Better (Small) Software Teams

Michael A. Heroux

Scientist in Residence, CSB|SJU

Senior Scientist, Sandia National Laboratories

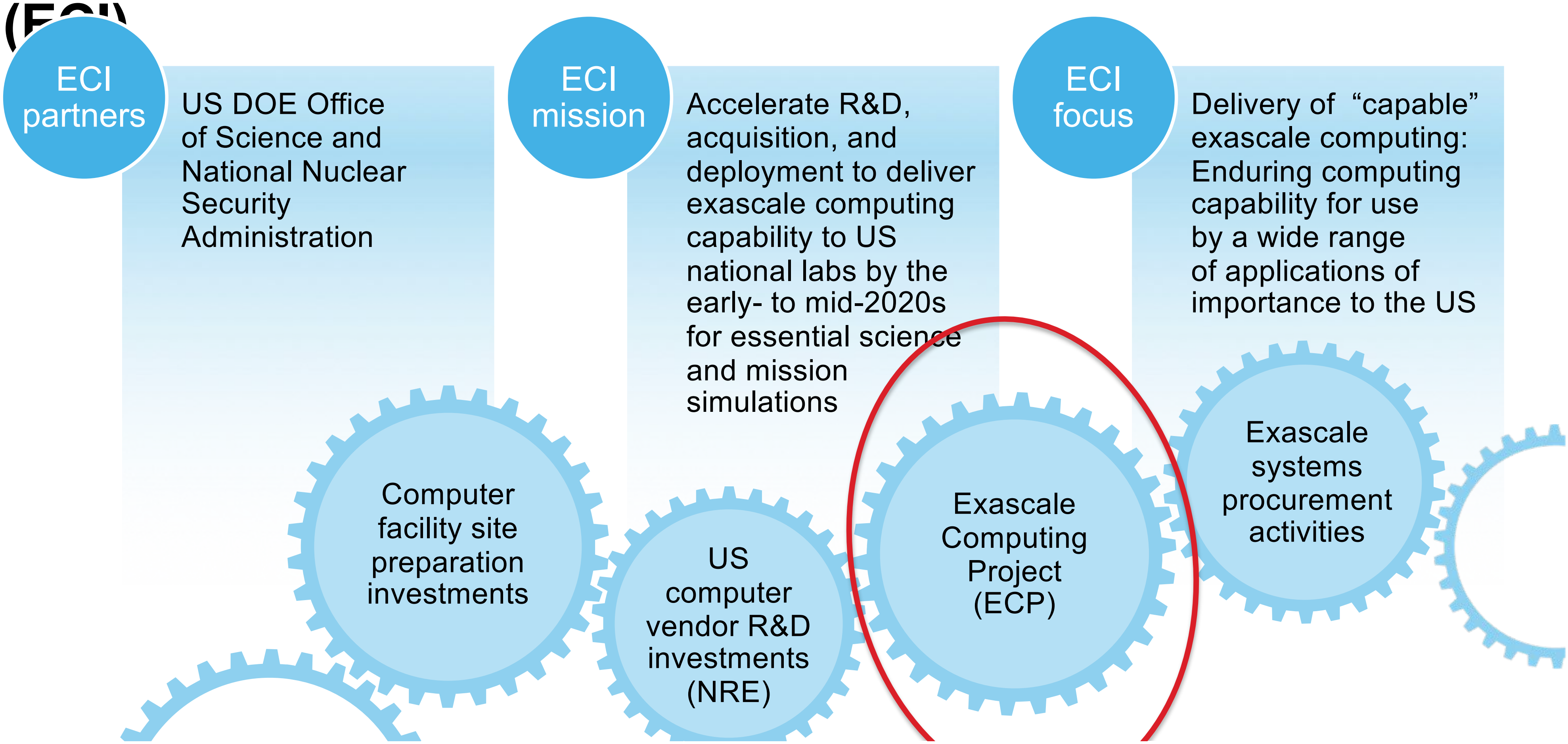
Director of Software Technology, US DOE Exascale Computing Project

EXASCALE COMPUTING PROJECT

Outline

- The US Exascale Computing Initiative
- Small Team Models, Challenges
- Agile workflow management for small teams
  - Intro to terminology and approaches
  - Overview of Kanban
  - Free tools: Trello, GitHub.
- Hands-on example of project management using GitHub

The US exascale strategy includes four major elements within the Exascale Computing Initiative (ECI)



ECI partners

US DOE Office of Science and National Nuclear Security Administration

ECI mission

Accelerate R&D, acquisition, and deployment to deliver exascale computing capability to US national labs by the early- to mid-2020s for essential science and mission simulations

ECI focus

Delivery of "capable" exascale computing: Enduring computing capability for use by a wide range of applications of importance to the US

Computer facility site preparation investments

US computer vendor R&D investments (NRE)

Exascale Computing Project (ECP)

Exascale systems procurement activities

ECP is a critical component of the broader US ECI strategy

ECP depends on other major ECI elements for success

- Deploying exascale systems quickly enough to impact schedule-sensitive mission problems
- Maintaining and advancing the “HPC ecosystem” after ECP
- Developing US industry and academia partnerships to ensure that the benefits
  of advanced computing have broad and enduring impacts

Enabling future US revolutions in technology development, scientific discovery, energy and economic security, and healthcare

ECP mission

Deliver exascale-ready applications and solutions that address currently intractable problems of strategic importance and national interest

Create and deploy an expanded and vertically integrated software stack on DOE HPC exascale and pre-exascale systems, defining the enduring US exascale ecosystem

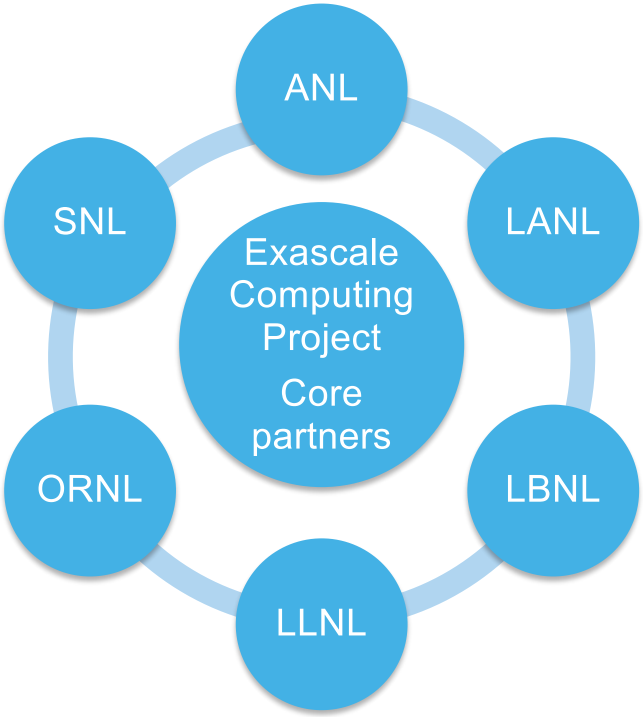
Leverage US HPC vendor R&D activities and products into DOE HPC exascale systems

ECP vision

Accelerating innovation with exascale simulation and data science solutions that enhance US economic competitiveness, improve our quality of life, and strengthen our national security

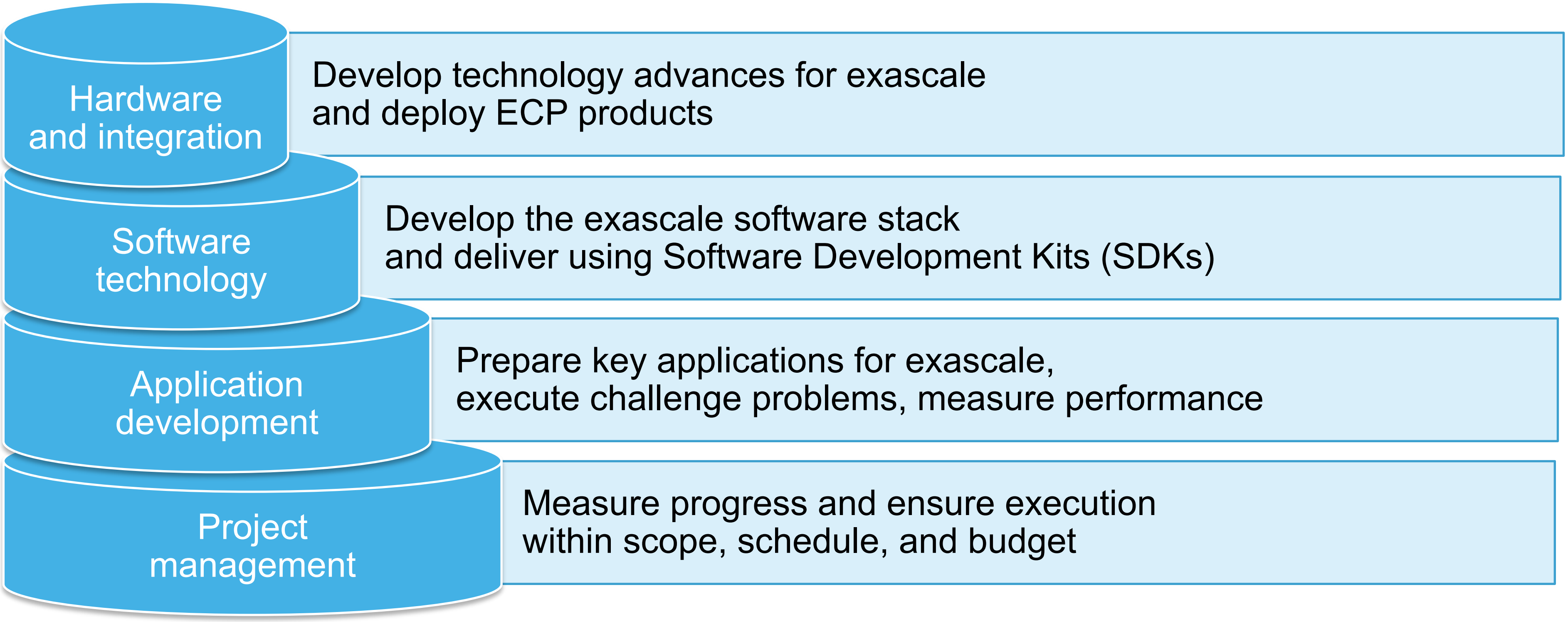
ECP is a Collaboration Among Six US DOE National Laboratories

