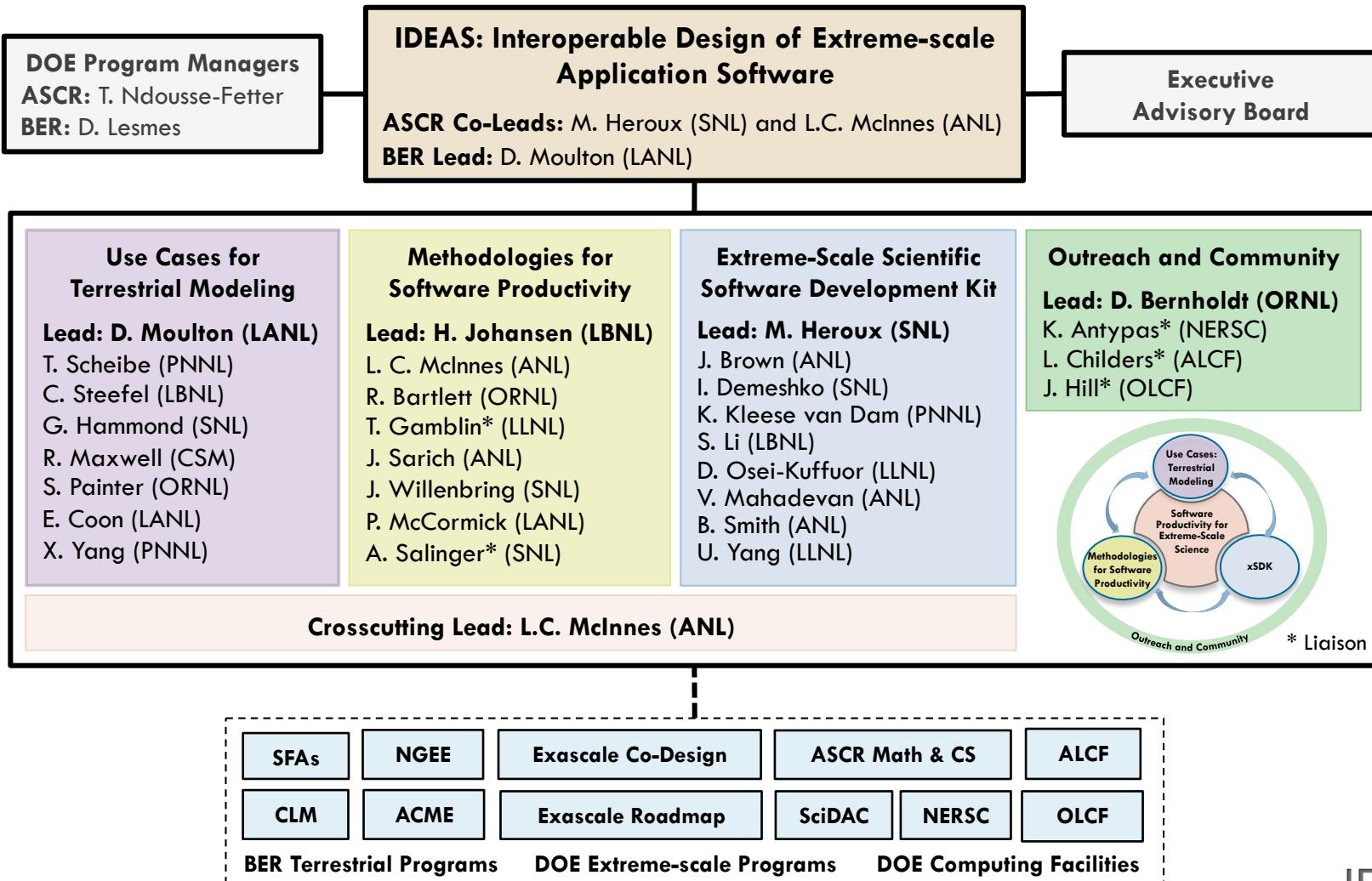
# Outreach and Community

11

- Begin changing the way computational and domain science communities think about software development.
- Training: Bringing *practical* information about techniques and tools to software developers, and advice on tailoring.
  - Targeting BER and broader DOE (via SciDAC, INCITE, ALCC, NNSA, ACTT, etc.) with 2 trainings per year.
- Community development: Online tools to facilitate conversation about productivity issues and solutions.
  - Incrementally build and refine community via outreach to different programs, offices.
- Leverage computing facility liaisons, and team's existing relationships with other programs for Outreach.

