

Exceptional service in the national interest



Trilinos Community 2.0



Michael A. Heroux
Sandia National Laboratories



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

WHY 2.0?

Day 1 of Package Life

From TUG
2006



- **CVS:** Each package is self-contained in Trilinos/package/ directory.
- **Bugzilla:** Each package has its own Bugzilla product.
- **Bonsai:** Each package is browsable via Bonsai interface.
- **Mailman:** Each Trilinos package, including Trilinos itself, has four mail lists:
 - package-checkins@software.sandia.gov
 - CVS commit emails. “Finger on the pulse” list.
 - package-developers@software.sandia.gov
 - Mailing list for developers.
 - package-users@software.sandia.gov
 - Issues for package users.
 - package-announce@software.sandia.gov
 - Releases and other announcements specific to the package.
- **New_package** (optional): Customizable boilerplate for
 - Autoconf/Automake/Doxygen/Python/Thyra/Epetra/TestHarness/Website

Day 1 of Package Life on GitHub & Atlassian

- **CVS -> Git:** Each package is self-contained, clonable.
 - **Bugzilla -> Issue tracking:** Integrated, Issue listings, Milestones & labels, Commit keywords. Pull requests.
 - **Bonsai -> Browsing:** Git tools, local clone.
 - **Mail lists -> Forums:** More diverse, interactive.
 - **New_package -> Stubbed out project.**
-
- **Atlassian:** Even more advanced but costly.
 - Dev Tools: Git, mercurial, other tools.
 - Jira: Issue tracking.
 - Confluence: Collaboration, communication.

Software Platforms are Commodity

- Projects can start from Day 1 with high-quality environment, global visibility.
- Need to re-state the Trilinos value propositions from first principles.

Trilinos Community 2.0

- GitHub, Atlassian:
 - Open source SW development, tools platforms.
 - Workflows for high-quality community SW product development.
- Trilinos value proposition:
 - Included these same things, but must re-evaluate.
 - Must address packages that want GitHub presence.
 - Must (IMO) move Trilinos itself to GitHub.
- New types of Trilinos packages (evolving):
 - Internal: Available only with Trilinos (traditional definition).
 - Exported: Developed in Trilinos repository, available externally.
 - Imported: Developed outside of Trilinos, available internally.

Trilinos Strategic Goals

- **Scalable Computations:** As problem size and processor counts increase, the cost of the computation will remain nearly fixed.
- **Hardened Computations:** Never fail unless problem essentially intractable, in which case we diagnose and inform the user why the problem fails and provide a reliable measure of error.
- **Full Vertical Coverage:** Provide leading edge enabling technologies through the entire technical application software stack: from problem construction, solution, analysis and optimization.
- **Universal Interoperability:** All Trilinos packages, and important external packages, will be interoperable, so that any combination of packages and external software (e.g., PETSc, Hypre) that makes sense algorithmically will be possible within Trilinos.
- **Universal Accessibility:** All Trilinos capabilities will be available to users of major computing environments: C++, Fortran, Python and the Web, and from the desktop to the latest scalable systems.
- **TriBITS Lifecycle:** Trilinos will be:
 - **Reliable:** Leading edge hardened, scalable solutions for each of these applications
 - **Available:** Integrated into every major application at Sandia
 - **Serviceable:** “Self-sustaining”.

Algorithmic
Goals

Software
Goals

Trilinos Community 2.0

- Case studies:
 - TriBITS: Was an internal package, now external.
 - DTK: Has always been external.
 - KokkosCore: Is internal. Needs to be available externally.
 - Tracked: STK, Percept, others?
- Issues to Resolve:
 - Package inclusion policies: Define for each package type.
 - Quality criteria: Contract between Trilinos and packages.
 - Workflows: Development, testing, documentation, etc.
 - Trilinos on GitHub: Evaluate.
 - Trilinos Value Proposition: Re-articulate Trilinos Strategic Goals implications.

MORE FROM 2006

Trilinos Interoperability Mechanisms

(Acquired as Package Matures)

From TUG
2006



Package builds under Trilinos
configure scripts.



Package can be built as part of a
suite of packages; cross-package
interfaces enable/disable
automatically

Package accepts user data as
Epetra or Thyra objects



Applications using Epetra/Thyra can
use *package*

Package accepts parameters from
Teuchos ParameterLists



Applications using Teuchos
ParameterLists can drive *package*

Package can be used via Thyra
abstract solver classes



Applications or other packages using
Thyra can use *package*

Package can use Epetra for private
data.



Package can then use other packages
that understand Epetra

Package accesses solver services
via Thyra interfaces



Package can then use other packages
that implement Thyra interfaces

Package available via PyTrilinos,
ForTrilinos, WebTrilinos



Package can be used with other
Trilinos packages via Python,
Fortran, Website.

Sample Package Maturation Process

From TUG
2006



Step	Example
Package added to CVS: Import existing code or start with new_package.	ML CVS repository migrated into Trilinos (July 2002).
Mail lists, Bugzilla Product, Bonsai database created.	ml-announce, ml-users, ml-developers, ml-checkins, ml-regression @software.sandia.gov created, linked to CVS (July 2002).
Package builds with configure/make, Trilinos-compatible	ML adopts Autoconf, Automake starting from new_package (June 2003).
Epetra objects recognized by package.	ML accepts user data as Epetra matrices and vectors (October 2002).
Package accessible via Thyra interfaces.	ML adaptors written for TSFCore_LinOp (Thyra) interface (May 2003).
Package uses Epetra for internal data.	ML able to generate Epetra matrices. Allows use of AztecOO, Amesos, Ifpack, etc. as smoothers and coarse grid solvers (Feb-June 2004).
Package parameters settable via Teuchos ParameterList	ML gets manager class, driven via ParameterLists (June 2004).
Package usable from Python (PyTrilinos)	ML Python wrappers written using new_package template (April 2005).

Startup Steps

Maturation Steps

NewPackage Package

- NewPackage provides jump start to develop/integrate a new package.
- NewPackage is a “Hello World” program and website:
 - Simple but it does work with autotools.
 - Compiles and builds.
- NewPackage directory contains:
 - Commonly used directory structure: src, test, doc, example, config.
 - Working autotools files.
 - Documentation templates (doxygen).
 - Working regression test setup.
- Really cuts down on:
 - Time to integrate new package.
 - Variation in package integration details.
 - Development of website.

Internal vs. Exported vs. Imported

- Internal:
 - Interoperable with other packages, both up and downstream.
 - Clearly within the scope of Trilinos functionality domain, e.g., solvers of any kind.
- Imported or exported?
 - Snapshotting provides dual view of package, imported or exported.
 - Imported: Most commits come from developers outside of Trilinos.
 - Exported: “ “ “ “ “ inside of Trilinos.
 - Imported: If you want to manage your own set of workflows or affiliate with another SW community.
 - Exported: If you don't.
- Other considerations?

Common Look-and-feel Expectations

- Consistent data management practices.
- Consistent API styles.
- Testing and other quality metric thresholds, e.g., coverity.
- What else?

Next Steps

- Characterize, prioritize activities:
 - Package classifications.
 - Inclusion policies.
 - Workflows.
 - Quality criteria.
- Reformulate value propositions.
 - What is valuable about participating in Trilinos community.
 - What is already true.
 - What is partly true and needs to improve.
 - What is missing and should be added.
- Move Trilinos to GitHub.
 - Sanity test first. Move aggressively unless required.
- Revisit NewPackage concept.
- What else?