Open-Source Scientific Software: Community Engagement and Commercial Adoption

Keith Beattie, Dan Gunter, Ludovico Bianchi
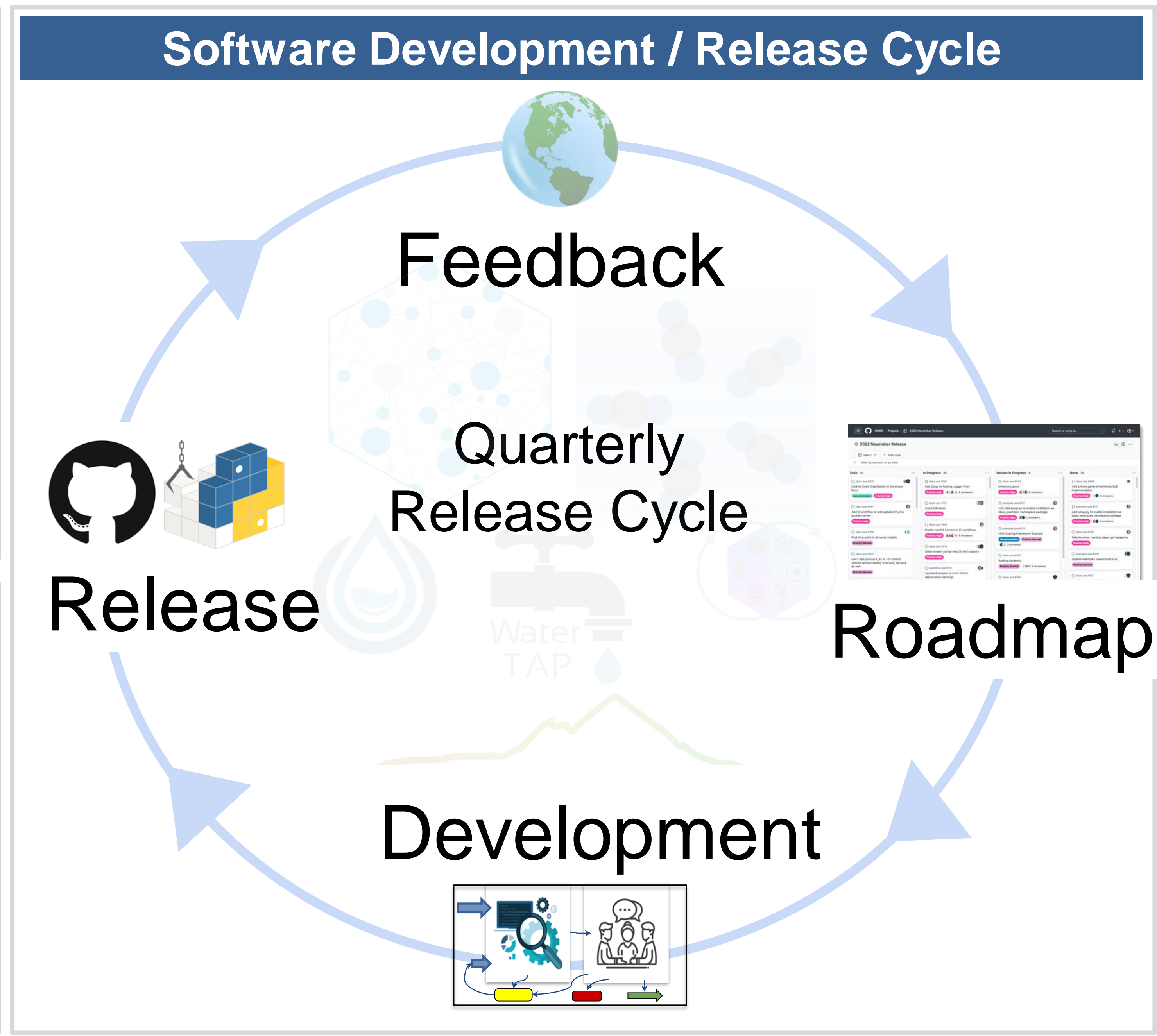
Lawrence Berkeley National Laboratory

Goals

- User & Stakeholder Confidence and results
- Best Practices in Scientific Software Engineering
- Regular date-driven Releases
- Transparent, inspectable, educational code
- Interactive engagement with users
- Automated regression testing & static analysis
- Protection of Proprietary Data
- Adoption by scientific and industry partners