# Management Structure Enables Frequent Communication Up & Out

12



# IDEAS Themes

13

- **Use cases:** Drive efforts. Traceability from all efforts.
  - But generalized for future efforts.
- **Methodologies for software productivity:**
  - Metrics: Define for all levels of project. Track progress.
  - Workflows, lifecycles: Document and formalize. ID best practices.
- **xSDK:** frameworks + components + libraries.
  - Build apps by aggregation and composition.
- **Outreach:** Foster communication, adoption, interaction.
- **First of a kind: Focus on software productivity.**

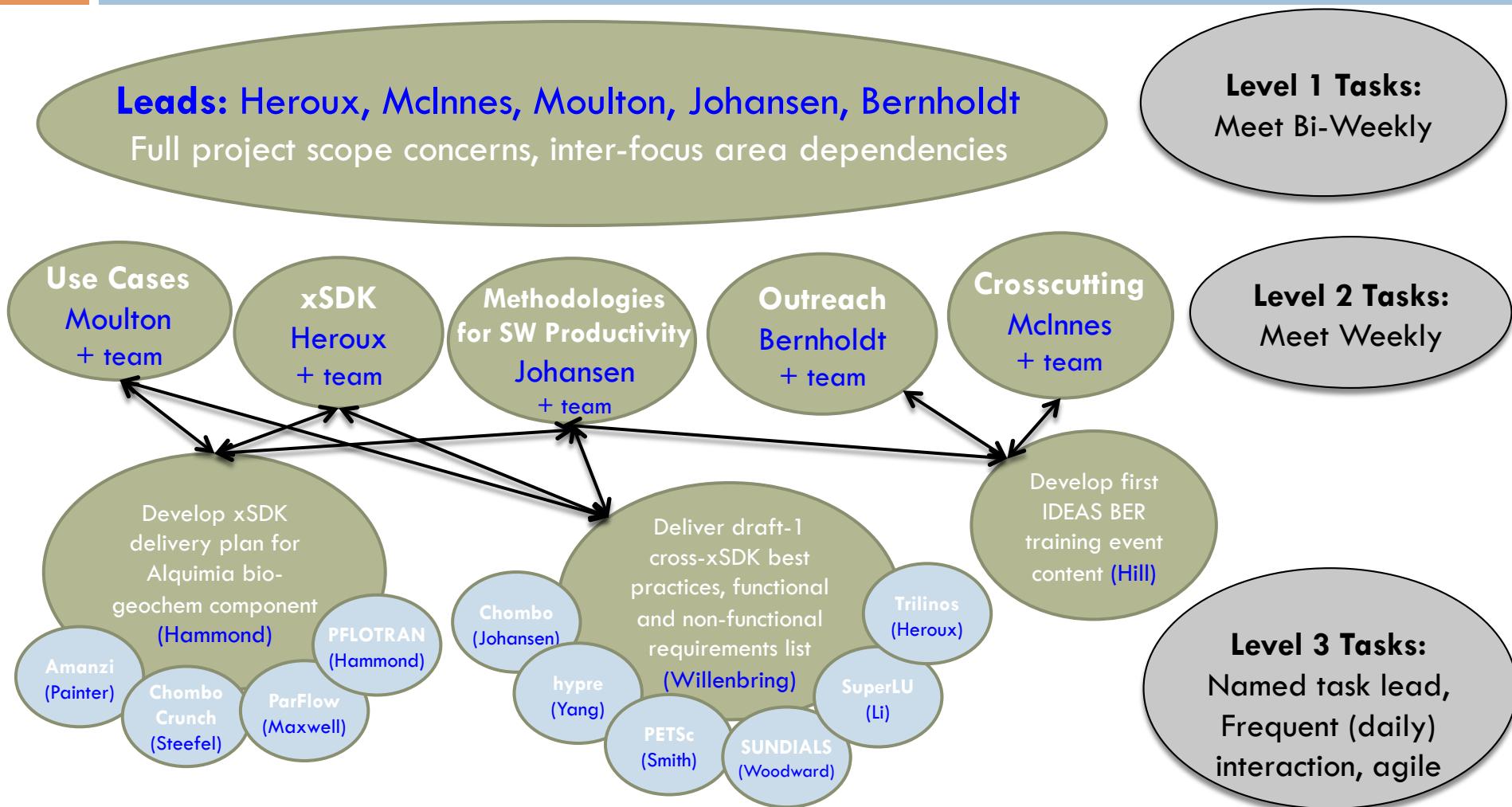
# IDEAS Project Roadmap

14

Focus Area	Level 1	Level 2 Milestones		
		Year 1	Year 2	Year 3
<b>Methodologies for Software Productivity</b>	Establish metric-based SW development workflow strategy templates & lifecycle model.	Assess existing scientific computing SW development workflows and lifecycle models.	Create SW devpt workflow strategy templates; deliver prototype IDEAS SW lifecycle model.	Assess effectiveness and usability of workflow strategy templates and lifecycle model; extend templates and model.
<b>Extreme-Scale Scientific Software Development Kit (xSDK)</b>	Release IDEAS interoperability layers for xSDK libraries and components.	Develop prototype robust IDEAS multi-library installation process and testing infrastructure.	Create Level-1 interoperability layer: linear solvers, including vector and matrix data objects.	Deploy prototype IDEAS xSDK: software, docs, testing.
<b>Use Cases for Terrestrial Modeling</b>	Deploy ecosystem of interoperable components using xSDK for BER applications.	Initiate phase-1 componentization of use-case codes. Co-develop and adopt SWP methodologies.	Expand componentization, enhance interfaces and dynamic coupling.	Deploy xSDK in three canonical development scenarios with defined metrics and assess productivity.
<b>Outreach and Community</b>	Develop and deliver SWP training materials; build a community of productivity-aware software developers.	Develop and deliver SWP trainings for BER and broader DOE audiences. Deploy online community infrastructure.	Evaluate, revise, extend, and deliver trainings and online training materials. Incrementally build online community within DOE.	Evaluate, revise, extend, and deliver trainings. Evaluate and expand online training materials. Incrementally build online community within DOE.

