

## Functional Tests

On C3, there are also literally hundreds of functional tests, which test the system from end to end, input files to output files. The test cases were defined by the users, and developed by a separate team whose job was to create tests.

The functional tests have their own special GUI, which displays each step of the test, its execution time, and whether or not it succeeded. There is a score displayed.

Functional tests did not have to score 100% during the project: adding features makes their scores go up. They do have to score 100% before release, and any substantial change in the tests is always worthy of note.

We have a short suite of functional tests that we run before moving the code to GemStone, q.v., and during development all functional tests were run every day and the scores reported to the team. The testing team was proactive in recognizing when test failed that used to run, and would inform the developers as soon as something appeared to have gone wrong.

One mistake that we made with the functional tests was that they were not sufficiently end to end. Even after we made them run from input files to output files, we subsequently discovered differences in the downstream programs that used the files. This typically happened when we had misinterpreted what the output should be; the file had on it what we intended, and the test checked for that value, but the legacy program expected something else. Let your functional tests span as far as possible from input to output: the defects will appear wherever you stop.

---

[\[ Home \]](#) [\[ XP \]](#) [\[ XP Practices Frame \]](#)

© 1997, 1998, Ronald E Jeffries  
[ronjeffries@acm.org](mailto:ronjeffries@acm.org)  
<http://www.armaties.com>