- The ECP draws from the Nation’s 6 premier computing national laboratories
- A Memorandum of Agreement for the ECP was signed by each Laboratory Director defining roles and responsibilities
- Funding comes from two sources: DOE Office of Science and NNSA Advanced Simulation and Computing (ASC) program



ECP is a large, complex project

Spanning the nation and structured for success



Hardware and integration

Develop technology advances for exascale and deploy ECP products

Software technology

Develop the exascale software stack and deliver using Software Development Kits (SDKs)

Application development

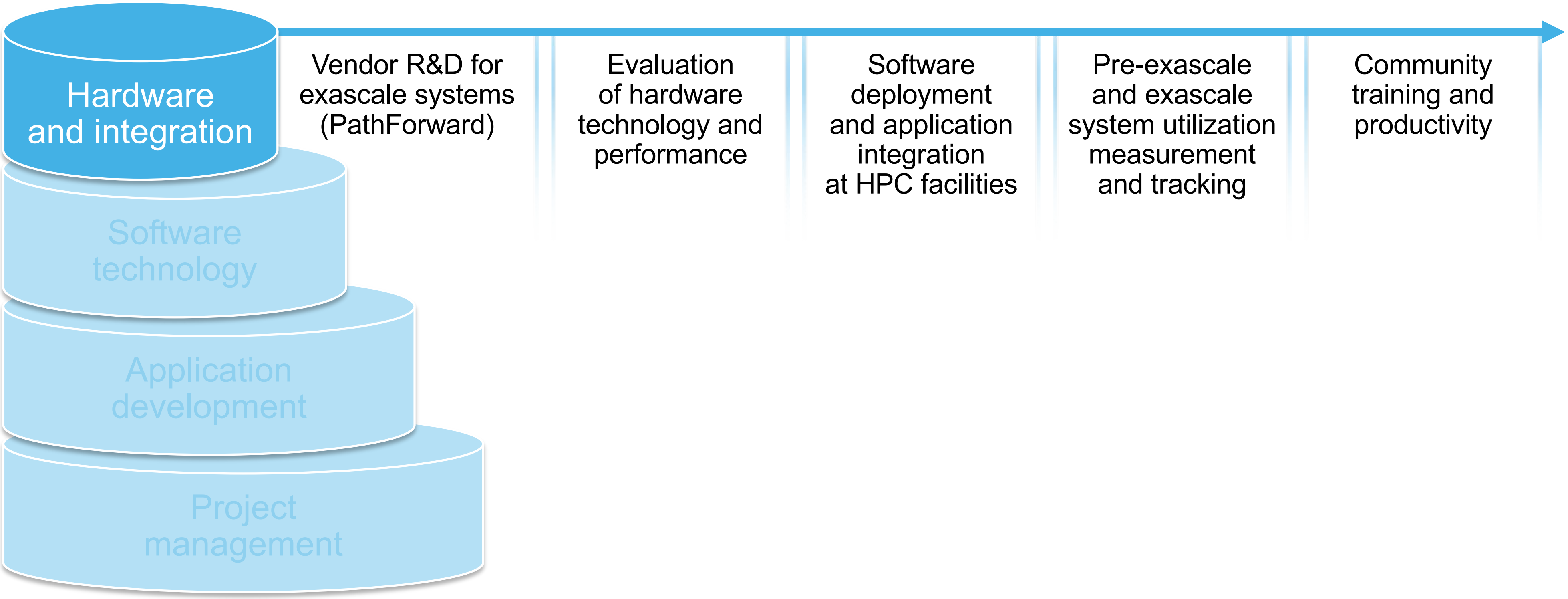
Prepare key applications for exascale, execute challenge problems, measure performance

Project management

Measure progress and ensure execution within scope, schedule, and budget

Hardware and integration (HI)

Develop technology advances for exascale and deploy ECP products



Hardware and integration

Software technology

Application development

Project management

Vendor R&D for exascale systems (PathForward)

Evaluation of hardware technology and performance

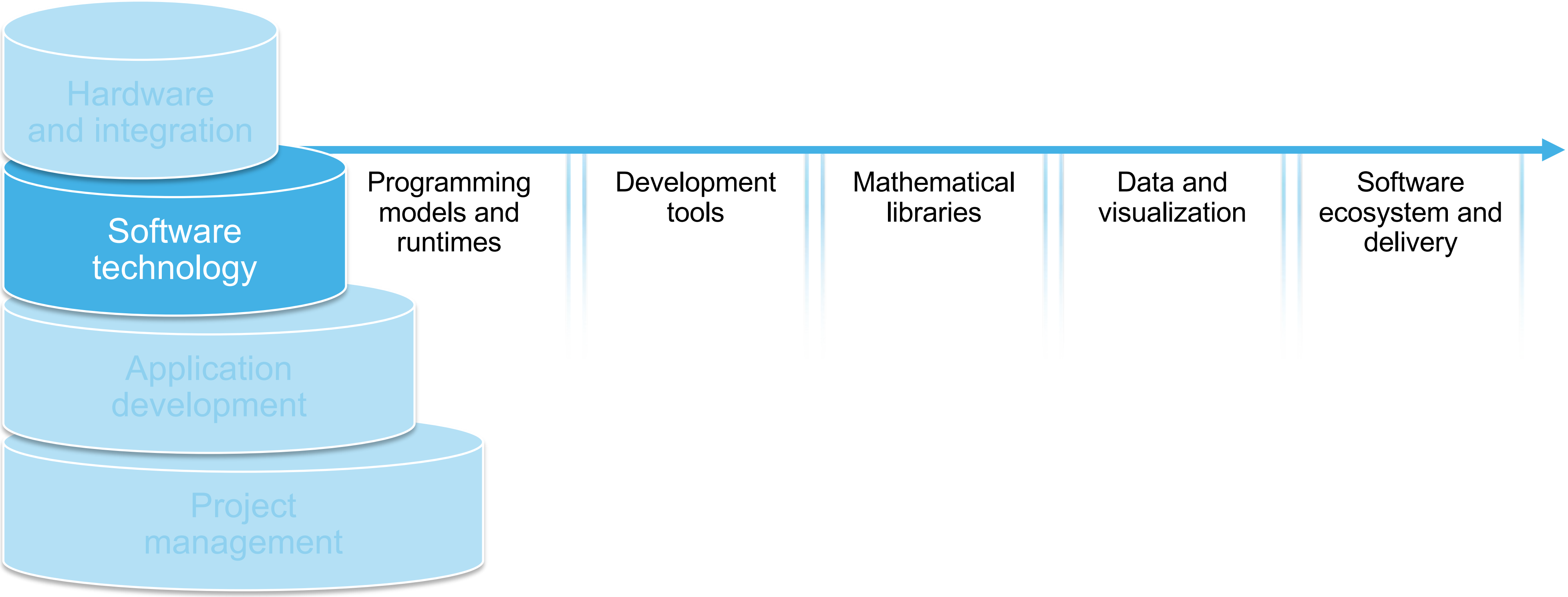
Software deployment and application integration at HPC facilities

Pre-exascale and exascale system utilization measurement and tracking

Community training and productivity

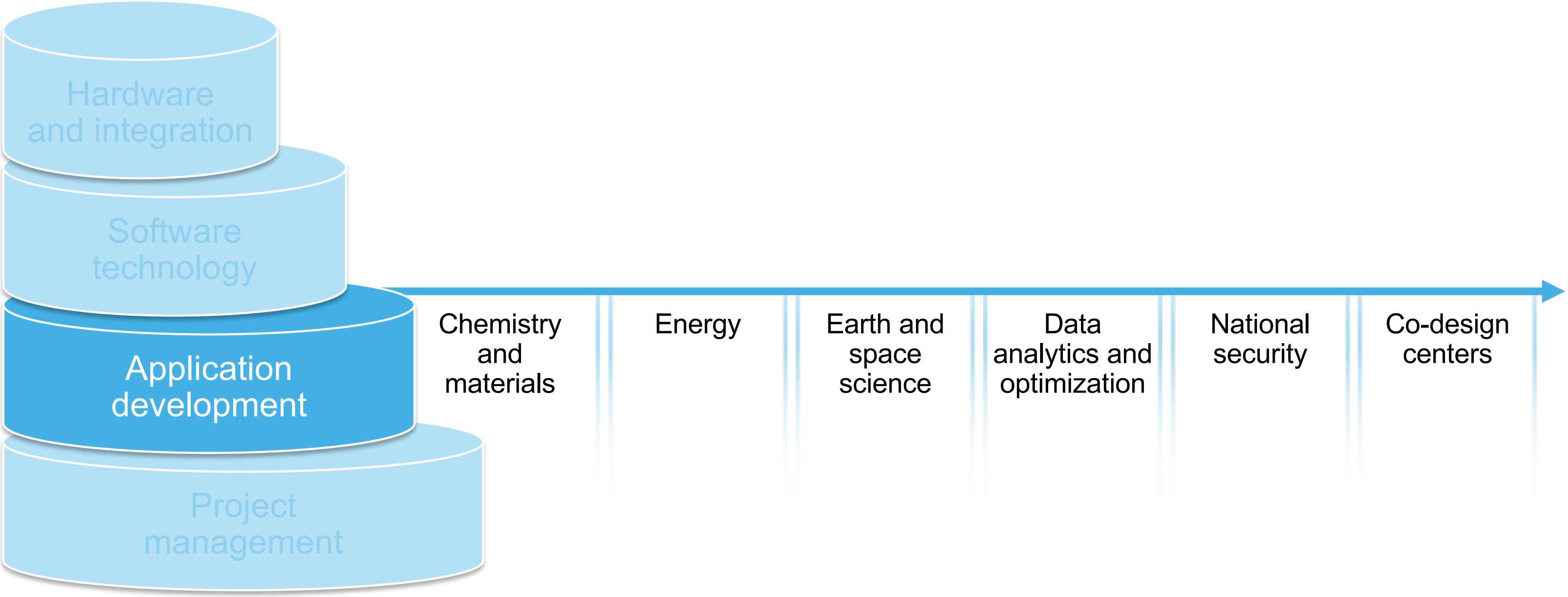
Software technology (ST)

