Quality Programming Ceremony

Computer programming as a social ceremony providing test driven development, continuous code reviews, and continuous code integration.

Purpose

To build quality into the code with each new line, to reduce the costs of adding new features into existing software systems, and to ensure the entire system is understood by multiple team members.

Preparation

A whole series of software development tools are selected and installed to support this ceremony.

Participants (roles)

Software Developers

Pattern (structure)

Open—A pair of developers selects their next task.

Execute—Working together as a pair, they perform all of the following in sequence in short cycles (always less than a work day long). No code is left checked-out overnight.

1. Run formal automated tests on the shared code repository, all tests must pass.
2. Check out the part of the code to be worked on.
3. Write a failing automated unit test.
4. Write code to pass the automated test.
5. Run the automated unit test and ensure all of them pass.
6. Improve the design of the code, doing the previous steps.
7. Continue to write code for the task, using the previous steps.
8. Ensure all automated tests pass.
9. Check the code back into the shared repository, with the new automated tests.
10. Run all the tests on the share code repository; if they do not pass 100%, remove the changes until they pass.

Close—The pair marks the task as complete.

Product

A functional and tested piece of code checked into the shared code repository.