

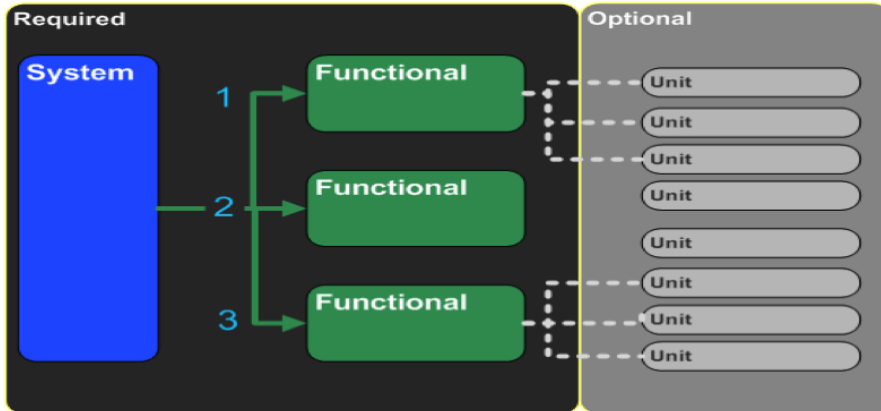
Development Process

The Scrubber build process is *test driven* with continuous build integration.

Scrubber follows the open source mantra *release early release often*, and all features and schedules are available in the Roadmap.

By *test driven* we intend that all new functionality is checked in with a corresponding functional test. The minimum test level is *functional* not *unit*. For example:

- Adding two integers does not require a unit test.
- Adding a new feature or fixing a bug requires a functional test.



Our Continuous Integration process is powered by Maven and [Bamboo](#). Note the following conventions:

- Committers - Before you check in a change please ensure that the build completes locally.
For example: `trunk$ mvn clean install`
- Bamboo continuously checks for changes to the source code
- After every check in, Bamboo runs the complete build, including tests.
- When the build completes, the built artifacts (jars, wars, and more) are deployed to our Maven repository.
This ensures that developers always have up-to-date versions of all dependencies.
- Bamboo runs integration tests across modules in the spin/integration project.

Release Early Release Often

Releasing often means short, iterative lifecycles that are tracked as issues, not *waterfall-style* specifications.

Please keep project members up to date with what you are doing by using the [mailing lists](#).

You Break It You Bought It

If you break the build please fix the build or acknowledge that we will revert your changes.

You Branch It You Own It

If you make changes in a branch, please keep everyone posted via the [mailing lists](#).

When you are done with the branch, please [contact us](#) about merging your changes back into the trunk.

Branches are rare, and mostly occur to accommodate a *stable release* while *trunk development* continues. For example, after a release we may make numerous point fixes while trunk development continues. All changes will be merged back into trunk.

No Forks Please

Even with distributed development, conflicting schedules, and many hospitals, we have managed to keep one codebase.

Please respect this time honored tradition.