- Cleanroom
  - attempt to mathematically-based, scientific engineering process of software development
  - "Cleanroom software engineering yields software that is correct by mathematically sound design, and software that is certified by statistically-valid testing" – SEI
  - Objective: achive quality by design rather than through testing – time is spent in design and verification

- Harlan Mills (Linger, Dyer, Poore), IBM, 1980
- Analogy with electronic component manufacture
- Use of statistical quality control features
- Certified software reliability
- Improved productivity; (near) zero defects at delivery
- Unit – software use
  - can be defined in number of ways based on application

Key Features

- Usage scenarios; statistical modeling
- Incremental development and release
- Separate development and acceptance testing
- Program proofs; no unit testing

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

- Box-Structured
- Specification
- Function-theoretic verification
  - before any code is compiled/executed
  - debugging not permitted
- Statistical usage testing

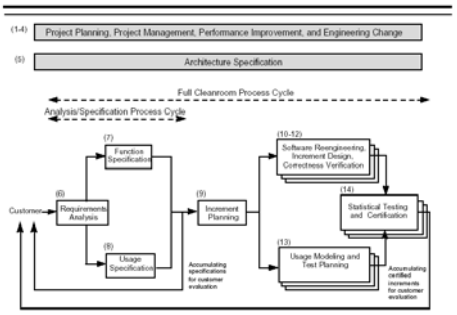

Figure 1. Cleanroom Process Flow

Incremental Development

- Typical system < 100KLOC
- Increment: 2 - 15KLOC
- Team size < 14 (6-8)
- 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 breakdown
- 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-box (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

Statistical Usage Testing

- Certification of reliability
- Process control
- Cost-effective orientation
- Guidelines for test completion (desired reliability reached) or redesign (too many failures found)
- Stratification mechanism for dealing with critical situations
- But questions exist on how to feed back the results of testing to the development team

Testing process

- Usage distribution models
  - From competitors, earlier versions, analysis
- Markov usage chain
  - State transition probability matrix
- Statistics
  - Π (proportion of time spent in each state)
  - n (number of states visited before a given state is reached)
  - s (number of tests needed to reach a state).
- Random test generation
  - Design required
- Test execution and test chain generation, including failure states
- Statistics
  - R (reliability)
  - MTBF (mean time between failures)
  - D (divergence of test chain from usage chain)

INITIAL USAGE MODEL

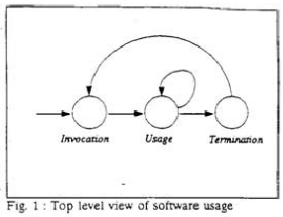

Fig. 1 : Top level view of software usage

REFINED USAGE MODEL

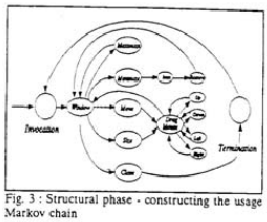

Fig. 3 : Structural phase - constructing the usage Markov chain

PROBABILITIES FROM SCENARIOS

USAGE MODEL STATISTICS

Table III : Analytical Results for the Example Usage Model

State	π	n	z
Invocation	0.093750	10.7	1
Window	0.187500	5.3	0.5
Maximize	0.015625	64	6
Minimize	0.015625	64	6
Icon	0.015625	64	6
Restore	0.015625	64	6
Move	0.031250	32	3
Size	0.031250	32	3
Drag Mouse	0.234375	4.3	0.4
Up	0.015625	64	6
Down	0.078125	12.8	1.2
Left	0.046875	21.3	2
Right	0.031250	32	3
Close	0.093750	10.7	1
Termination	0.093750	10.7	1

USAGE AND TESTING CHAINS

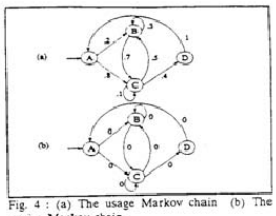

Fig. 4 : (a) The usage Markov chain (b) The testing Markov chain

PATH FREQUENCY MEASUREMENT

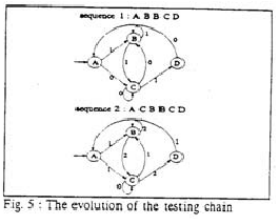

Fig. 5 : The evolution of the testing chain

DEALING WITH FAILURES

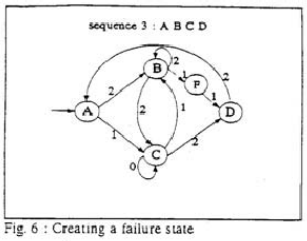

Fig. 6 : Creating a failure state

TESTING MODEL STATISTICS

Table IV : Analytical Results for the Example Usage Model

Sequence #	R	MTBF	D(1/F)
1	1	–	–
2	1	–	–
3 (failure 1)	0.750000	47.000000	–
4	0.800000	58.000000	0.051781
5 (failure 2)	0.784286	54.500000	0.066327
6	0.750000	48.000000	0.064144
7	0.777778	49.000000	0.038756
8	0.800000	44.000000	0.033571
9	0.818182	46.000000	0.050839
10	0.833333	56.500000	0.052332
11	0.846154	58.500000	0.050483
12	0.857143	68.500000	0.052500
13	0.866667	70.500000	0.051166
14	0.875000	72.500000	0.050313
15 (failure 3)	0.833529	50.000000	0.019183
16	0.833333	52.333333	0.059069
17	0.842105	54.666667	0.060906
18	0.850000	56.000000	0.060327
19	0.857143	59.000000	0.062373
20	0.863636	64.000000	0.0311877
.			
25	0.888889	82.000000	0.026581
.			
50	0.906250	118.333333	0.023458
.			
35	0.918919	134.333333	0.022276
.			
40	0.928571	158.333333	0.019938

References:

- http://www.sei.cmu.edu/str/descriptions/cleanroom.html
- http://www.embedded.com/97/feat9609.htm
- http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr022.96.pdf
- http://www.uta.edu/cse/levine/fall99/cse5324/cr/clean/page.html
- http://csdl2.computer.org/comp/proceedings/hicss/1998/8248/06/82480122.pdf
- http://www.stsc.hill.af.mil/crosstalk/1996/11/xt96d11f.asp