Develop the exascale software stack and deliver using Software Development Kits (SDKs)



Application development (AD)

Prepare key applications for exascale, execute challenge problems, measure performance



Hardware and integration

Software technology

Application development

Project management

Chemistry and materials

Energy

Earth and space science

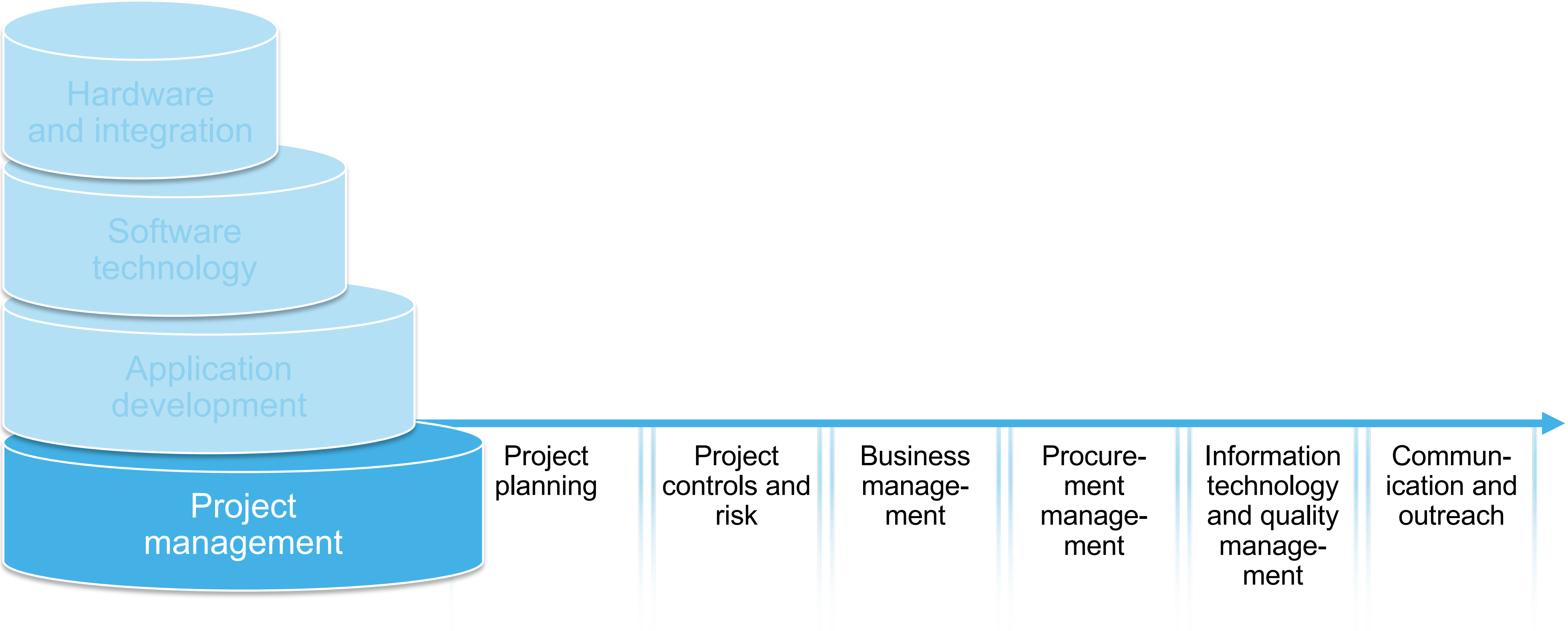
Data analytics and optimization

National security

Co-design centers

Project management (PM)

Measure progress and ensure execution within scope, schedule, and budget



Relevant US Pre-Exascale and Exascale Systems for ECP

Pre-Exascale Systems

2013

Argonne
IBM BG/Q

ORNL
Cray/NVidia K20

LLNL
IBM BG/Q

2016

Argonne
Intel/Cray KNL

LBNL
Cray/Intel Xeon/KNL

LANL/SNL
Cray/Intel Xeon/KNL

2018

ORNL
IBM/NVidia
P9/Volta

LLNL
IBM/NVidia
P9/Volta

2020

NERSC-9

LBNL
TBD

Crossroads

LANL/SNL
TBD

Exascale Systems

2021-2023

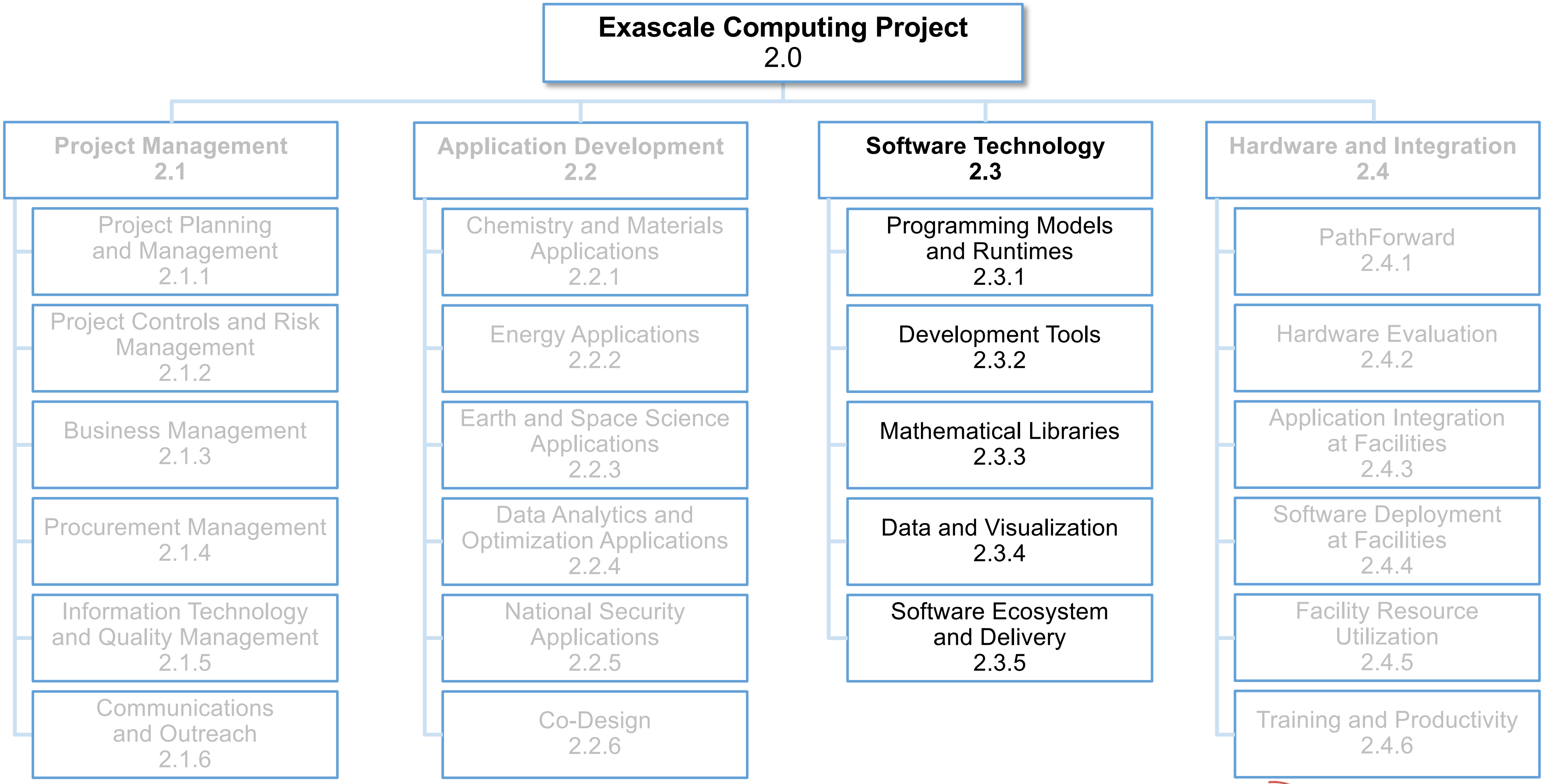
Argonne
Intel/Cray TBD

Frontier

ORNL
TBD

El Capitan

LLNL
TBD



Exascale Computing Project
2.0

Project Management
2.1

Project Planning and Management
2.1.1

Project Controls and Risk Management
2.1.2

Business Management
2.1.3

Procurement Management
2.1.4

Information Technology and Quality Management
2.1.5

Communications and Outreach
2.1.6

Application Development
2.2

Chemistry and Materials Applications
2.2.1

Energy Applications
2.2.2

Earth and Space Science Applications
2.2.3

Data Analytics and Optimization Applications
2.2.4

National Security Applications
2.2.5

Co-Design
2.2.6

Software Technology
2.3

Programming Models and Runtimes
2.3.1

Development Tools
2.3.2

Mathematical Libraries
2.3.3

Data and Visualization
2.3.4

Software Ecosystem and Delivery
2.3.5

Hardware and Integration
2.4

PathForward
2.4.1

Hardware Evaluation
2.4.2

Application Integration at Facilities
2.4.3

Software Deployment at Facilities
2.4.4

Facility Resource Utilization
2.4.5

Training and Productivity
2.4.6

ECP Software Technology Leadership Team

Mike Heroux, Software Technology Director
Mike has been involved in scientific software R&D for 30 years. His first 10 were at Cray in the LIBSCI and scalable apps groups. At Sandia he started the Trilinos and Mantevo projects, is author of the HPCG benchmark for TOP500, and leads productivity and sustainability efforts for DOE.

Jonathan Carter, Software Technology Deputy Director
Jonathan has been involved in the support and development of HPC applications for chemistry, the procurement of HPC systems, and the evaluation of novel computing hardware for over 25 years. He currently a senior manager in Computing Sciences at Berkeley Lab.

Rajeev Thakur, Programming Models and Runtimes (2.3.1)
Rajeev is a senior computer scientist at ANL and most recently led the ECP Software Technology focus area. His research interests are in parallel programming models, runtime systems, communication libraries, and scalable parallel I/O. He has been involved in the development of open source software for large-scale HPC systems for over 20 years.

