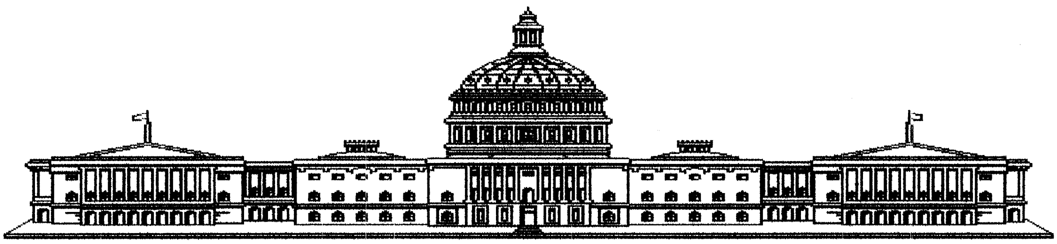


NATIONAL SOFTWARE QUALITY EXPERIMENT
A LESSON IN MEASUREMENT
1992-1997

Don O'Neill
Independent Consultant
(301) 990-0377

http://members.aol.com/ONeillDon/index.html

Experiment Purpose

Don O'Neill Consulting

To measure progress towards the national objective

Reduce software problems by a factor of 10 by the year 2000

Set by the DOD Software Technology Strategy in 1992

To benchmark the state of software product quality

Some of the Questions Asked and Answered in the Experiment

Don O'Neill Consulting

To what extent is there a continuing stream of requirements changes?

What are the leading types of errors?

Are errors traced to people or process?

Is a standard development process followed?

To what extent are wrong software functions being developed?

To what extent are there shortfalls in real time performance?

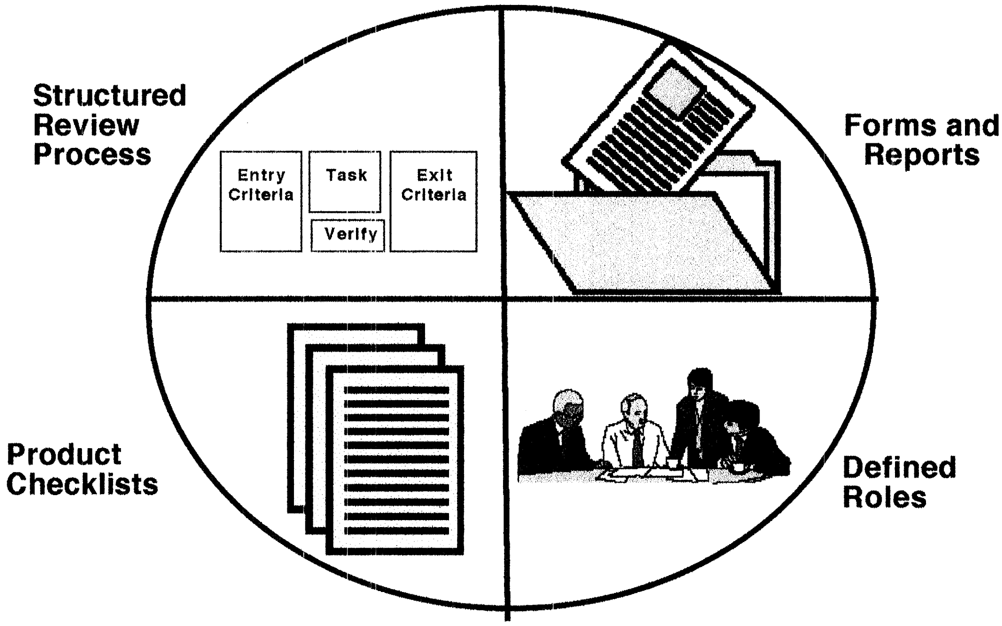
Is gold plating a problem?

Experiment Participants

Don O'Neill Consulting

- Accounting, personnel, administration
- Administrative and management decision support
- Aircraft jet engine diagnostics, logistics, and maintenance
- Artillery fire control system
- Avionics flight on-board control
- Control devices for avionics applications
- Credit card application
- Department of State embassy support
- Electronic commerce
- Electronic warfare
- FAA communications
- Factory line support
- Financial services
- Global positioning system user sets
- Insurance and medical information
- International banking
- Joint Chiefs of Staff support
- Medical information system
- Naval surface weapons system
- Pre and post flight space application
- Telecommunications

Experiment Centerpiece: Inspection Lab



Product Checklist Themes

Completeness
Traceability from code to requirements

Correctness
Intended function with faithful elaboration of steps that carry it out