# IDEAS Project Management: Three-tiered Structure

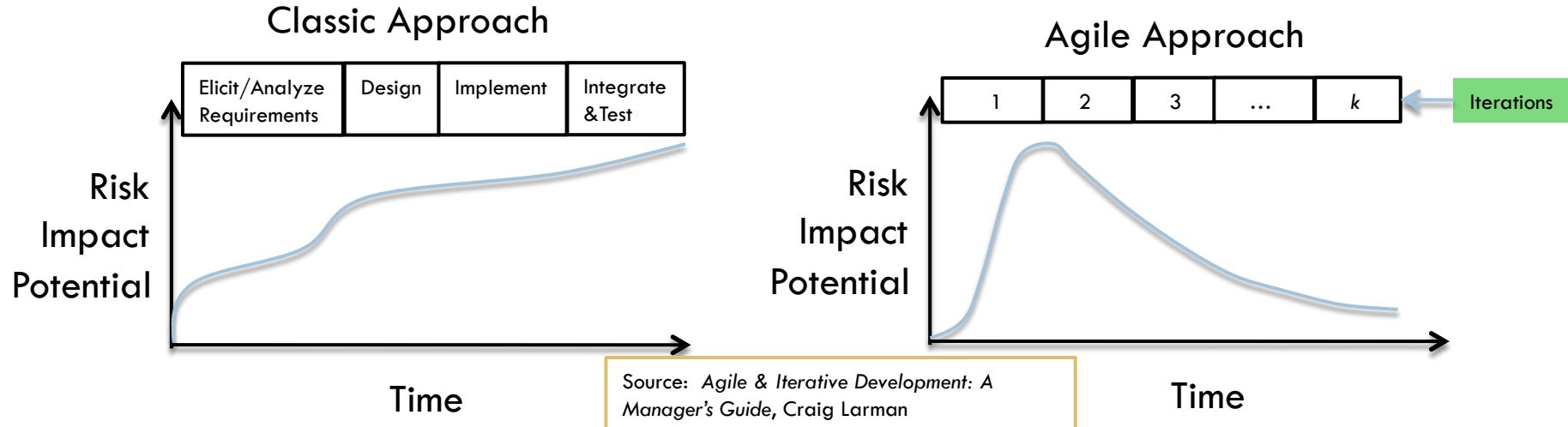
15



# Risk Management: Classic vs Agile

*IDEAS will use agile risk management workflows.*

16

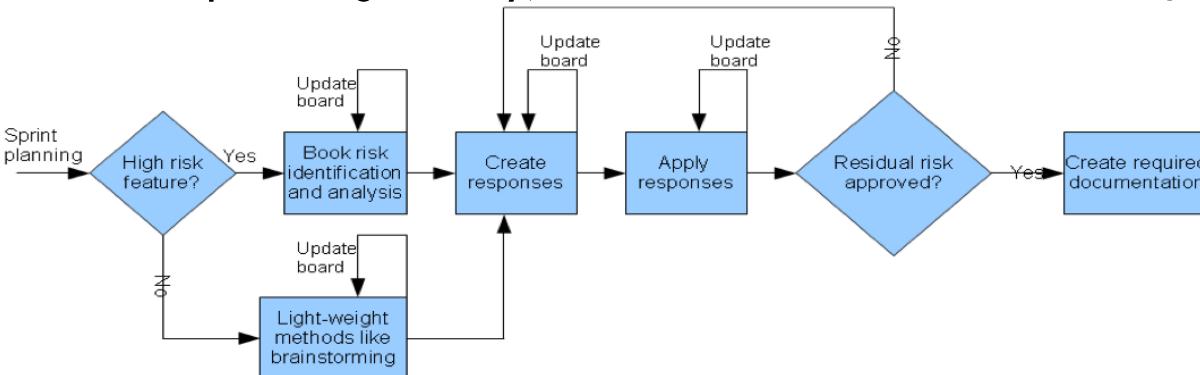


## □ Agile is better:

- Risk managed incrementally.
- Impact high early, lower later.

## □ Risk mgmt, mitigation easier:

- Adapt: Less to refactor.
- Drop: Less invested (lower loss).



**Source:** A Model For Risk Management In Agile Software Development, Ville Ylimannela

# Software Development Methodologies

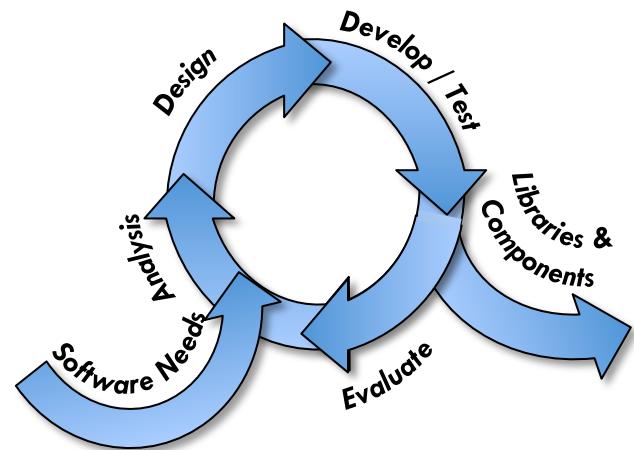
*If I had eight hours to chop down a tree, I would spend six sharpening my axe.*

- Abraham Lincoln