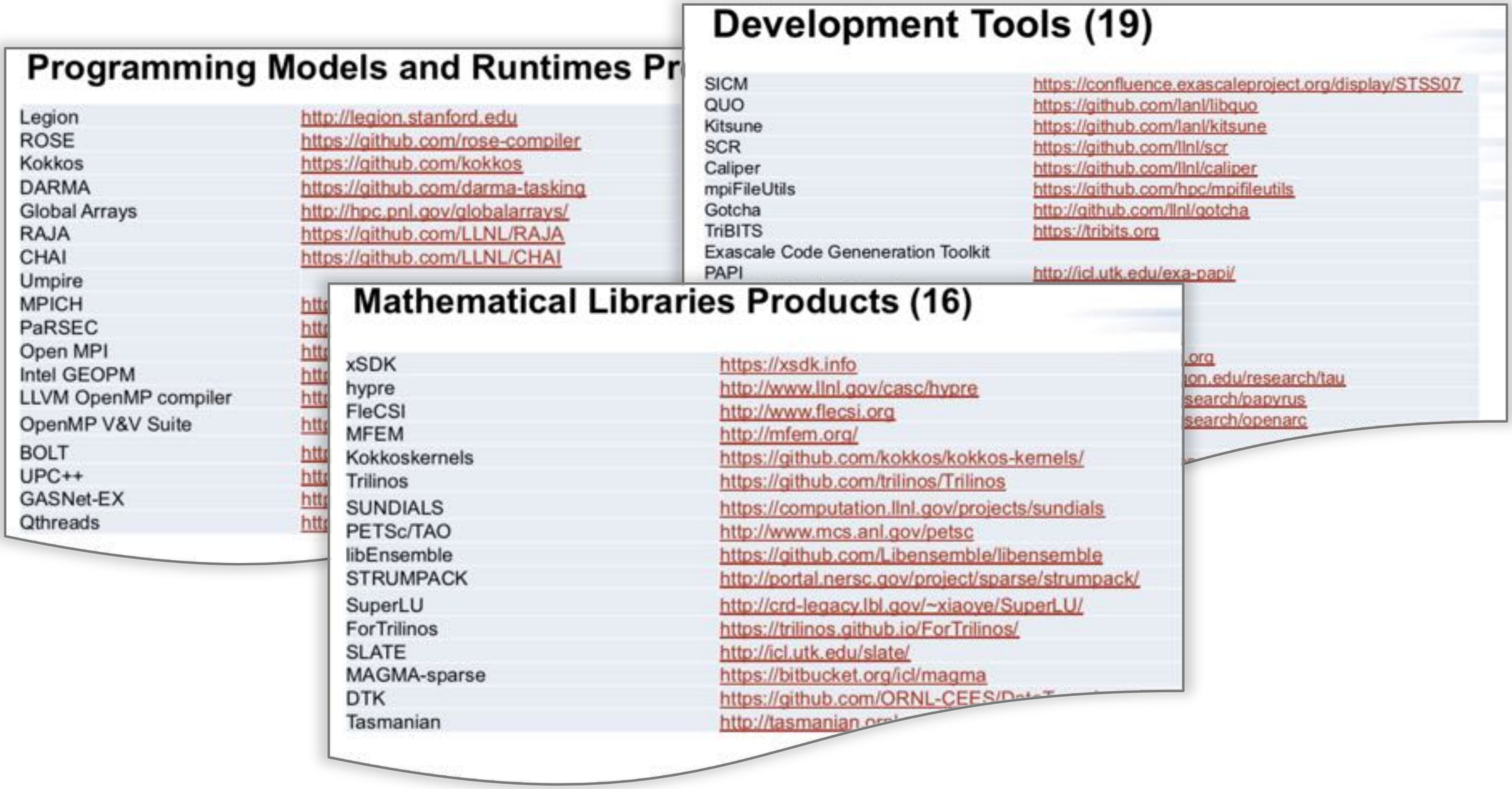
Jeff Vetter, Development Tools (2.3.2)
Jeff is a computer scientist at ORNL, where he leads the Future Technologies Group. He has been involved in research and development of architectures and software for emerging technologies, such as heterogeneous computing and nonvolatile memory, for HPC for over 15 years.

Lois Curfman McInnes, Math Libraries (2.3.3)
Lois is a senior computational scientist in the Mathematics and Computer Science Division of ANL. She has over 20 years of experience in high-performance numerical software, including development of PETSc and leadership of multi-institutional work toward sustainable scientific software ecosystems.

Jim Ahrens, Data and Visualization (2.3.4)
Jim is a senior research scientist at the Los Alamos National Laboratory (LANL) and an expert in data science at scale. He started and actively contributes to many open-source data science packages including ParaView and Cinema.

Rob Neely, Software Ecosystem and Delivery (2.3.5)
Rob has several leadership roles at LLNL spanning applications, CS research, platforms, and vendor interactions. He is an Associate Division Leader in the Center for Applied Scientific Computing (CASC), chair of the Weapons Simulation and Computing Research Council, and the lead for the Sierra Center of Excellence.

ECP ST efforts currently contribute to 89 unique products



Programming Models and Runtimes Pr

Legion	http://legion.stanford.edu
ROSE	https://github.com/rose-compiler
Kokkos	https://github.com/kokkos
DARMA	https://github.com/darma-tasking
Global Arrays	http://hpc.pnl.gov/globalarrays/
RAJA	https://github.com/LLNL/RAJA
CHAI	https://github.com/LLNL/CHAI
Umpire	
MPICH	http
PaRSEC	http
Open MPI	http
Intel GEOPM	http
LLVM OpenMP compiler	http
OpenMP V&V Suite	http
BOLT	http
UPC++	http
GASNet-EX	http
Qthreads	http

Development Tools (19)

SICM	https://confluence.exascaleproject.org/display/STSS07
QUO	https://github.com/lanl/libquo
Kitsune	https://github.com/lanl/kitsune
SCR	https://github.com/llnl/scr
Caliper	https://github.com/llnl/caliper
mpiFileUtils	https://github.com/hpc/mpifileutils
Gotcha	http://github.com/llnl/gotcha
TriBITS	https://tribits.org
Exascale Code Geneneration Toolkit	
PAPI	http://icl.utk.edu/exa-papi/
	.org
	on.edu/research/tau
	search/papyrus
	search/openarc

Mathematical Libraries Products (16)

xSDK	https://xsdk.info
hypre	http://www.llnl.gov/casc/hypre
FleCSI	http://www.flecsi.org
MFEM	http://mfem.org/
Kokkoskernels	https://github.com/kokkos/kokkos-kernels/
Trilinos	https://github.com/trilinos/Trilinos
SUNDIALS	https://computation.llnl.gov/projects/sundials
PETSc/TAO	http://www.mcs.anl.gov/petsc
libEnsemble	https://github.com/Libensemble/libensemble
STRUMPACK	http://portal.nersc.gov/project/sparse/strumpack/
SuperLU	http://crd-legacy.lbl.gov/~xiaoye/SuperLU/
ForTrilinos	https://trilinos.github.io/ForTrilinos/
SLATE	http://icl.utk.edu/slate/
MAGMA-sparse	https://bitbucket.org/icl/magma
DTK	https://github.com/ORNL-CEES/
Tasmanian	http://tasmanian.or

Small Teams

Ideas for managing transitions and steady work.

Small team interaction model

- Team composition:
  - Senior staff, faculty:
    - Stable presence, in charge of science questions, experiments.
    - Know the conceptual models well.
    - Spend less time writing code, fuzzy on details.
  - Junior staff, students:
    - Transient, dual focus (science results, next position).
    - Staged experience: New, experienced, departing.
    - Learning conceptual models.
    - Write most code, know details.

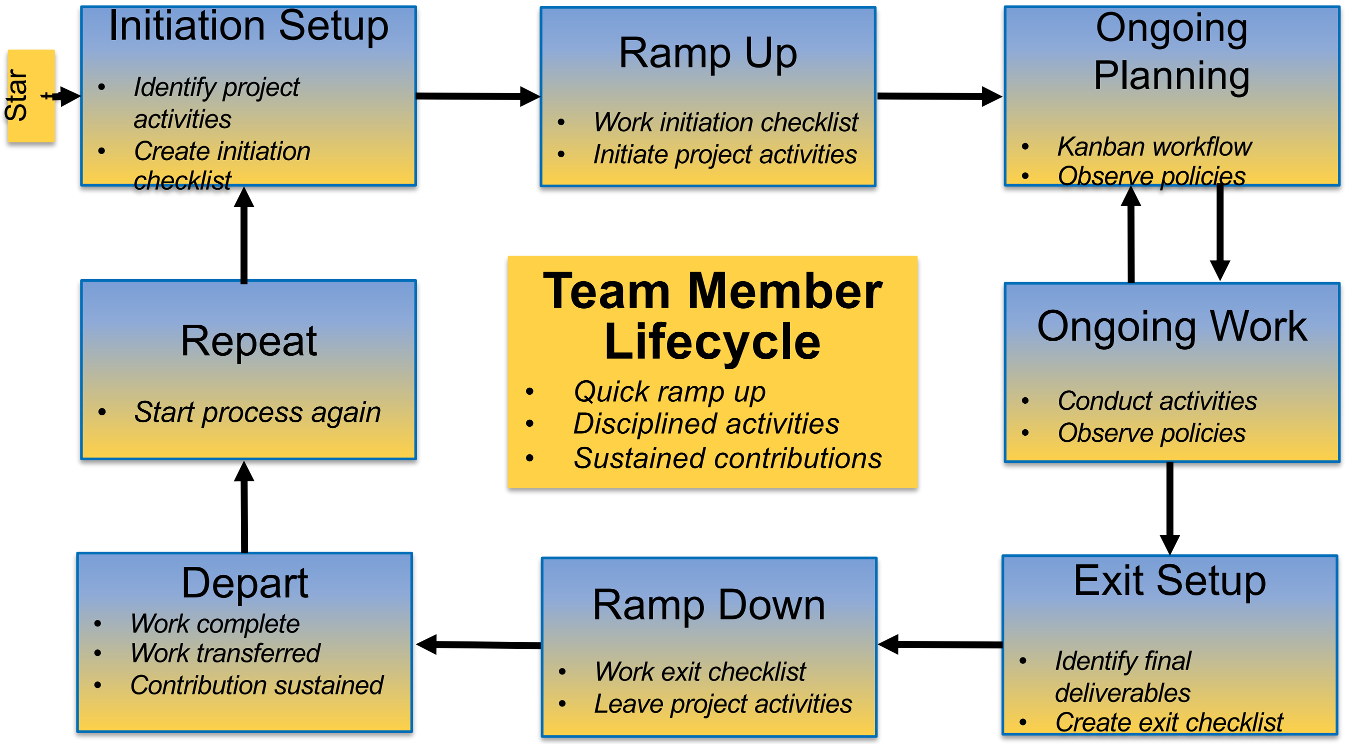
Large team challenges

- Composed of small teams (and all the challenges).
- Additional interaction challenges.
- Policies, regularly cultural exchanges important.

