

Policy for software development and quality assurance

Designing and coding

System parts (NPGM, NAS, N4A) have separate and distinct areas of responsibility in the system architecture. Application Programming Interfaces are kept as clean as possible to ease the development, integration and testing.

We strive for writing clean code. Clean code is defined as “software code that is formatted correctly and in an organized manner so that another coder can easily read or modify it”.

Developers help each other to improve the source code by reading it often and suggesting improvements.

We document our source code using DoxyGen style.

Controlling versions and changes

Subversion and DreamWeaver are used for version control of our software; both production code and test code. Git will be introduced as the common version control system during 2016.

Developers commit their day-to-day work on the main trunk. The main trunk must compile and pass regression tests at all times. Branches are created only when really needed.

Change requests are given a unique identity, prioritized and planned for. The unique identity is referred to when commenting the specific changes in the source code.

When the system has been tested and is ready for release, the source code for that release is tagged.

Testing

Unit testing, function testing, integration and system testing is done in development systems. System characteristics are tested in a production-like system.

Unit tests and function tests for new or changed functions are written and run immediately.

System parts are integrated as soon as possible to early reveal any misinterpretation of interfaces.

The functionality and characteristics of the whole system is then tested before release.

Tests are automated as far as possible to speed up regression tests and keep a high system quality.