# Common Developer Workflows

18

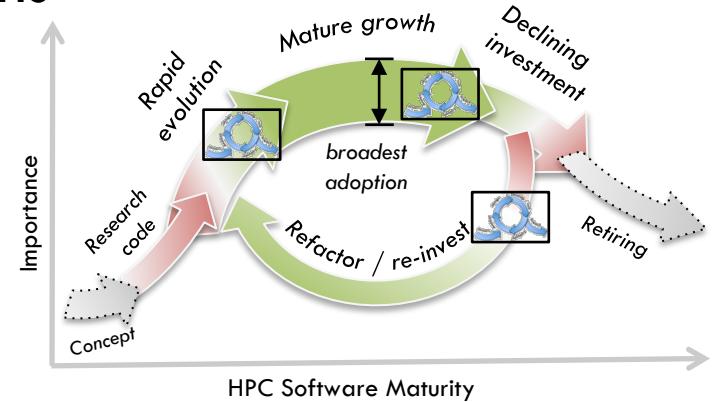
- Define and elaborate key workflows:
  - **Software performance workflows:**
    - Performance analysis, refactoring.
  - **Development workflows:**
    - Clean-slate, augmentation, refactoring [of legacy code].
    - Sprints in an R&D culture.
  - **Repository collaboration workflow models:**
    - Centralized vs feature branch vs. forking, etc.
  - **Documentation workflows:**
    - Domain, user, reference.
  - **Test development & integration workflows:**
    - Test-driven development, test harnesses, auto-regression tests.
- Identify, promote effective tools & best practices:
  - Tool use, enhancement driven by methodology needs.