Small team challenges

- Ramping up new junior members:
  - Background.
  - Conceptual models.
  - Software practices, processes, tools.
- Preparing for departure of experienced juniors.
  - Doing today those things needed for retaining work value.
  - Managing dual focus.

Research Team Member Lifecycle



Start

Initiation Setup

- *Identify project activities*
- *Create initiation checklist*

Ramp Up

- *Work initiation checklist*
- *Initiate project activities*

Ongoing Planning

- *Kanban workflow*
- *Observe policies*

Ongoing Work

- *Conduct activities*
- *Observe policies*

Exit Setup

- *Identify final deliverables*
- *Create exit checklist*

Ramp Down

- *Work exit checklist*
- *Leave project activities*

Depart

- *Work complete*
- *Work transferred*
- *Contribution sustained*

Repeat

- *Start process again*

Team Member Lifecycle

- *Quick ramp up*
- *Disciplined activities*
- *Sustained contributions*

Checklists & Policies

Team Member Phase		
New Team Member	Steady Contributor	Departing Member
Checklist	Policies	Checklist

- New, departing team member checklists:
  - Example: Trilinos New Developer Checklist.
  - https://software.sandia.gov/trilinos/developer/sqp/checklists/index.html
- Steady state: Policy-driven.
  - Example: xSDK Community policies.
  - https://xsdk.info/policies/

Your checklists & policies?

- Checklist: New team member?
- Policies: Ongoing work?
- Checklist: Before someone departs?

Collaborative Work Management

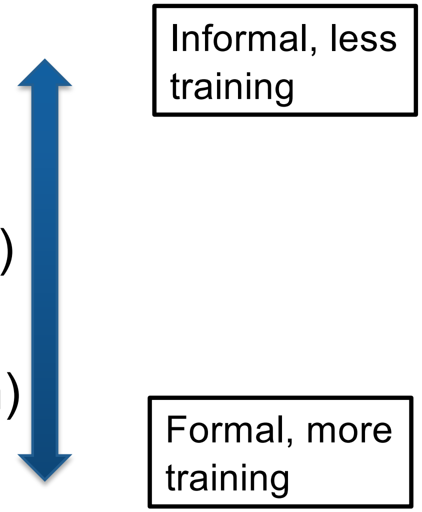
Managing with Kanban

IDEAS productivity

Managing issues: Fundamental software process

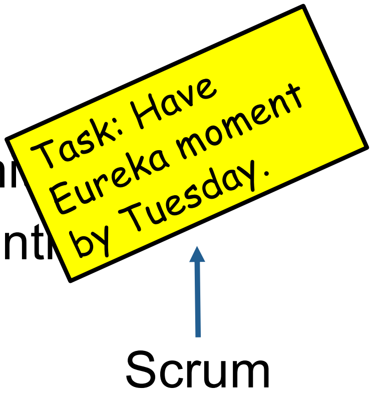
Continual improvement

- Issue: Bug report, feature request
- Approaches:
  - Short-term memory, office notepad
  - ToDo.txt on computer desktop (1 person)
  - Issues.txt in repository root (small co-located team)
  - …
  - Web-based tool + Kanban (distributed, larger team)
  - Web-based tool + Scrum (full-time dev team)



Kanban principles

- Limit number of “In Progress” tasks
- Productivity improvement:
  - Optimize “flexibility vs swap overhead” balance. No overcommi
  - Productivity weakness exposed as bottleneck. Team must identi and fix the bottleneck.
  - Effective in R&D setting. Avoids a deadline-based approach. Deadlines are dealt with in a different way.
- Provides a board for viewing and managing issues
- *Can be applied to any existing software project immediately!*



Task: Have Eureka moment by Tuesday.

Scrum

Basic Kanban

Backlog	Ready	In Progress	Done
• Any task idea • Trim occasionally • Source for other columns	• Task + description of how to do it. • Could be pulled when slot opens. • Typically comes from backlog.	• Task you are working on *right now.* • **The only kanban rule: Can have only so many "In Progress" tasks.** • Limit is based on experience, calibration. • **Key: Work is *pulled*. You are in charge!**	• Completed tasks. • Record of your life activities. • Rate of completion is your "velocity".

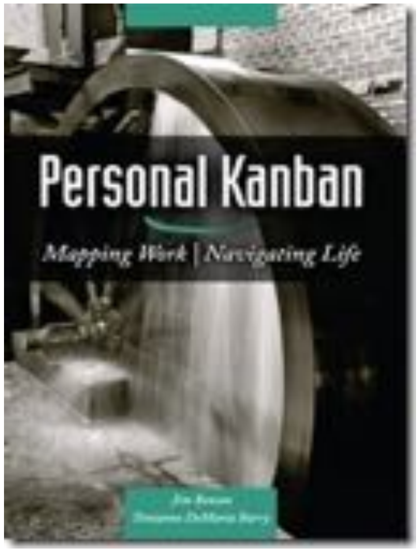
Notes:

- Ready column is not strictly required, sometimes called "Selected for development".
- Other common column: In Review
- Can be creative with columns:
  - Waiting on Advisor Confirmation.
  - Tasks I won't do.

Personal Kanban

- Personal Kanban: Kanban applied to one person.
  - Apply Kanban principles to your life.
  - Fully adaptable.

- Personal Kanban: Commercial book/website.
  - Useful, but not necessary.



http://www.personalkanban.com

Kanban tools

- Wall, whiteboard, blackboard: Basic approach.
- Software, cloud-based:
  - Trello, JIRA, GitHub Issues.
  - Many more.
- I use Trello (browser, iPhone, iPad).
  - Can add, view, update, anytime, anywhere.

Big question: How many tasks?

- Personal question.
- Approach: Start with 2 or 3. See how it goes.
- Use a freeway traffic analogy:
  - Does traffic flow best when fully packed? No.
  - Same thing with your effectiveness.
- Spend time consulting board regularly.
  - Brings focus.
  - Enables reflection, retrospection.
  - Use slack time effectively.
  - When you get out of the habit, start up again.

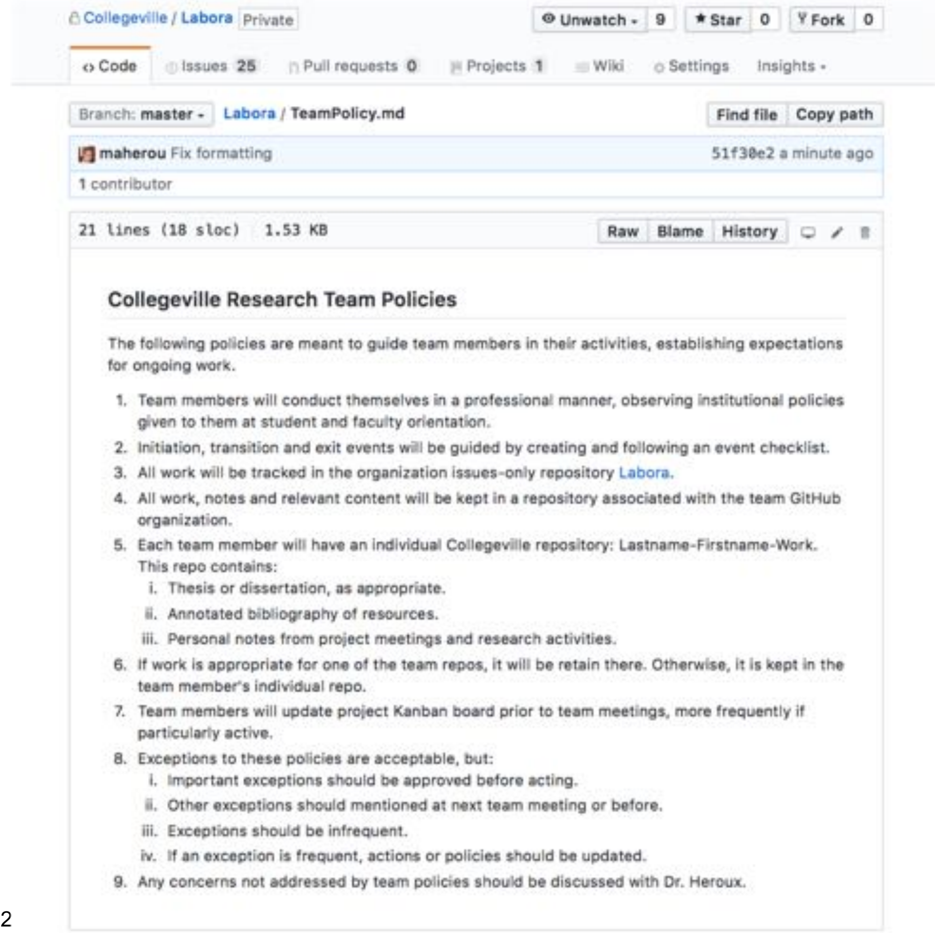
Importance of “In Progress” concept for you

- Junior community members:
  - Less control over task.
  - Given by supervisor.
- In Progress column: Protects you.
  - If asked to take on another task, respond:
    - Is this important enough to become less efficient?
    - Sometimes it is.

