# Distributing Production-Ready Software

Brian Lin
OSG Software Area Coordinator
University of Wisconsin–Madison

PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

HT CENTER FOR HIGH THROUGHPUT COMPUTING

# Who We Are

- OSG Software Team
  - Pulls in software from various upstream providers
  - Round off rough edges with “glue” software, patches
  - Provides transition plans for unsupported software [1][2]
- Core mission: provide a well-tested, integrated software stack for sites to contribute resources to the OSG
- Currently 3 FTEs on the Software Team (packaging/development) + 0.5 FTE release manager
- Historically 3-8 FTEs for Software + Release

# Policy

- Sets expectations and allows users to make informed decisions
- OSG release series [3] define:
  - A stable software stack
  - Levels of support and timelines
  - Boundaries to make major changes to the software stack
- Software repositories [4] within each release series:
  - Different levels as software passes through the release process
  - Upcoming: software for users interested in the newest features, potentially cutting edge
  - Contrib: software that doesn't receive OSG support
- Specifying prerequisites, e.g. supported platforms [5]
- Handling of security issues [6]

# Packaging

- Package versioning provides a common understanding of state
- Follow community standards where it makes sense: OSG RPMs generally follow the Fedora packaging guidelines [7]
- Or if they don’t make sense, develop one for your community, like the OSG container image policy [8]
- Bad releases will happen: preserve the ability to rollback!
- Dependable dependencies?

# Testing

- Developing software with confidence
  - Unit tests for software developed in-house
  - Integration suite that test basic functionalities of the most commonly used parts of the software stack
- Integration tests run nightly on supported platforms to help us catch unexpected breakages due to external dependencies
- Before production release, we seek community testing of critical components [9]

# Release

- Clear internal release processes and assigned roles help us ship production-ready software
- Robust ticketing systems make releases possible
  - Keep track of items going into a release, their state, and any dependencies between items
  - Provides a paper trail to review in case of a bad release
  - Can be used for reporting metrics with sufficient metadata
- Announcements and changelogs are tools to communicate relevant updates and criticality of releases

# Support

- Support scope limited by clear policies
- Efficiency of good documentation scales with the size of your user base
- But there is a documentation baseline
  - Tell users how to update if manual intervention is required
  - How to install your software, what's required
  - Troubleshooting common issues
  - How to properly report other issues

# Links


[1] https://opensciencegrid.org/technology/policy/globus-toolkit/
[2] https://opensciencegrid.org/technology/policy/gridftp-gsi-migration/
[3] https://opensciencegrid.org/technology/policy/release-series/
[4] https://opensciencegrid.org/docs/common/yum/
[5] https://opensciencegrid.org/docs/release/supported_platforms/
[6] https://opensciencegrid.org/security/SoftwareVulnerabilityHandling/
[7] https://opensciencegrid.org/technology/software/rpm-development-guide/
[8] https://opensciencegrid.org/technology/policy/container-release/
[9] https://opensciencegrid.org/technology/policy/community-testing/

# Questions?

- Ping us on the IRIS-HEP Slack
- Reach out in #software on the OSG Slack
- Send us email at help@opensciencegrid.org

This material is based upon work supported by the National Science Foundation under Grant Nos. 1148698, 1836650 and 2030508. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

OSG

HTC CENTER FOR HIGH THROUGHPUT COMPUTING