# Methodologies: Lifecycles

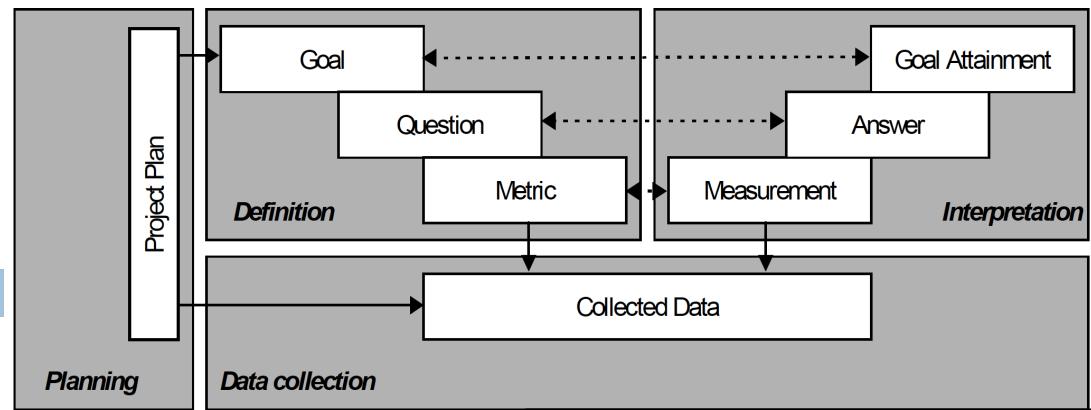
19

- Establish & demonstrate use of effective lifecycles.
  - Phased expectations: Experimental to maintenance.
  - Expectations within each phase:
    - Experimental: Project plan – Funding proposal, artifacts: publications.
    - Maintenance: Domain document, automated regression testing, etc.
  - Promotion criteria, embedded phase regressions, etc.
  - Starting point: TriBITS, collaborations with Human Brain Project, EPFL.
- Training & adoption:
  - Materials, interaction with LCFs.



# Methodologies: SW Productivity Metrics

20



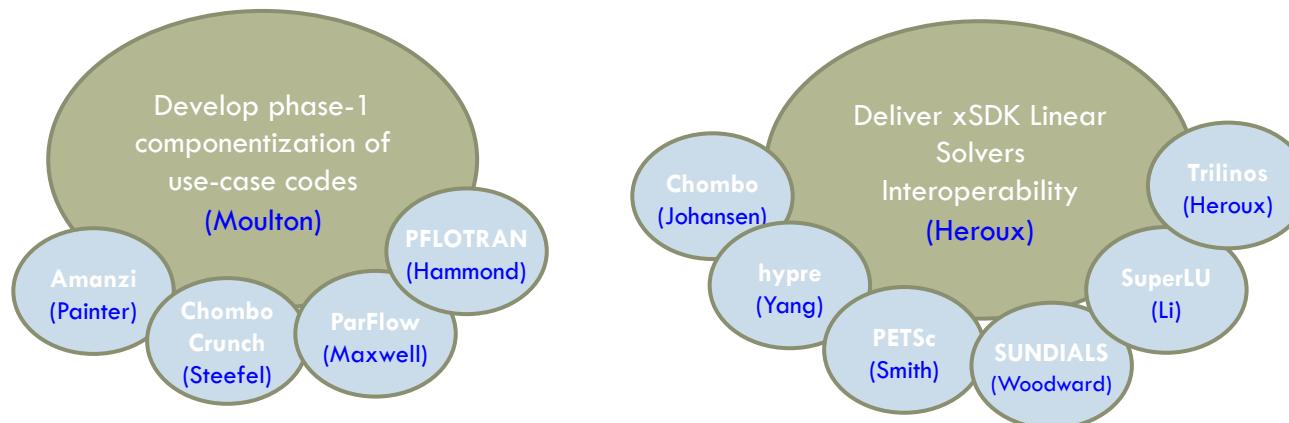
Source: *The GQM Method: A Practical Guide for Quality Improvement of SW Development*, Solingen and Berghout.

- Define processes to define metrics.
  - Starting point: Goals, questions, metrics (GQM).
    - Define goals, ID questions to answer, define progress metrics.
- GQM Example:
  - Goal: xSDK Interoperability.
  - Question: Can IDEAS xSDK components & libs link?
  - Metric: Number of namespace collisions.
- Cultivate effective use of metrics:
  - Use metrics to drive and track use case progress.
- Promote use of metrics via Outreach.
- Note: This is where we *really* need Jeff and Boyana.

