Centralize Change

Supporting Software Development with ChOP

ChOP in Debugging
• Run tests with every change
• Tests run in background
• Developer is warned of defects while programming
• Use change dependencies to determine interesting spots in source code that could be the cause of the defect.

ChOP in Testing
• Test suites grow
• Inefficient testing
• Run only the relevant tests
• Find relevant tests using change dependencies

ChOP in Conflict Detection
• When merging two sets of changes
• Detect conflicts by checking preconditions of the changes.

By Quinten David Soetens

Quinten David Soetens (quinten.soetens@ua.ac.be)

Change Oriented Programming (ChOP)
• Changes as First Class Entities
• Change = Main development entity
• Software System = Set of Changes
• Software Evolution = Set of Changes
• Incremental view on software
• Fine granularity

ChOP originated in a PhD from the VUB: "A bottom-up approach to program variation" by Peter Ebraert, 2009

Test of the Undo Functionality

Buggy Undo Functionality

Corrected Undo Functionality

Which tests should I run when changing this method?