Style
Naming, commentary, alignment, case, highlighting, templates

Rules of Construction
Application domain specific reference architecture

Multiple Views
Programmer, tester, user, computer resources, security, Y2K

Inspection Reporting Form

Issue Number	Page/ Line	Checklist	Defect Category	Defect Severity	Defect Type	Defect Origin	Defect Description

Defect Category: Missing, Wrong, Extra
Defect Severity: Major, Minor

Defect Type: Interface, Data, Logic, I/O, Performance, Functionality, Human Factors, Standards, Documentation, Syntax, Maintainability, Other

Defect Severity and Category Summary

Don O'Neill Consulting

Defect Severity and Category Summary

	Missing	Wrong	Extra	Total
Major	7.44	5.95	.88	14.27
Minor	49.76	27.63	8.32	85.73
Total	57.20	33.60	9.20	100.00

Inspection Lab Operations Summary

Don O'Neill Consulting

INSPECTION LAB OPERATIONS

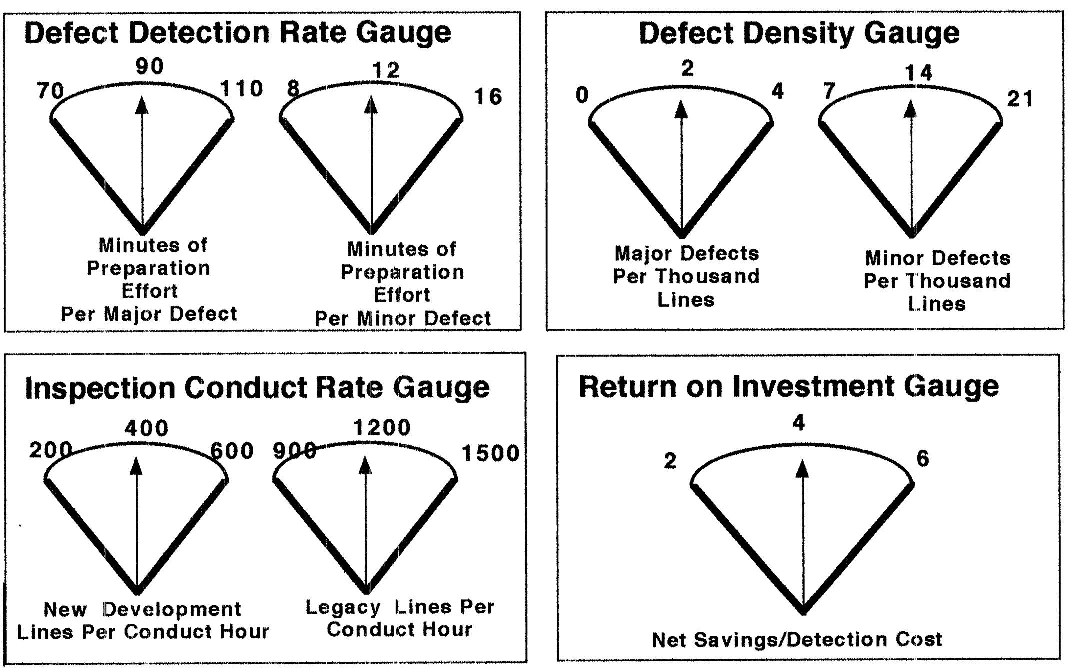
Sessions	Prep Effort	Conduct Time	Major Defects	Minor Defects	Size in Lines
2317	142,306	52,196	1854	9521	788,459

Metrics:

1.	12.51	minutes of preparation effort per defect
2.	76.76	minutes of preparation effort per major defect
3.	2.35	major defects per KSLOC
4.	12.08	minor defects per KSLOC
5.	906	lines per conduct hour
6.	4.91	Defects per session
7.	4.48	Return on Investment

