

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.



Exceptional service in the national interest

RESULTS OF TRILINOS DEVOPS PIPELINE SURVEY

Sam Browne, Jonathan Hu, Curt Ober, Roger Pawlowski, Jim Willenbring

Trilinos Users Group 2023 – Developer’s Day

11:00 am November 2nd, 2023



TRILINOS DEVOPS PIPELINE (TDOP) SURVEY

- **Primary Goals**

- Assess satisfaction with current Trilinos DevOps Pipeline (TDOP)
- Determine improvements that should be considered for TDOP
 - Development, configuration, building, testing and delivery/deployment
- Used to help guide TDOP planning
- Survey distributed to primarily Trilinos Developers
 - Trilinos announce email list
 - SIAM CS&E and Supercomputing interest groups
 - **Thanks to all the respondents for your time and input!**
- Respondents
 - 37 respondents
 - Mostly Sandians and Academia (70%)
 - Most are on small teams (<6)
 - Most are developers/researchers (95%)
 - Most have more than 5 years of experience with Trilinos (81%)

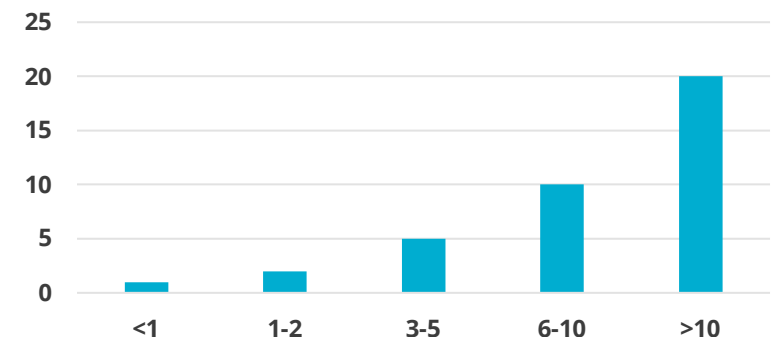
Caveats on survey

- Small population
- Small differences in statistics are probably not significant.
- Some opinions are just that.

