

CLEANROOM SOFTWARE ENGINEERING

- Harlan Mills (Linger, Dyer, Poore), IBM, 1980
- Analogy with electronic component manufacture
- Use of statistical process control features
- Certified software reliability
- Improved productivity; zero defects at delivery

KEY FEATURES

- Usage scenarios; statistical modeling
- Incremental development and release
- Separate development and acceptance testing
- No unit testing or debugging
 - Formal reviews with verification conditions

CLEANROOM PROJECTS

Table 1.
Selected sample of Cleanroom projects-
(All other projects known to authors report substantial improvements in quality and productivity.)

Year	Applied technologies	Implementation	Results
1980	Stepwise refinement Functional verification	Census, 25 KLOC (Pascal)	<ul style="list-style-type: none"> • No failure ever found • Programmer received gold medal from Baldrige
1983	Functional verification Inspections	Wheelwriter, 63 KLOC, three processors	<ul style="list-style-type: none"> • Millions of users • No failure ever found
1980s	Functional verification Inspections	Space shuttle, 500 KLOC	<ul style="list-style-type: none"> • Low defect over entire function • No defect in any flight • Work received NASA's Quality Award
1987	Cleanroom engineering	Flight control, 33 KLOC (Jovial), three increments	<ul style="list-style-type: none"> • Completed ahead of schedule • 2.5 errors/KLOC before any execution • Error-fix effort reduced by a factor of five
1988	Cleanroom engineering	Commercial product, 80 KLOC (PL/I)	<ul style="list-style-type: none"> • Certification testing failure rate of 3.4 failures/KLOC • Deployment failures of 0.1/KLOC • Productivity of 740 lines/man-month
1989	Partial Cleanroom engineering	Satellite control, 30 KLOC (Fortran)	<ul style="list-style-type: none"> • Certification testing error rate of 3.3 failures/KLOC • 50-percent improvement in quality • Productivity of 780 lines/man-month • 80-percent improvement in productivity
1990	Cleanroom engineering with reuse and new Ada design language	Research project, 12 KLOC (Ada and ADL)	<ul style="list-style-type: none"> • Certified to 0.9978 with 989 test cases; 36 failures found during certification (20 logic errors, or 1.7 errors/KLOC)

DEFECT RATES

- Traditional
 - Unit testing: 25 faults / KLOC
 - System testing: 25 / KLOC
 - Inspections: 20 - 50 / KLOC
- Cleanroom
 - < 3.5 / KLOC delivered
 - Average 2.7 / KLOC between first execution and delivery

BASIC TECHNOLOGIES

- Incremental Development
- Box-Structured Specification
- Function-theoretic verification
- Statistical usage testing

INCREMENTAL DEVELOPMENT

- Typical system < 100KLOC
- Increment: 2 - 15KLOC
- Team size < 14
- Each increment *End-to-End*
- Overlapped development of increments
- 12 - 18 weeks from beginning of specification to end of test
- Partitioning is difficult and critical

FORMAL SPECIFICATION

- Box-structured design
 - Black box: stimulus-response
 - State box: formal model of system state
 - Clear box: hierarchical refinement
- Program functions
- Verification properties of control structures

BOX-STRUCTURED SPECIFICATION AND DESIGN

- **Black Box:** stimulus / condition / response; organized into tasks; Z has been used for specification; top-down, stepwise refinement; concurrency supported
- **State Box:** data / history view; model oriented
- **Clear Box:** procedural control (sequence, alternation, iteration, concurrent; contains nested black boxes)
- Box Definition language

STATE BOXES

(MODEL-BASED FORMAL SPECIFICATION)

- Description of system state in terms of *domains* (data structures without memory limitations)
 - Sets, sequences, records, lists, maps, relations
- Specification of state *invariant*
- Specification of operations
 - Name
 - Arguments with domains
 - Validity condition (*precondition*)
 - Effect on state (*postcondition*)
- Each operation must maintain the invariant

RESULTS

- Defects: 2 - 5 / KLOC versus 10-30 / KLOC for debugging
- Productivity: 3 - 5 × improvement in verification over debugging
- Reliability: statistical usage testing 20 × as effective as coverage testing

CLEANROOM TOOLS

- Test case generator
- Reliability analysis package
 - Spreadsheet
- Verification-based inspection syntax analyzer
 - Script for inspection
- Management assistant
 - Reports on process