Software Validation & Verification Plan for Team Cyclone
Cyclone Stellar Monitoring
20 December 2011
Nyota Uhura, Tester & Integrator

I. <u>Purpose</u>
This document describes the verification and validation plan for the software for the Cyclone Team's Stellar Monitoring Project[1]. Briefly, validation means "Did we build the right system?", that is, are we building the system that the customer wants, as expressed in the requirements, and verification means "Are we building the system right?", i.e., does the system conform to its specifications?

II. <u>Method</u>
We will use a variety of validation and verification techniques for the software, as shown in the following list.[2]
1. Requirements reviews by stakeholders, particularly including potential customers, and the Cyclone Team. Requirements reviews will be accomplished in the following ways.
   a. Review of Concept of Operations by potential customers.
   b. Review of Software Requirements Specification (SRS) by architects, developers, and testers.
   c. Review of prototypes of user interface and key functionality by potential customers.
   d. Construction of prototypes intended to focus on validating the technology that we will use.
   e. Trial use of alpha and beta prototypes by potential customers showing that the riskier requirements can be met and the system will be usable.
2. Design reviews
   a. Module and other structures, including module guide, uses structure, and process structure.
   b. Module interface specification reviews.
   c. For critical modules, module internal reviews.
3. Code reviews.
   a. All code must be reviewed at least once. Code in critical modules must be formally reviewed. For less critical modules, the review method will be at the discretion of the developer of the module.
4. Testing
   a. All modules will be black-box tested against their interface specifications, at a minimum showing that they pass all test cases specified in the interface specification.

---

[2] Note that, as specified in the configuration control plan, all documents and code will be baselined after review.

b. Module white-box testing will be performed by the developer(s) of the module. All critical modules must achieve at least 75% code coverage, as determined through the use of the PolyFlaw tool.

c. Integration testing will be based on the uses relation among modules. The integration tests to be performed will be based on the requirements. Progressively larger subsets of the system will be integration tested, depending on how the requirements are organized into subsets (based on use cases) and the Concept of Operations.[3] See Appendix A for the initial proposed subsets.

5. Model checking.

a. Model checking will be used to assure that the system produces outputs at the appropriate intervals and without deadlocks occurring.

For reviews we will use the active review process. We will build review questions into each specification and into comments in the code.

III. Schedule and Resources

Reviews are incorporated into the schedule shown in the WBS as indicated in the project plan. As a document or code for a module is completed as part of a use case implementation, it will undergo review. The role with responsibility for the document or code, e.g., systems engineer, architect, or developer, will schedule the review and ensure that it is conducted as required.

IV. Measures

The measures described in the project plan will be used as part of the measures for the V&V process, as appropriate. In addition, our goals will be to assure that there are no critical or major defects, and at most 10 minor defects in the first release of the Stellar Monitoring System, that at least 60% of the code has been covered in tests, and that there are at most 10 minor typographical errors per specification, including requirements, concept of operations, design, module guide, and code comments.

V. Acceptance Criteria

Consistent with the goals stated in section IV, Measures, our acceptance criteria for the first, and following, releases are as follows.

Zero critical defects.

Zero major defects.

At most 10 minor defects.

At most 10 typographical errors in the Concept of Operations, and no other errors.

At most 10 typographical errors in the SRS, and no other errors.

At most 10 typographical errors in the specification or description of any architectural structure, including module structure, uses structure, and process structure, and no other errors.

Zero errors in the specifications of any black-box test cases, including module interface tests and integration tests.

---

[3] See Concept of Operations for the Cyclone Stellar Monitor.

At most 10 typographical errors in the comments in the code for any module, and no other errors.

Zero inconsistencies among SRS, design specifications of any type, code comments, and specifications of test cases.

Code coverage in testing of 60%.

VI. Responsibilities

Quality Assurance (QA) has responsibility for deciding whether or not acceptance criteria have been met for any release. QA is also responsible for tracking measures during development and posting them where they may be viewed by anyone on the development team. QA will issue a report at the end of each development iteration containing the measures for that iteration, and a report at the end of each release to customers.

Appendix A: Initial Proposed Subsets for Integration Testing

Subset 1: Basic data collection, without averaging or transmitting.
        Modules and processes to be included: TBD

Subset 2: Subset 1 plus averaging and additional data analysis.
        Modules and processes to be included: TBD

Subset 3: Subset 2 plus data transmission.
        Modules and processes to be included: TBD