

42nd SYMPOSIUM ON SAFETY AND QUALITY IN SPACE ACTIVITIES (D5)
From Parts to Systems : Contribution of Tests on Performance Prediction and Assessment (1)

Author: Mr. Marcos Romani
Instituto de Aeronáutica e Espaço (IAE), Brazil, mromani@iae.cta.br

Mr. Pedro Takahashi
Brazil, takahashi@iae.cta.br
Dr. Carlos Lahoz
Institute of Aeronautics and Space (IAE), Brazil, lahozchnl@iae.cta.br

A PROCESS OF CODE INSPECTION FOR SPACE SOFTWARE

Abstract

The purpose of this paper is to present a process of code inspection adapted to mission critical software, based on the experience learned from the project of the Brazilian Satellite Launcher VLS-1. The source code of the VLS-1 was used in three different prototypes of this launcher, but it has never suffered a code inspection. This process is aimed to be included in the activities of software Validation and Verification of the Institute of Aeronautics and Space (IAE). The process of code inspection created to mission critical software of the IAE consists of execution of following: planning, preparing, definition of checklist, inspection, meetings, follow-up, closing. The inspection was planned and coordinated by the Software Quality Assurance Group that leaded the meetings, assigned roles and their responsibilities, and distributed tasks for each member of inspection group. The inspection was designed based on a checklist previously defined, which in turn was based on the specific characteristics of the code (legacy code, shortage of available documentation) and on good practice rules for language C programming. Based on the activities of code inspection, a technical report describing the adopted procedures was written. And the experience gained through them could be used to enhance software V&V, making the software more reliable. The inspected code with correction of its anomalies will be used as a base for the version of the next prototype, planned to be launched in year 2010. The creation of the process of code inspection for software projects in the IAE, beyond detection of code anomalies, has brought other benefits like: transferring of the experience among the participants, developing team work, increasing in the reliability of the new version of the software, finding evidences that checklist is a good form of transmitting lessons learned. The inspection results, anomalies detected, can be used as an indicator of which software components will require more rigorous tests and which will need more code coverage.