For Reference

- 138 People in Trilinos GitHub organization.
- ~72 committers in last year

Years of Experience with Trilinos



SURVEY OF TRILINOS USAGE

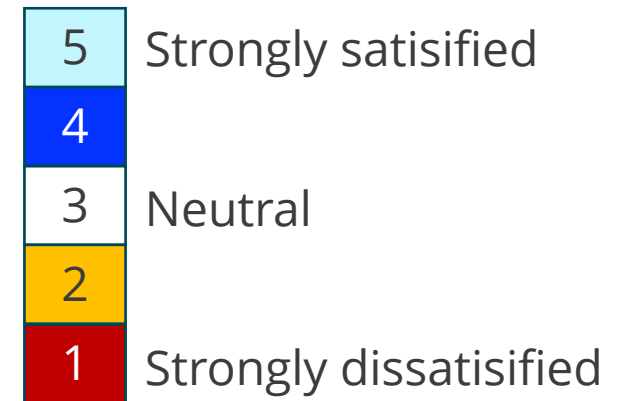
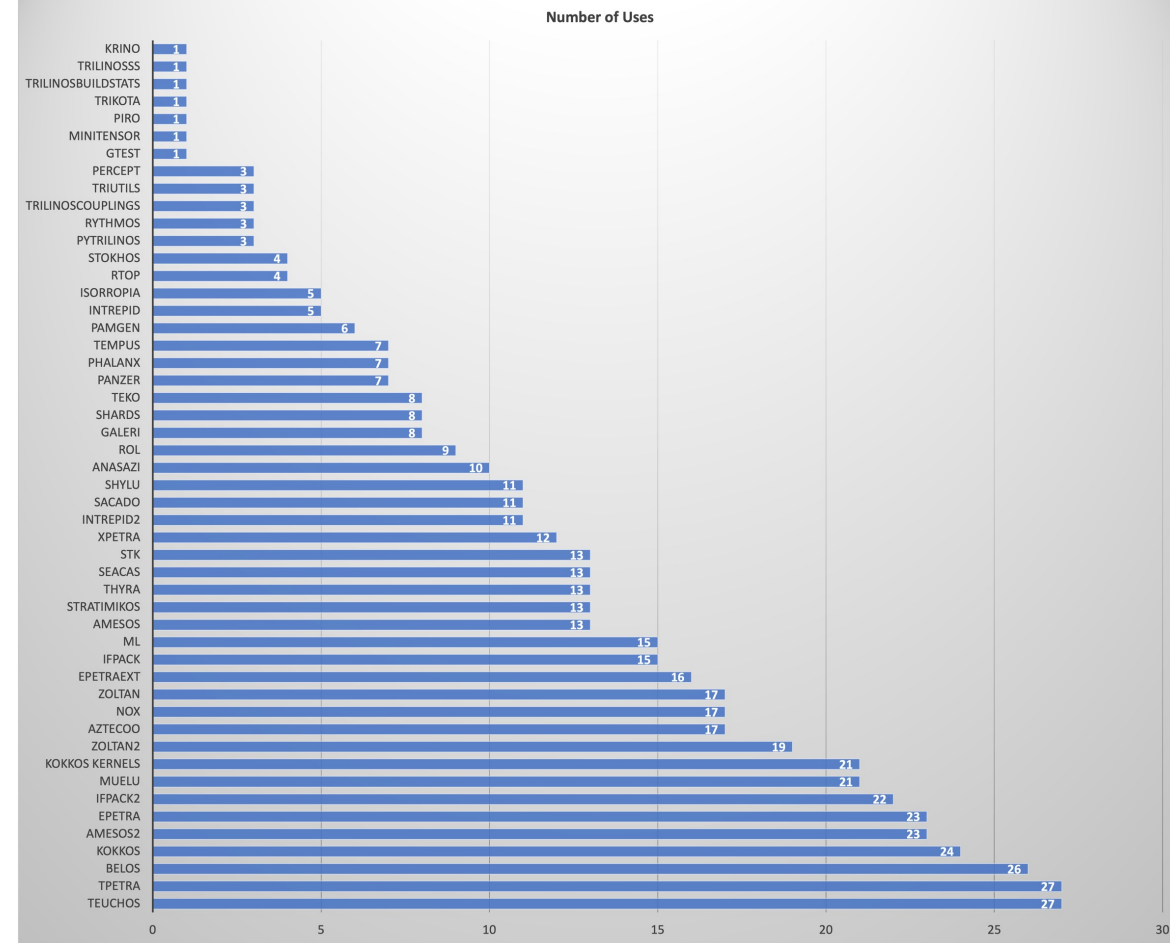
- Top 10 packages are primarily “modern” stack

- | | |
|------------|-------------------|
| 1) Teuchos | 7) Ifpack2 |
| 2) Tpetra | 8) MueLu |
| 3) Belos | 9) Kokkos-Kernels |
| 4) Kokkos | 10) Zoltan2 |
| 5) Amesos2 | |
| 6) Epetra | |

- Satisfaction with current TDOP, etc.

Version Control (git)	4.1	
Issue Submission	4.0	
Pull Request build/test	3.5	
Merge Process	3.5	
Deployment model	3.4	
Documentation	2.6	
Training	2.7	

} non-TDOP

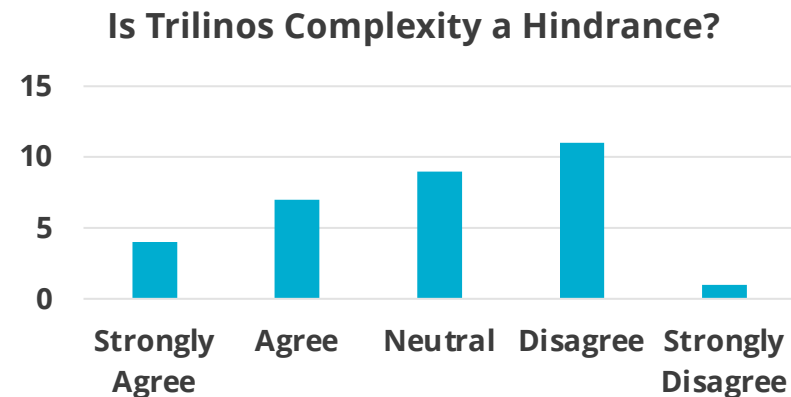
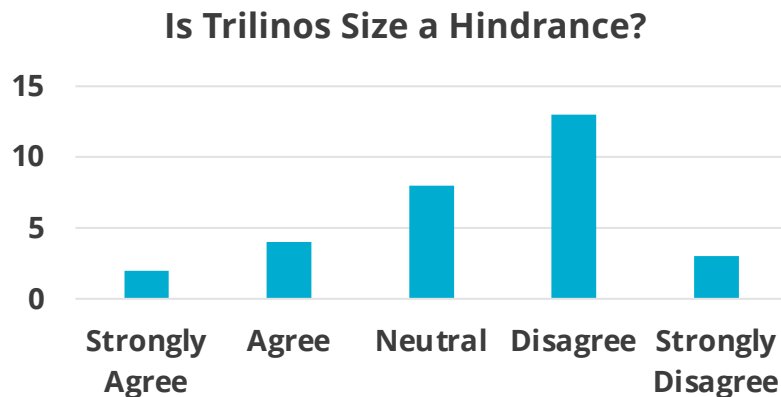