# Science Drivers Requiring ASCR Investment: Gaps and Risks in Reduced Funding

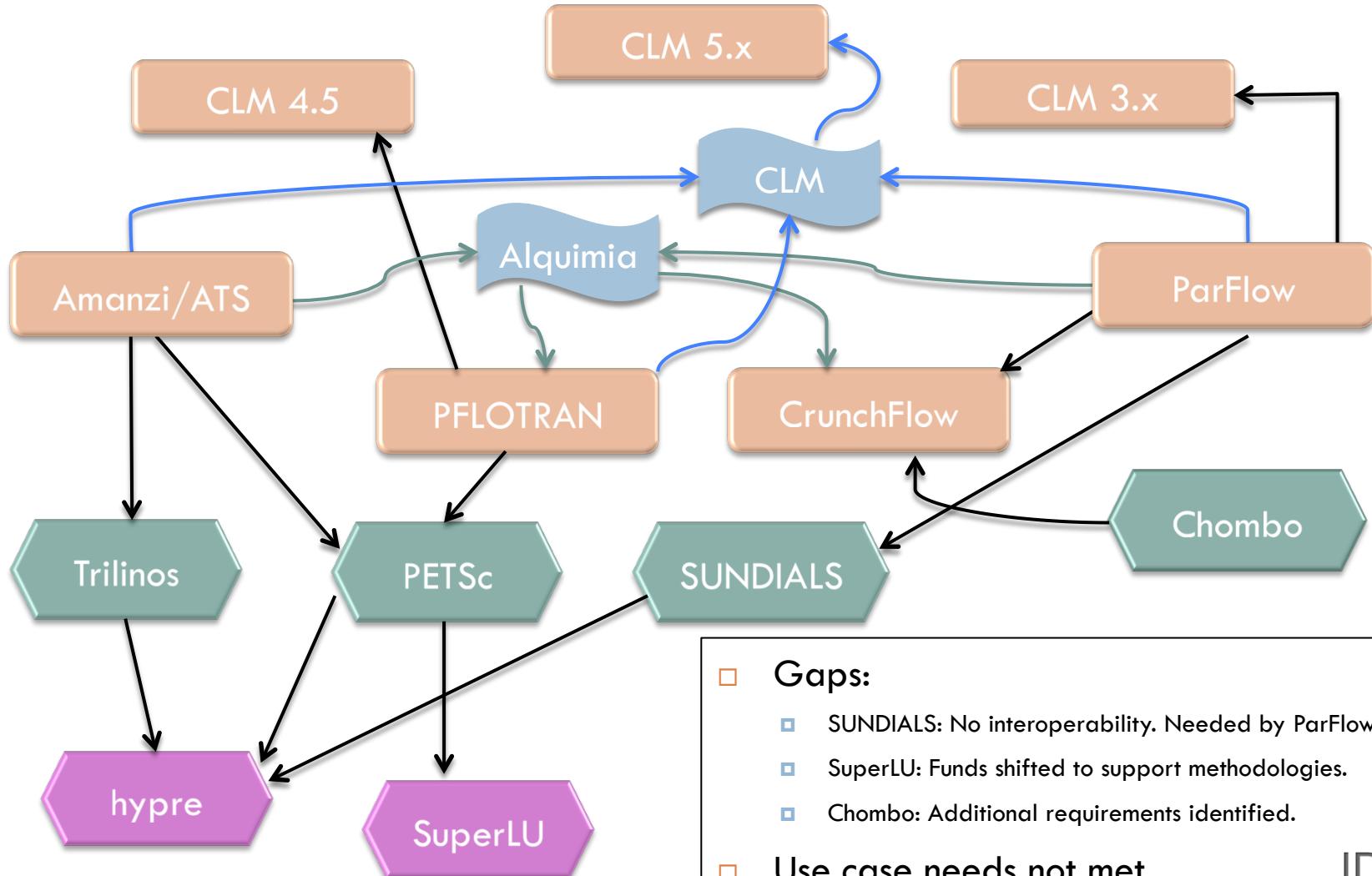
21

- Terrestrial modeling use cases are built on:
  - Amanzi, Chombo-Crunch, ParFlow, PFLOTRAN.
  - These apps require Chombo, hypre, PETSc, SUNDIALS, SuperLU, Trilinos.
- Gaps/risks: Software ecosystem must support science objectives.
  - Need full support for software engineering and productivity processes.
  - Foundations of xSDK: Science drivers require package compatibility and interoperability across components and 6 IDEAS libraries.