Continuous Integration

Static Analysis, Unit & Regression Coverage

Software Development / Release Cycle

Feedback

Quarterly Release Cycle

Release

Roadmap

Development

Engagement & Methodology

Discussion Forums

Email & Lists

Workshops

Virtual Office Hours

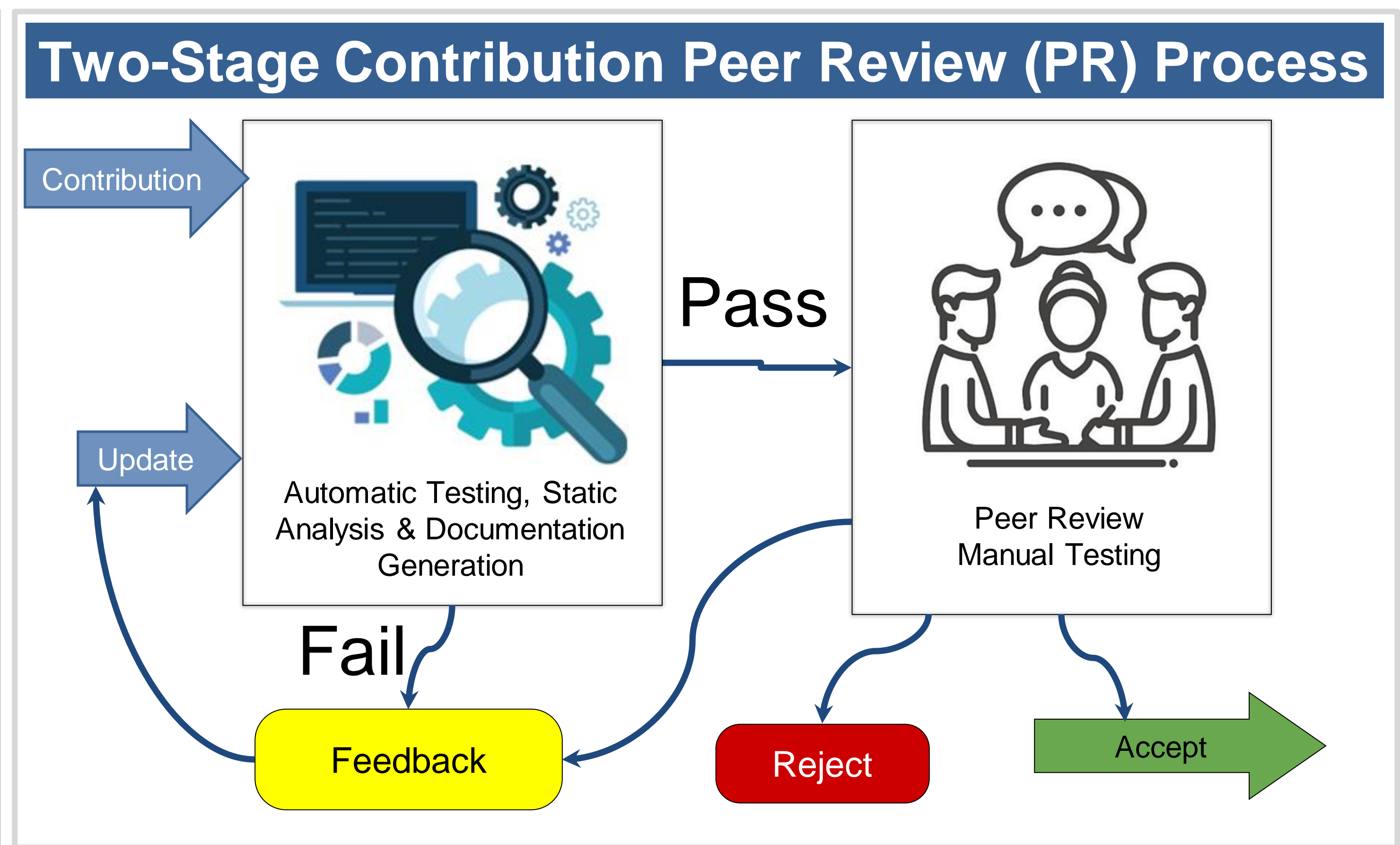
Kanban style boards for project management

Docs, Examples, Tutorials & Videos

Online docs

Examples & Tutorials

YouTube Videos

Two-Stage Contribution Peer Review (PR) Process

Contribution

Update

Automatic Testing, Static Analysis & Documentation Generation

Pass

Peer Review Manual Testing

Fail

Feedback

Reject

Accept

Common Practices & Dependencies

- All projects follow common SW development methodology & tools
- Advanced packaging system being adopted to support complex dependencies and contributions
- Many lessons learned and applied over 10+ years of multi-institutional, multi-discipline, scientific software development.