SURVEY OF TRILINOS USAGE *CONTINUED*

- Question: Given the future deprecation of Epetra-based packages (Fall 2025), do you have concerns/issues in switching to Tpetra? Yes (12) and **No (25)**
 - Reasons for concerns - Differences in solver features, and migration costs/uncertainty
- Which versions of Trilinos do you use?
 - Develop (42%), Master (27%), Release (14%), and Modified Versions (14%), Other (3%)
- Is Trilinos's size a hindrance? ~**Neutral/Disagree**
 - Too many configuration options (43% of hindrance IDers)
- Is Trilinos's complexity a hindrance? ~**Neutral**
 - Too many packages/dependencies (55% of hindrance IDers)

Notes: Neutral response on hindrances





SHOULD THE TRILINOS REPO BE BROKEN UP?

- Question: Should Trilinos be “partitioned” (i.e., no longer maintained as single repository)?
 - An important issue so we asked several related survey questions.
- Responses in favor of partitioning Trilinos (~22%)
 - Will reduce “complexity” (e.g., easier upgrade packages and a lot of functionality as independent)
 - “Maximum flexibility in development and integration workflows.”
 - Difficult to configure Trilinos due to complex interdependencies
- Responses against partitioning Trilinos (~78%)
 - Single repository
 - Package interdependencies and interoperability are tested and automatically maintained
 - No complex integration workflows
 - Partitioning increases complexity for developers and applications, e.g.,
 - Additional integration testing above normal PR testing
 - More work to manage versions, handle dependencies and coordinate multiple repos
 - Increase configure/build/integration bugs and maintenance time

Summary: Strong preference for keeping Trilinos as single repository.

HOW DO YOU PREFER TO CONFIGURE TRILINOS?

- Question : Which of the following methods do you prefer?

- **CMake/Scripting** (33%)

- Additional complexity of TriBITS not needed with modern CMake
- CMake mature enough to be used alone, industry standard.
- Package managers unready for development work.

- **CMake/TriBITS [current Trilinos model]** (37%)

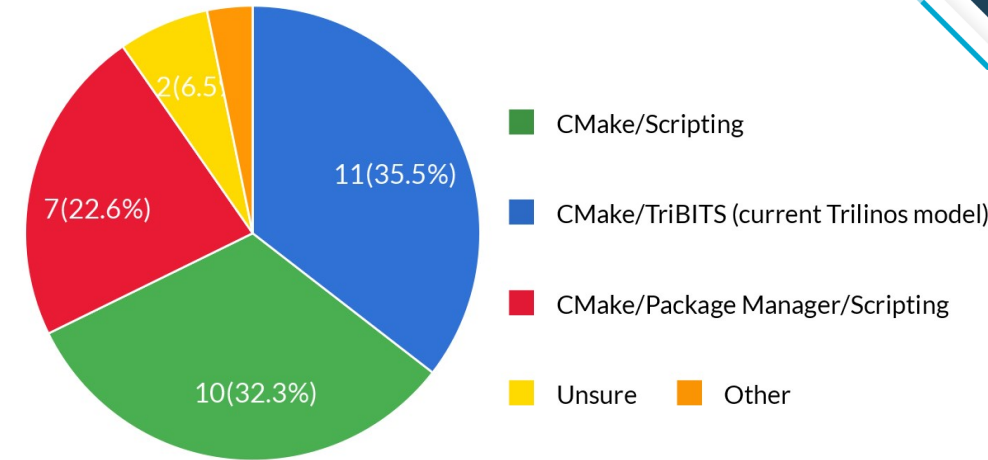
- TriBITS makes it easy to do things correctly with many package dependencies and their tests.
- TriBITS is a collection of CMake functions that make hard things simple.
- TriBITS provides ability to build each package separately, provide other features, and protects corner cases.