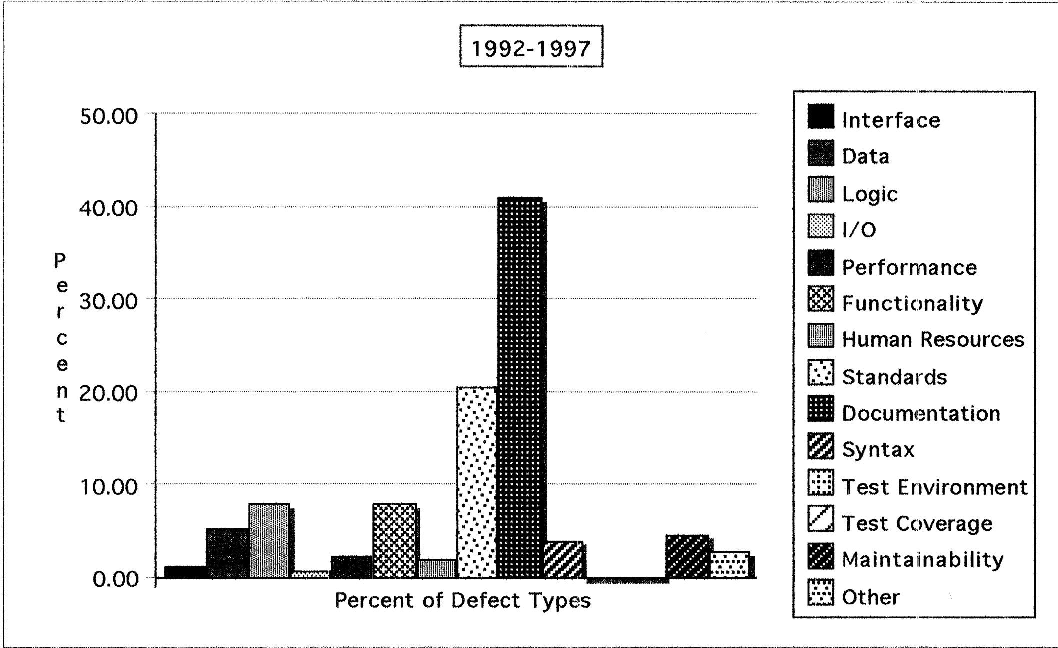
Software Inspections Control Panel

Don O'Neill Consulting



Defect Types

Don O'Neill Consulting



Common Problems

Don O'Neill Consulting

1. Software product source code components are not traced to requirements.

As a result, the software product is not under intellectual control, verification procedures are imprecise, and changes cannot be managed.

2. Software engineering practices for systematic design and structured programming are applied without sufficient rigor and discipline.

As a result, high defect rates are experienced in logic, data, interfaces, and functionality.

3. Software product designs and source code are recorded in an ad hoc style.

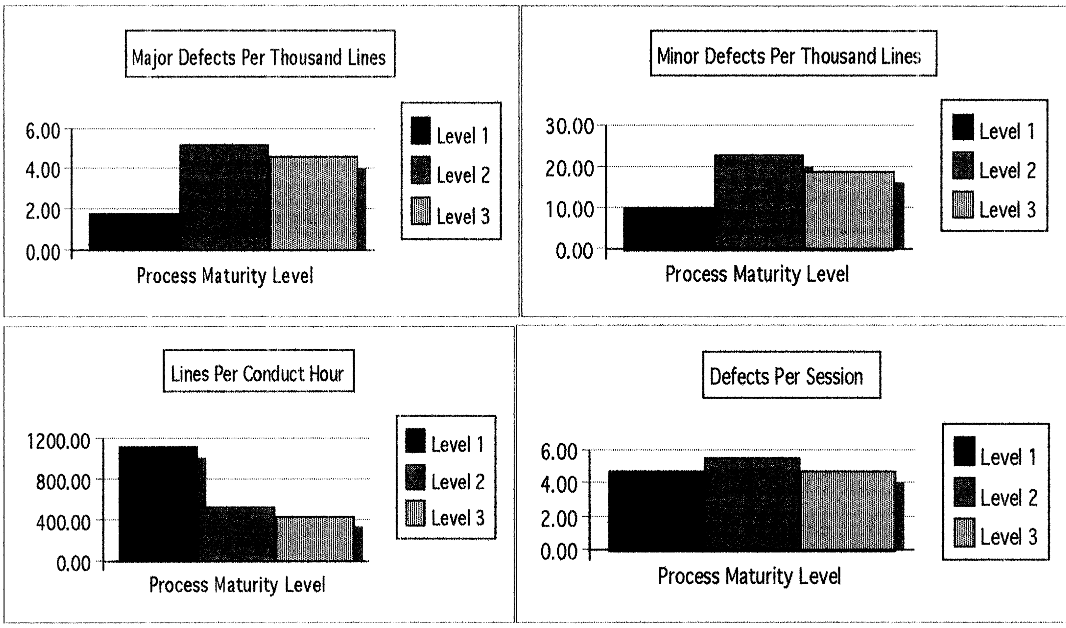
As a result, the understandability, adaptability, and maintainability of the software product are directly impacted.

4. The rules of construction for the application domain are not clearly stated, understood, and applied.

As a result, common patterns and templates are not exploited in preparation for later reuse.