Key Team Management Elements

- **Checklists:**
  - Initiation, Transition, Exit
- **Policies:**
  - How team conducts its work
- **Issue tracking system:**
  - All work tracked, visible to team
  - Milestones: Aggregate related issues.
  - Kanban board
  - Regular meetings, updates

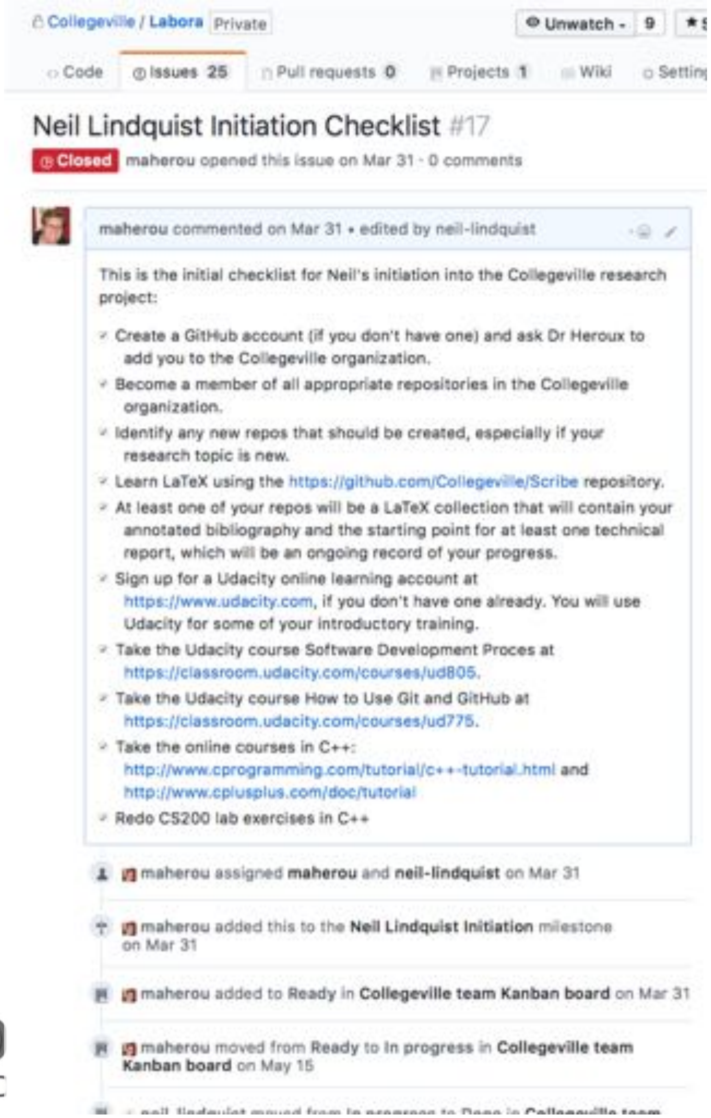
Samples from Collegeville Org: Policies, Initiation Checklist



Collegeville Research Team Policies

The following policies are meant to guide team members in their activities, establishing expectations for ongoing work.

1. Team members will conduct themselves in a professional manner, observing institutional policies given to them at student and faculty orientation.
2. Initiation, transition and exit events will be guided by creating and following an event checklist.
3. All work will be tracked in the organization issues-only repository Labora.
4. All work, notes and relevant content will be kept in a repository associated with the team GitHub organization.
5. Each team member will have an individual Collegeville repository: Lastname-Firstname-Work. This repo contains:
   i. Thesis or dissertation, as appropriate.
   ii. Annotated bibliography of resources.
   iii. Personal notes from project meetings and research activities.
6. If work is appropriate for one of the team repos, it will be retain there. Otherwise, it is kept in the team member's individual repo.
7. Team members will update project Kanban board prior to team meetings, more frequently if particularly active.
8. Exceptions to these policies are acceptable, but:
   i. Important exceptions should be approved before acting.
   ii. Other exceptions should mentioned at next team meeting or before.
   iii. Exceptions should be infrequent.
   iv. If an exception is frequent, actions or policies should be updated.
9. Any concerns not addressed by team policies should be discussed with Dr. Heroux.



Neil Lindquist Initiation Checklist #17

Closed maherou opened this issue on Mar 31 · 0 comments

maherou commented on Mar 31 • edited by neil-lindquist

This is the initial checklist for Neil's initiation into the Collegeville research project:

- ☑ Create a GitHub account (if you don't have one) and ask Dr Heroux to add you to the Collegeville organization.
- ☑ Become a member of all appropriate repositories in the Collegeville organization.
- ☑ Identify any new repos that should be created, especially if your research topic is new.
- ☑ Learn LaTeX using the https://github.com/Collegeville/Scribe repository.
- ☑ At least one of your repos will be a LaTeX collection that will contain your annotated bibliography and the starting point for at least one technical report, which will be an ongoing record of your progress.
- ☑ Sign up for a Udacity online learning account at https://www.udacity.com, if you don't have one already. You will use Udacity for some of your introductory training.
- ☑ Take the Udacity course Software Development Proces at https://classroom.udacity.com/courses/ud805.
- ☑ Take the Udacity course How to Use Git and GitHub at https://classroom.udacity.com/courses/ud775.
- ☑ Take the online courses in C++: http://www.cprogramming.com/tutorial/c++-tutorial.html and http://www.cplusplus.com/doc/tutorial
- ☑ Redo CS200 lab exercises in C++

maherou assigned **maherou** and **neil-lindquist** on Mar 31

maherou added this to the **Neil Lindquist Initiation** milestone on Mar 31

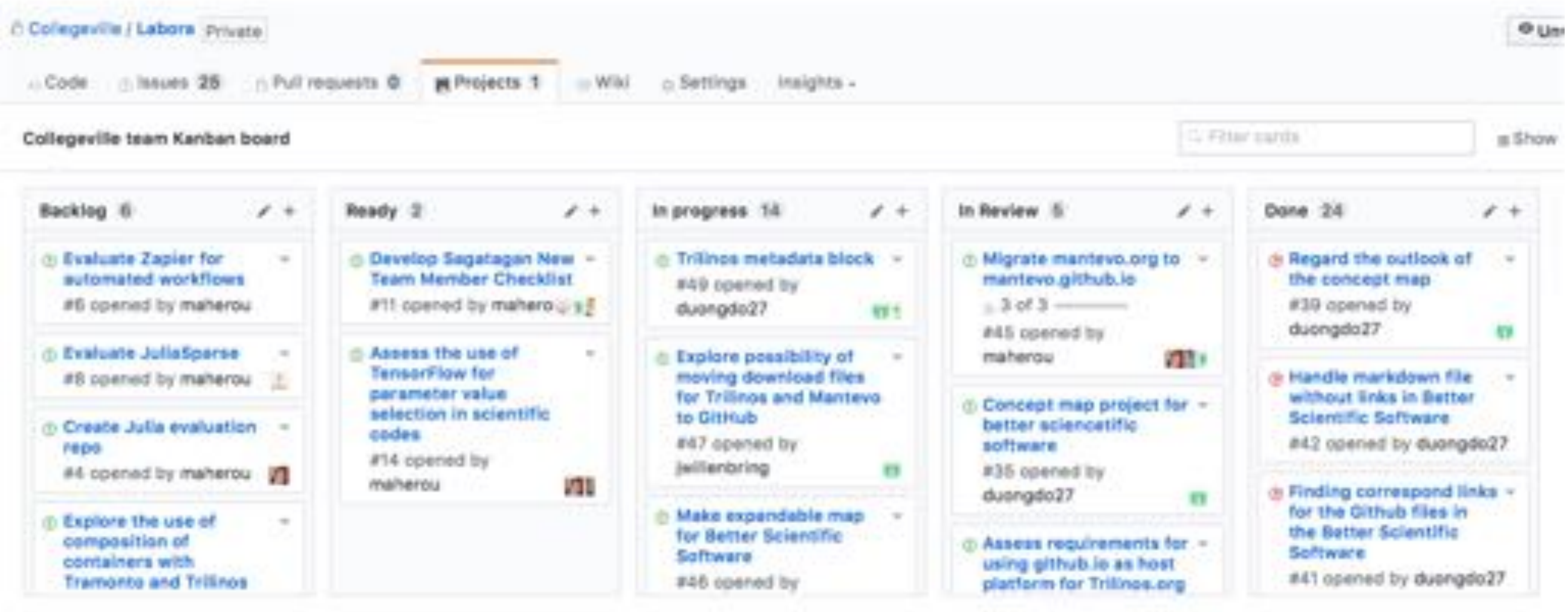
maherou added to Ready in **Collegeville team Kanban board** on Mar 31

maherou moved from Ready to In progress in **Collegeville team Kanban board** on May 15

neil-lindquist moved from In progress to Done in **Collegeville team**

IDprc

Samples from Collegeville Org: Kanban Board



Collegeville / Labora Private

Code | Issues 25 | Pull requests 0 | Projects 1 | Wiki | Settings | Insights

Collegeville team Kanban board

Filter cards

Show

Backlog 6

- Evaluate Zapier for automated workflows #6 opened by maherou
- Evaluate JuliaSparse #8 opened by maherou
- Create Julia evaluation repo #4 opened by maherou
- Explore the use of composition of containers with Tramonto and Trilinos

Ready 2

- Develop Sagatagan New Team Member Checklist #11 opened by mahero
- Assess the use of TensorFlow for parameter value selection in scientific codes #14 opened by maherou

In progress 14

- Trilinos metadata block #49 opened by duongdo27
- Explore possibility of moving download files for Trilinos and Mantevo to GitHub #47 opened by jwillenbring
- Make expandable map for Better Scientific Software #46 opened by

In Review 5

- Migrate mantevo.org to mantevo.github.io 3 of 3 #45 opened by maherou
- Concept map project for better sciencetific software #35 opened by duongdo27
- Assess requirements for using github.io as host platform for Trilinos.org

Done 24

- Regard the outlook of the concept map #39 opened by duongdo27
- Handle markdown file without links in Better Scientific Software #42 opened by duongdo27
- Finding correspond links for the Github files in the Better Scientific Software #41 opened by duongdo27