- **CMake/Package Manager/Scripting** (23%)

- Rely on more "standard" tools, e.g., CMake and Spack.
- Everyone uses this model.

- **Unsure** (7%)

- TriBITS is very flexible and portable. Inexperienced developers/users have trouble compiling Trilinos.



Summary:

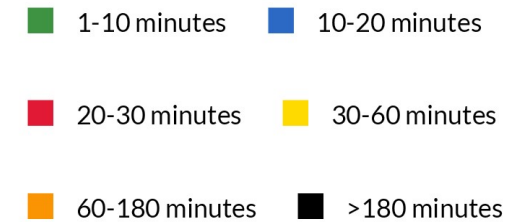
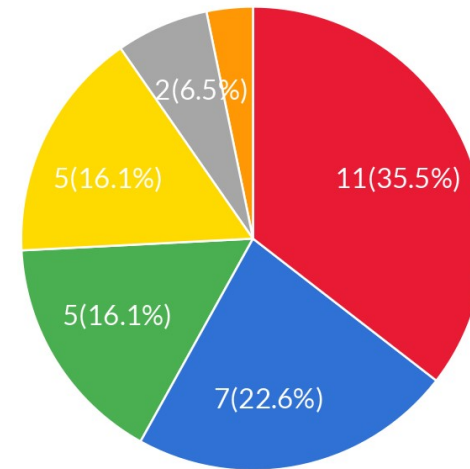
- Majority prefer either CMake or CMake + TriBITS.



DOES TRILINOS TAKE TOO LONG TO BUILD?

• Build times

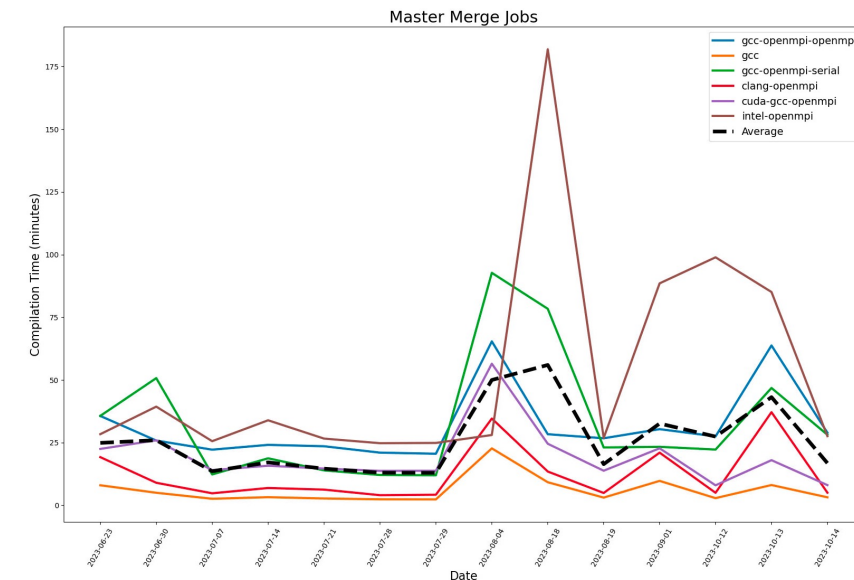
- 74% under 30 minutes
- 90% under 1 hour
- ~43 Builds and Platforms
- ~41 Compilers and versions



• Do you find Trilinos build times long relative to other parts of development cycle (e.g., application build times)?

- Yes (9), **No (21)**

Summary: Majority unconcerned with Trilinos build times, but this could stand to be improved.



