



Troubleshooting

- **Current Initiatives**

- Identifying and addressing OSG usability problems
- Troubleshooting through user-level testing
- VORS timeouts at Vanderbilt
- Partnership between NCSA and OSG
- Documentation

- **Accomplishments Since Last Report**

- Completed the FAQ documentation
- Testing OSG 0.9.1
- Fixed a Gratia configuration problem
- VO Validation (SE)
- Resolved the problem with Atlas pilot jobs
- GIP debugging
- Resolve the TTU LIGO failure
- OSG User's Meeting
 - Attended the meeting and made a presentation on how troubleshooting helps users

- **Issues / Concerns**

- See the wish list on the third slide



1.4	Troubleshooting	Wang	75%
1.4.1	Troubleshoot end-to-end operational and performance problems	Wang	75%
1.4.1.1	Solve each identified troubleshooting task (ticket)	Wang	75%
1.4.1.2	Provide and present analysis of each troubleshooting task activity to Facility Coordinator	Wang	75%
1.4.1.3	Maintain the troubleshooting twiki pages and records	Wang	75%
1.4.2	Define Troubleshooting tools	Wang	75%
1.4.2.1	Identify and acquire tools to help VOs identify/diagnose problems	Wang	60%
1.4.2.2	Evaluate and implement OSG troubleshooting tools	Wang	50%
1.4.2.3	Cross-train expertise across troubleshooting team	Frey	35%
1.4.3	Improve documentation by leveraging user support	Wang	85%
1.4.3.1	Publish and update FAQs quarterly (to help VOs fix problems)	Wang	80%
1.4.4	Conduct formal feedback to evaluate effectiveness and efficiency of troubleshooting function	Wang	60%
1.4.5	Quarterly sub-area status, progress, issues reporting into twiki	Wang	70%



Additional Perspectives

- **Wish List (Things that we cannot do as much as we hope)**

- **Troubleshooting tools**
 - Investigate/acquire tools that would provide end-to-end tracking for failures and errors
- **Improving troubleshooting capabilities and strategies**
 - Establish a troubleshooting test bed that would provide a “typical” OSG environment where end users can quickly replicate problems.
 - Periodically select a OSG software component and conduct tests to evaluate its reliability on production sites and identify and resolve any problems encountered. Such proactive tests will give us a broader view of the infrastructure to identify trouble spots and before they become user issues
 - For example recently we are dealing with ReSS problems with around 10 sites missing from ReSS classads, though they were publishing to BDII. A LIGO user encountered this problem and with our help was able to track the issues. With a proactive tests this problem could have been identified and resolved long before the user encountered it
 - lead to constant improvement ON usability for end-users and VOs
 - Help ensure that users can run across many sites
 - Collect and track failure reports from VOs
 - Help identify the major reasons for job failures
 - Help prioritize and direct troubleshooting efforts to significant problems
- **Documentation**
 - Update troubleshooting guides
- **Gathering input from end-users, application developers, and VOs**
 - Educate and train end users troubleshooting basics.
 - Work with OSG-Education to develop modules to teach users steps they could take to identify and isolate problems
 - Each VO tends to measure their job efficiencies differently leading to an inconsistent view of the OSG
 - Work with VOs to establish a coherent measurement of calculating efficiency of jobs submitted to OSG
- **Anticipating future needs**
 - With the increasing use of a stand-alone storage elements (SEs) and deployment of opportunistic data storage, its use among end-users is bound to increase
 - We would like to anticipate the types of problems users will encounter as VOs begin to use the storage elements capabilities
 - As more users start migrating to pilot jobs, the troubleshooting teams need to be prepared for the new kinds of challenges that will be faced by pilot job users