# NASA Uses GrammaTech to Increase Satellite Network Uptime

NASA operates a complex satellite telecommunications network that links low earth orbiting spacecraft with ground stations. Human spaceflight, space science, and earth science missions depend on the network to provide launch support, on-orbit communications, and tracking services. This critical national resource is in operation 24x7, 365 days a year, and must provide continuous uninterrupted service. Consequently, NASA is continually exploring ways to improve hardware and software reliability for this mission-critical system.

The NASA Space Network, also referred to as the Tracking and Data Relay Satellite System (TDRSS), consists of nine on-orbit telecommunications satellites stationed at geosynchronous stationary positions, and the associated ground stations located at White Sands, New Mexico and Guam. NASA has an outstanding record of performance on the network, maintaining the required 99.5% proficiency every month, and usually exceeding 99.9%. Much of their success is a result of producing high quality code, though this is an ongoing challenge.

TDRSS ground stations are controlled by over eight million lines of software. Any time a code change is made, even a bug fix, there is a risk of introducing new defects. Measured over a six year period, 27% of lost network service time with the control of the Space Network was the result of software defects. In reality, it is impossible for NASA to thoroughly test such a large volume of code. Looking for ways to improve software quality, NASA explored the use of source code analysis tools for mission critical software. GrammaTech's CodeSonar® was chosen for the pilot study.

## Exploring the Benefits of Static Analysis

NASA evaluated CodeSonar to determine whether the benefits of static code analysis outweighed the cost of using the product. To do this they looked at the number and types of defects CodeSonar found, and then estimated the resulting savings based on reduced engineering expense for testing and maintenance.

Looking over Discrepancy Reports (DRs), it became clear that two software components, CSCI A and CSCI B accounted for 42% of the network down time. NASA decided to evaluate CodeSonar based on its effectiveness at identifying defects in these two components.

CodeSonar identified 585 defects in CSCI A and CSCI B. Of these, 59 were judged to be urgent. Urgent means there is a risk of disrupting the network with data corruption, processor resets, etc. The time required to identify each defect averaged about 15 minutes.



*"Beyond cost comparisons, CodeSonar provides a good value to the Space Network because a single critical error, latent in operations, puts at risk human life or once-in-a-lifetime scientific discovery."*

**– Markland Benson, NASA Computer Systems Manager**

All photos courtesy of NASA

Using its historical data, NASA calculated the average network down time due to software error was 0.35 hours per DR. Based on this, it was expected that correcting the 59 urgent DRs would result in a reduction of 20.65 hours of down time over a year, which is equivalent to half the down time for the previous year.

## Benchmarks of Excellence

To gain some perspective on the results reported by CodeSonar's analysis, NASA compared these results with similar projects. CodeSonar reported 1.18 defects per thousand source lines of code for the two modules. In comparison, Jet Propulsion Laboratories (JPL) performed a case study reporting 1.2 fielded defects per thousand lines of code. In another case, researcher Caper Jones found that CMMI Level 5 organizations deliver software that has 1.05 defects per thousand lines of code. This puts NASA code on a par with the best in the industry.

Motivated to make further improvements in software quality, NASA continued its costs versus benefits analysis of using GrammaTech. At this point, NASA had collected enough information to complete its mathematical model for estimating the cost savings. Since the ground station software was already in operation, all new code would be coming from change requests or bug fixes. NASA used statistical data from 204 prior DRs for its calculations. It found that on average for each DR, 209 lines of code were created or modified. Also, 47.6 hours of engineering or testing time were required.

Using CodeSonar's result of 1.18 defects per thousand lines of code as a multiplier for the 209 lines of code per DR, NASA calculated that each change request would result in 0.25 total defects or approximately 0.03 urgent defects. This means that if another 204 DRs were implemented, programmers would be injecting 51 total defects or 6 urgent defects. These are defects not previously in the system and that could be detected and prevented by using CodeSonar.

## Calculating the Return on Investment

For the sake of argument, consider that 10%, or about 6, of the newly introduced defects are discovered in test and require rework. Had these defects been found by the programmer, historical data shows that for each defect, 6.29 hours of time for software test could have been prevented. So, approximately 36 hours of wasted effort occurs by not using CodeSonar here. Now, consider that an additional 10% of the newly introduced defects are propagated to operations and must be closed out via the discrepancy report process. Using $50 per hour as an estimate of the cost to perform the work,

each discrepancy caught in test costs $315 (6.29 X $50) to rework and each discrepancy caught in operations costs $2,380 (47.6 X $50) to correct. Multiplying each of these by the 6 DRs comes to a total cost of $16,170. Additionally, the six defects entering operations would result in two hours of network down time.

Though these costs are important, they pale in comparison to the life threatening situations that can occur with a serious network failure. One serious incident can endanger the TDRSS fleet, astronauts, or the collection of crucial scientific data.

From a financial perspective as well, a serious event is much more expensive. Special teams are created to investigate root causes, taking into account all aspects of the ground and space segments. This can be an extensive process involving hundreds, or in the most severe cases, thousands of hours of effort. For example, a single incident that takes 500 hours to resolve at $50 per hour would cost $25,000. Reviewing operational records, NASA found that had they used CodeSonar, one to two significant anomalies (requiring hundreds of hours of work to close) each year might have been prevented.

It is clear that the greatest benefit derived from using GrammaTech CodeSonar is the reduced risk of serious incidents. NASA's analysis of the return on investment was a useful exercise, and proved that the savings in operational costs alone justify the cost of CodeSonar. NASA concluded its analysis with the recommendation that any organization that depends on internally maintained software for mission critical functions should strongly consider adopting static analysis as part of its product lifecycle.

### References

1.  NASA. *Technology Infusion of CodeSonar into the Space Network Ground Segment (RII07) Final Report*. URL: http://sarpresults.ivv.nasa.gov/ViewResearch/144.jsp

2.  NASA. *Technology Infusion of CodeSonar into the Space Network Ground Segment (Technical Briefing).* URL: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20090038682_2009038631.pdf

**GRAMMATECH**

**GrammaTech, Inc.**
531 Esty Street
Ithaca, NY 14850
Tel: 607.273.7340
www.grammatech.com