USAGE OF PACKAGE MANAGERS AND TPLS

- Question: To what level do you interact with TriBITS?
 - Only through Trilinos (62%), Dealt directly with TriBITS (27%) and Use outside of Trilinos (11%)
- Question: How are TPLs acquired?
 - System installed (29%), build/maintain own versions (33%), and package manager (e.g., Spack) (33%), and Other (6%)
- Question: Do you use a package manager for your work?
 - None (34%), Spack (51%), Conan (6%) and Other (9%)
- Questions: Spack usage questions
 - How knowledgeable/experienced are you with Spack? ~somewhat
 - Most use Spack directly (no intermediate management layer) (75%)
 - Maintain own Spack recipe (47%), Use recipe from Spack (32%) and Other (21%)

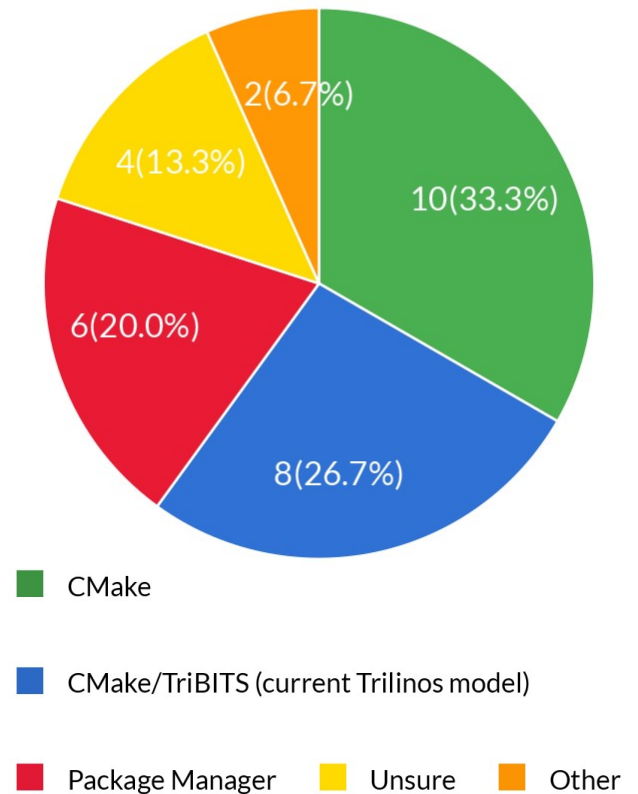
Summary:

- ~50% use Spack in some capacity

USAGE OF PACKAGE MANAGERS AND TPLS (CONTINUED)

• Question: “For TPL management, which of the following approaches do you prefer?”

- CMake (33%)
 - TriBITS makes it harder than it has to be (e.g., trying to determine libraries via “-l”, absolute path, or ...)
 - CMake mature and industry standard.
 - Package managers do not work for development.
- CMake/TriBITS (current Trilinos model) (27%)
 - Because it works. No need to change.
- Package Manager (20%)
 - Easier to upgrade/support different versions while allowing customization
- Unsure (13%)
 - Know TriBITS and works, but new developers/users have difficulties.
- Other (7%)
 - Blended – Spack can simplify TPL management.
 - TriBITS should still check compatible versions.



Summary: Majority prefer either CMake or CMake + TriBITS.



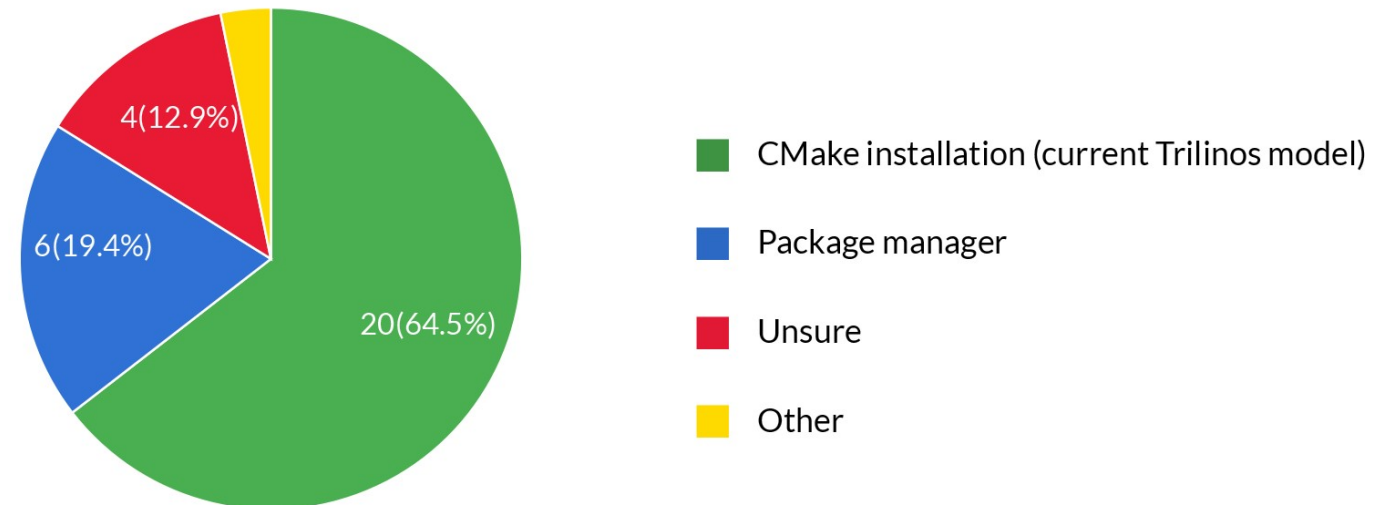
COMMENTS ON TESTING ...