Team Management Example

Team Policy

Checklists

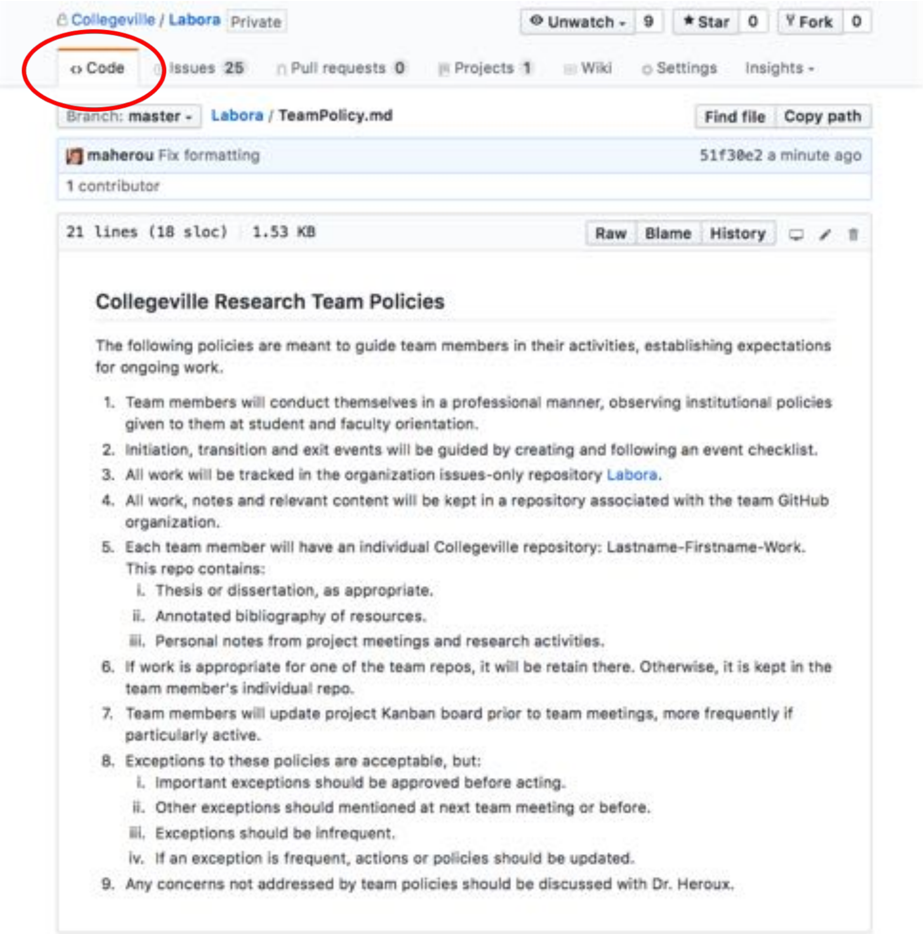
Kanban Board

Step 1: Create Issues-only GitHub repo

- Go to https://github.com/username
  - Example: https://github.com/maherou
- Create new repo:
  - Click on "+" (upper right).
  - Select New repository…
  - Give repo a name, e.g., **Issues**
  - Select Public. In real life, this repo is often private (requires $ or special status)
  - Init with README.
  - Don't add .gitignore or license.
  - Click Create Repository.

Step 2: Define Team Policy

- Create file:
  - Go to new repo: Issues.
  - Select <> Code tab.
  - Select Create new file TeamPolicy.md
- Questions to address:
  - How members support team?
  - How team supports members?
- Community version:
  - http://contributor-covenant.org
- Policy is living document:
  - Informal good practices added.
  - Avoidable bad situations addressed.



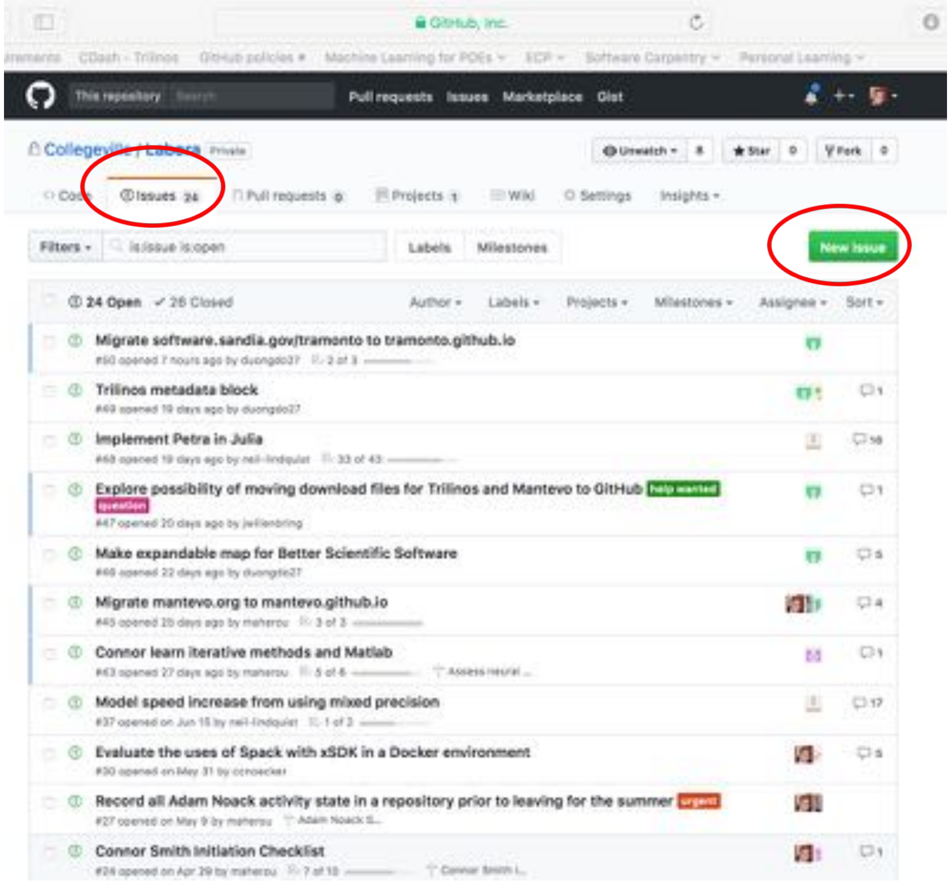
Collegeville Research Team Policies

The following policies are meant to guide team members in their activities, establishing expectations for ongoing work.

1. Team members will conduct themselves in a professional manner, observing institutional policies given to them at student and faculty orientation.
2. Initiation, transition and exit events will be guided by creating and following an event checklist.
3. All work will be tracked in the organization issues-only repository Labora.
4. All work, notes and relevant content will be kept in a repository associated with the team GitHub organization.
5. Each team member will have an individual Collegeville repository: Lastname-Firstname-Work. This repo contains:
   i. Thesis or dissertation, as appropriate.
   ii. Annotated bibliography of resources.
   iii. Personal notes from project meetings and research activities.
6. If work is appropriate for one of the team repos, it will be retain there. Otherwise, it is kept in the team member's individual repo.
7. Team members will update project Kanban board prior to team meetings, more frequently if particularly active.
8. Exceptions to these policies are acceptable, but:
   i. Important exceptions should be approved before acting.
   ii. Other exceptions should mentioned at next team meeting or before.
   iii. Exceptions should be infrequent.
   iv. If an exception is frequent, actions or policies should be updated.
9. Any concerns not addressed by team policies should be discussed with Dr. Heroux.

Step 3a: Create Issues

- Select the Issues tab.
- Click on New Issue.
- Type in task statement 1 (from list).
  - Type in title only.
- Click Submit new issue
- Repeat.



Migrate software.sandia.gov/tramonto to tramonto.github.io
#50 opened 7 hours ago by duongdo27 2 of 3

Trilinos metadata block
#49 opened 19 days ago by duongdo27 1

Implement Petra in Julia
#48 opened 19 days ago by neil-lindquist 33 of 43 16

Explore possibility of moving download files for Trilinos and Mantevo to GitHub help wanted question
#47 opened 20 days ago by jwillenbring 1

Make expandable map for Better Scientific Software
#46 opened 22 days ago by duongdo27 5

Migrate mantevo.org to mantevo.github.io
#45 opened 25 days ago by maherou 3 of 3 4

Connor learn iterative methods and Matlab
#43 opened 27 days ago by maherou 5 of 6 Assess neural ... 1

Model speed increase from using mixed precision
#37 opened on Jun 15 by neil-lindquist 1 of 3 17

Evaluate the uses of Spack with xSDK in a Docker environment
#30 opened on May 31 by ccnoecker 5

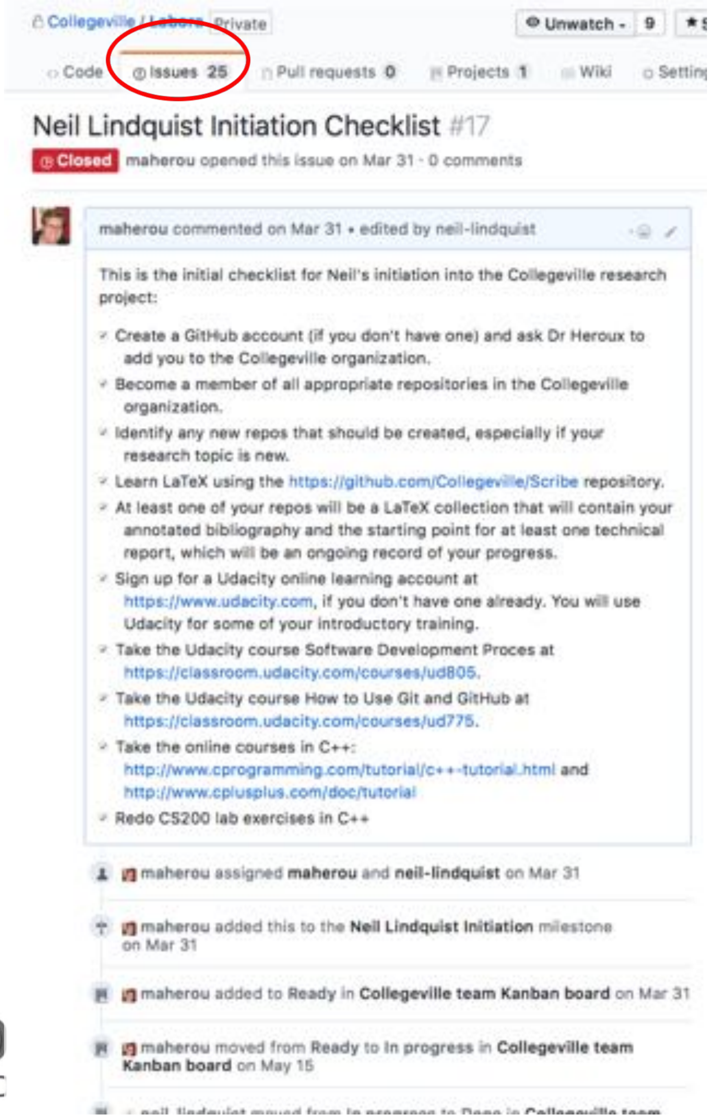
Record all Adam Noack activity state in a repository prior to leaving for the summer urgent
#27 opened on May 9 by maherou Adam Noack S...