# IDEAS Codes and Libraries

22



# Elements of a SciSDK

# Domain Components

24

- Infrequent to moderately reusable functionality.
- Full encapsulation.
- Full testing.
- Sufficient documentation.
- Need to target a “one-to-few” support model.

# Frameworks

25

## □ Framework:

- Infrastructure requiring “buy-in”.

### □ Examples:

- Doxygen.
- Cmake, make.

### □ Benefits:

- Requires only unique content. (Faster & cheaper)
- Promotes uniformity, quality (Better).

### □ Risks:

- Dependence on outside capabilities.
- One-size-fits-all.

### □ Goals:

- Lightweight, incremental adoption, nearby alternative.

# Libraries

26

- Collection of components.
  - Supports *à la carte* approach.
- Advantages:
  - Low risk to adoption:
    - Via API, if alternative exists, can replace with own code.
- Risks:
  - Few if done well.
  - Many if done poorly:
    - On critical path.
    - Poor documentation.
    - Lack of support.
- Barriers:
  - Learning curves.
  - Fears of dependencies.
  - Lack of training.
  - Unawareness.

# Applications

27

- Combinations of toolkits, frameworks and unique code.
  - Requirements and analysis, design:
    - determine usable toolkits, frameworks.
  - Implementation:
    - Composition of components into frameworks.
    - Development of unique code (if any)
    - Configuration of components using the “business rules” of the target app.
  - Example: Albany apps.

# Toolkit of Frameworks

28

- Frameworks have a bad history.
- Not inherently bad, just badly done.
- Toolkits of lightweight frameworks.
  - More than one solution to ...
    - Make, Cmake.
  - Easy to adopt and see some value.
    - Doxygen, make.

# 40-Day State of the Project Report

29

- IDEAS all-hands kickoff meeting: Jan 20 – 22, 2015, LBNL.
- IDEAS subgroup sessions feed into all-hands meeting:
  - **Use Case 1:** modeling & code capabilities, Sept 19, CSM.
  - **xSDK subgroup:** Sept 30, ANL:
    - Developers of IDEAS libraries (Chombo, hypre, PETSc, SUNDIALS, SuperLU, Trilinos) + Facilities liaisons (ALCF, NERSC, OLCF).
    - Identified/prioritized activities to qualitatively improve compatibility and interoperability across xSDK libraries, and to provide better documentation, testing, and infrastructure to make these features sustainable and extensible.
  - **Modeling breakout session:** Oct 23-24, LBNL-SFA annual retreat.
  - **Methodologies subgroup:** fully virtual meeting, Nov 13.
  - **Use-case subgroup:** working sessions + meeting at AGU, Dec.
- Project communication infrastructure:
  - Established initial IDEAS project website.
  - Exploring wiki and repo options for internal project collaboration, (e.g., confluence), technologies for remote meetings.

Brainstorming from discussion on Sept 30, 2014

- Common cmake/configure API.
- Template app using all IDEAS libraries.
- Performance portability.
- Understand current development workflows.
- Testing infrastructure.
- Pursue use case input (BER partners).
- Determine interoperability points for libraries.
- Next-generation libraries.
- Metrics of success.
- Explore opportunities for facilities collaborations.
- How to mentor junior people for software productivity, xSDK use.
- Best practices for IDEAS xSDK.
- Determine collaboration tools to use for IDEAS project.

# IDEAS Summary

30

- **Enabling production of high-quality science results, rapidly and efficiently.**
  - Specifically: Multiscale terrestrial ecosystem science.
  - Broadly: Extreme-scale scientific applications across DOE.
- **Delivering first-of-a-kind extreme-scale scientific software ecosystem:**
  - Extreme-Scale Scientific Software Development Kit (xSDK):
    - Components & libraries.
    - Frameworks & tools.
    - Methodologies & practices.
  - Apps built by composition:
    - Leverage xSDK.
    - Provide only what is unique.
    - High-quality apps from high-quality SDK.
    - Rapid, efficient development.
- **Exploiting new exascale opportunities:**
  - Raw computing capabilities of advanced architectures.
  - Coupled multiphysics and multiscale.

IDEAS provides first direct funding for:

- **Productivity focus.**
- **Sustainable interoperability across DOE libs.**
- **Research in SW Quality.**