- Package interdependencies and interoperability are tested and automatically maintained
- Tests/Builds are in better shape!
- Introduce “popular builds”; test them; include compiler version they work for.
- PR testing needs to be visible for external developers.
- Need more testing
 - Broader performance testing.
- Need less testing
 - Amount tests run during development; time costs running tests
- TriBITS provides common testing features
- Need monolithic testing to ensure interoperability
- Integration testing causes additional overhead



PREFERRED METHOD OF DEPLOYMENT

- Question: In relationship to deployment, which of the following approaches do you prefer?
 - CMake installation (current Trilinos model) (65%)
 - CMake mature/industry standard. Package managers don't work well enough for development
 - Package manager (19%)
 - Path for the future, work with HPC ecosystem/community, and only option forward.
 - Unsure (13%)
 - Like having full control (current model)
 - But see benefits of alternative approaches



Summary: Majority prefer CMake



OTHER THINGS WE SHOULD KNOW ...

- Question: Are there other things you want us to know (current "pain points")?
 - Improve Pipeline
 - Configuration/build/test/installation easier/faster
 - Introduce “popular builds”; test them; include compiler version they work for.
 - Broader performance testing.
 - PR testing needs to be visible for external developers.
 - Reduce Complexity
 - Remove package redundancy; Combine packages
 - Do not add more snapshotted packages.
 - Improve Accessibility
 - Need to improve adoption. Learning curve is too steep for new user/developers.
 - Documentation needs to be improved (10x)
 - Improve Communication
 - Improve informal discussions; More responsive to questions (5x)
 - Other
 - If removing TriBITS, transition to CMake. Do not start from scratch.
 - Spack wastes days of time, and not meant for development
 - Most issues related to GPUs
 - Provide Windows support.



POINTS FROM ASC DEVOPS SURVEY

- Develop
 - Too many Trilinos packages.
- Configure
 - Framework team has limited knowledge of TriBITS
 - Trilinos configuration/build too complex for our users; can't use package managers
 - Trilinos does not manage multiple configurations well.
- Build
 - Build warning clutter
 - Keep ability for single configure and build of all packages
 - External link to Kokkos build (2x)
- Test
 - Trilinos does not build/test with our configuration.
- Deliver/Deploy
 - Improve ability to integrate Trilinos consistently to applications
 - Trilinos developers might not know Spack well enough to handle build errors
 - Trilinos has a lack of frequent releases
 - Trilinos should support Spack builds, but NOT require them
 - Challenge to port to new platforms (ATS-3/4)

Note: Survey was primarily to ASC Stakeholders



THIS INFORMED THE TRILINOS DEVOPS PIPELINE PLANNING (TDOP)

Trilinos plans to ...

DEVELOP

- Remain single repository to maintain developer productivity
- Retain key capabilities of TriBITS and form a support team

CONFIGURE

- Utilize ASC DevOps common Trilinos configurations (e.g., RAMSES and CompSim)
- Provide/maintain a Spack recipe that others can use (e.g., ASC Stakeholders and Spack)

BUILD

- Maintain/support CMake+TriBITS and Spack builds
- Incorporate Containers and GitHub Actions to catch build errors and keep builds clean

TEST

- Add Integration testing for Trilinos packages (e.g., Kokkos and Kokkos Kernels)
- Support application's integration testing of Trilinos to mitigate integration issues

DELIVER DEPLOY

- Support both delivery (Trilinos GitHub) and deployment (Spack)
- Steward Trilinos's Spack recipe with support from Framework and Trilinos Developers



QUESTIONS?