Connor Smith Initiation Checklist
#24 opened on Apr 29 by maherou 7 of 10 Connor Smith I... 1

Step 3b: Create Initiation Checklist

- Select the Issues tab.
- Click on New Issue.
- Select a classmate.
- Type in title: Pat Evans Initiation Checklist
- Add checklist items:
  - Use syntax:
    - [ ] Description

Spaces required



Neil Lindquist Initiation Checklist #17

Closed maherou opened this issue on Mar 31 · 0 comments

maherou commented on Mar 31 • edited by neil-lindquist

This is the initial checklist for Neil's initiation into the Collegeville research project:

- ✓ Create a GitHub account (if you don't have one) and ask Dr Heroux to add you to the Collegeville organization.
- ✓ Become a member of all appropriate repositories in the Collegeville organization.
- ✓ Identify any new repos that should be created, especially if your research topic is new.
- ✓ Learn LaTeX using the https://github.com/Collegeville/Scribe repository.
- ✓ At least one of your repos will be a LaTeX collection that will contain your annotated bibliography and the starting point for at least one technical report, which will be an ongoing record of your progress.
- ✓ Sign up for a Udacity online learning account at https://www.udacity.com, if you don't have one already. You will use Udacity for some of your introductory training.
- ✓ Take the Udacity course Software Development Proces at https://classroom.udacity.com/courses/ud805.
- ✓ Take the Udacity course How to Use Git and GitHub at https://classroom.udacity.com/courses/ud775.
- ✓ Take the online courses in C++: http://www.cprogramming.com/tutorial/c++-tutorial.html and http://www.cplusplus.com/doc/tutorial
- ✓ Redo CS200 lab exercises in C++

maherou assigned **maherou** and **neil-lindquist** on Mar 31

maherou added this to the **Neil Lindquist Initiation** milestone on Mar 31

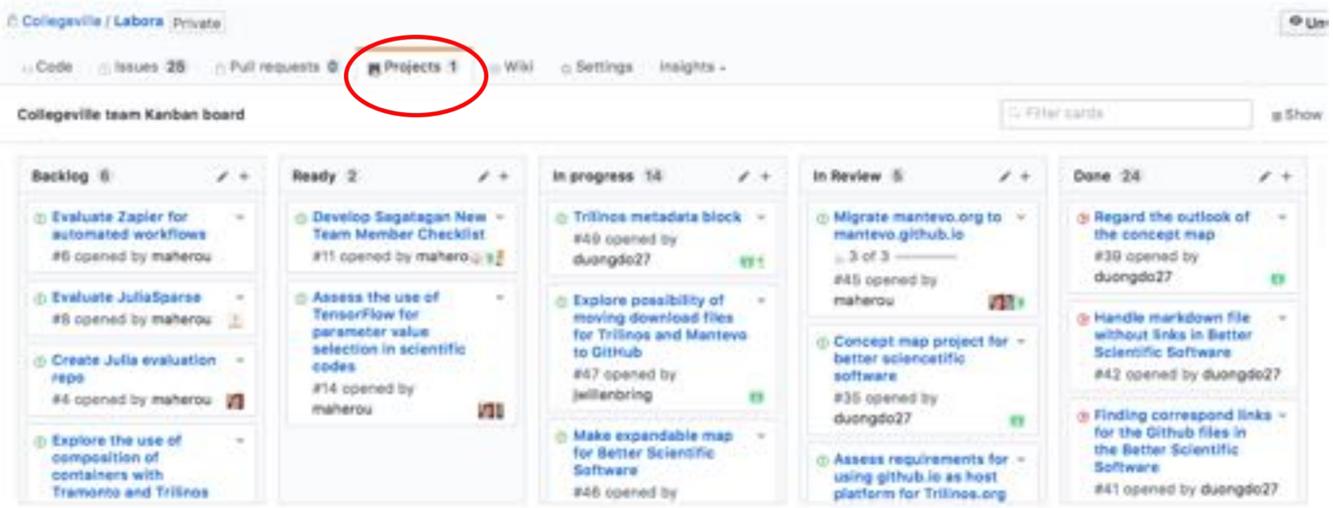
maherou added to Ready in **Collegeville team Kanban board** on Mar 31

maherou moved from Ready to In progress in **Collegeville team Kanban board** on May 15

neil-lindquist moved from In progress to Done in **Collegeville team**

Step 4: Create Kanban Board

- Select Projects tab
- Click New Project
- Use title
  - Team Kanban board
- Add these columns:
  - Backlog, Ready, In progress, In review, Done.
- Click on +Add cards (upper right).
  - Move each issue to the proper Kanban column



Collegeville team Kanban board

Backlog 6

- Evaluate Zapier for automated workflows
  #6 opened by maherou
- Evaluate JuliaSparse
  #8 opened by maherou
- Create Julia evaluation repo
  #4 opened by maherou
- Explore the use of composition of containers with Tramonto and Trilinos

Ready 2

- Develop Sagatagan New Team Member Checklist
  #11 opened by maherou
- Assess the use of TensorFlow for parameter value selection in scientific codes
  #14 opened by maherou

In progress 14

- Trilinos metadata block
  #49 opened by duongdo27
- Explore possibility of moving download files for Trilinos and Mantevo to GitHub
  #47 opened by jwillenbring
- Make expandable map for Better Scientific Software
  #46 opened by

In Review 5

- Migrate mantevo.org to mantevo.github.io
  3 of 3
  #45 opened by maherou
- Concept map project for better sciencetific software
  #35 opened by duongdo27
- Assess requirements for using github.io as host platform for Trilinos.org

Done 24

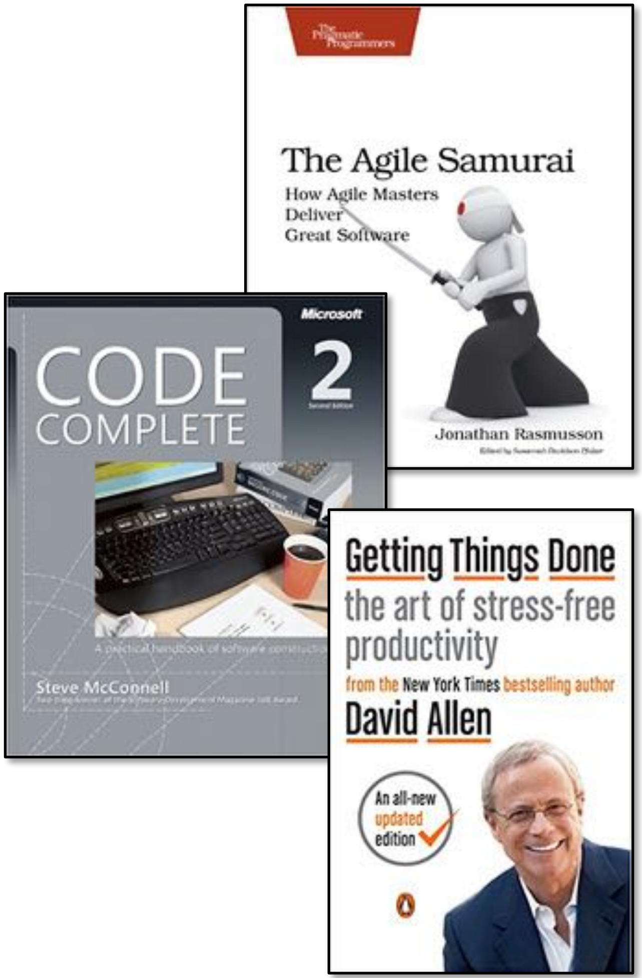
- Regard the outlook of the concept map
  #39 opened by duongdo27
- Handle markdown file without links in Better Scientific Software
  #42 opened by duongdo27
- Finding correspond links for the Github files in the Better Scientific Software
  #41 opened by duongdo27

Next Steps: Real Life

- Create a GitHub Org and set of repos for your team:
  - Each team member has an individual repo.
  - Each project has a repo.
  - One special repo for issues.
- Track all work:
  - Use checklists for initiation, exit, any big new effort.
  - Create Kanban board. Keep it current.
  - Aggregate related issues using milestones.
- Drive meetings using Kanban board.
- Adapt this approach to meet your needs.
- When you start to get sloppy, get back on track.

Other Resources

- **The Agile Samurai: How Agile Masters Deliver Great Software (Pragmatic Programmers),** Jonathan Rasmusson.
  - http://a.co/eUGIe95
  - Excellent, readable book on Agile methodologies.
  - *Also available on Audible.*
- **Code Complete: A Practical Handbook of Software Construction,** Steve McConnell.
  - http://a.co/eEgWvKj
  - Great text on software.
  - *Construx website has large collection of content.*
- **Getting Things Done: The Art of Stress-Free Productivity,** David Allen
  - http://a.co/22EPvt6
  - A classic in the personal productivity literature


The Pragmatic Programmers
The Agile Samurai
How Agile Masters Deliver Great Software
Jonathan Rasmusson
Microsoft
CODE COMPLETE
2
Steve McConnell
Getting Things Done
the art of stress-free productivity
from the New York Times bestselling author
David Allen
An all-new updated edition