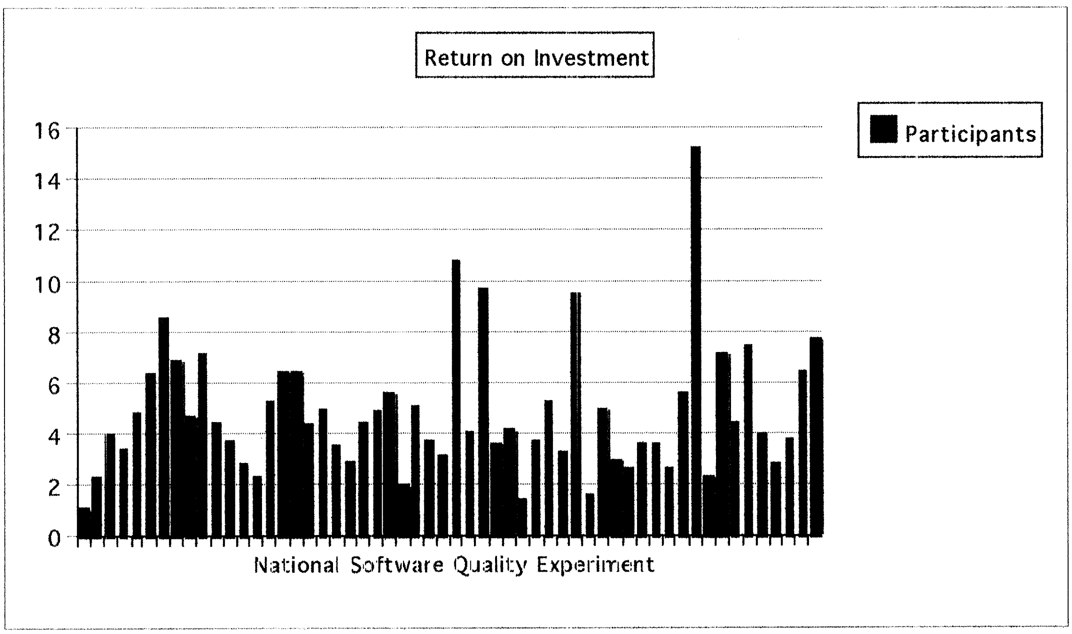
Software Process Maturity Level

Don O'Neill Consulting



Return on Investment

Don O'Neill Consulting



Experiment Findings Summary

Don O'Neill Consulting

Lack of Progress

-The objective to reduce software problems by a factor of 10 is not being met

Looking Harder, Finding More

-By reducing the size of artifacts inspected

Program Size Matters

-Defect density decreases with program size
-Starting, finishing, and fitting in are all more error prone than the body of the program which gives it size

Software Process Maturity Insight

-Legacy software anchors many organizations at level 1
-These are often commercial enterprises

Process Neglect Exceeds Personal Defects

-Organization neglect of its software process exceeds the poor workmanship of individual programmers as the source of errors
-Documentation and standards defect types account for nearly two-thirds of all defects

Return on Investment High

-Software inspections deliver a favorable return on investment with
-Savings exceed costs by 4 to 1

Field Measurement Lessons

Don O'Neill Consulting

1. Measurement must be aligned with business and performance needs.

These activities must be built into the normal operation of the organization.

To do this, the goals to be met and questions to be answered in management, engineering, and operations must precede the collection of data.

2. Metrics must be carefully pinpointed and rigorously defined.

Extraordinary steps must be applied to obtain consistency and uniformity.

Without a well defined process for data collection and analysis, the variance in the measurement process itself impacts the accuracy of results.

3. Attention must be paid to the confidentiality of results.

The opportunity for improvement is increased when the measured results are made more widely available.

-However, individuals and groups naturally resist having their shortcomings made public.
-If ignored, this resistance will defeat the measurement program.
-The organization must strike a balance between public and private data.

NATIONAL SOFTWARE QUALITY EXPERIMENT
A LESSON IN MEASUREMENT

PROLOGUE

The nation's prosperity is dependent on software. The nation's software industry is slipping, and it is slipping behind other countries. The National Software Quality Experiment is riveting attention on software product quality and revealing the patterns of neglect in the nation's software infrastructure.

ABSTRACT

In 1992 the DOD Software Technology Strategy set the objective to reduce software problem rates by a factor of ten by the year 2000. The National Software Quality Experiment is being conducted[1] to benchmark the state of software product quality and to measure progress towards the national objective.

The National Software Quality Experiment is a mechanism for obtaining core samples of software product quality. A micro-level national database of product quality is being populated by a continuous stream of samples from industry, government, and military services. This national database provides the means to benchmark and measure progress towards the national software quality objective and contains data from 1992 through 1997.

The centerpiece of the experiment is the Software Inspection Lab where data collection procedures, product checklists, and participant behaviors are packaged for operational project use. The uniform application of the experiment and the collection of consistent measurements are guaranteed through rigorous training of each participant. Thousands of participants from dozens of organizations are populating the experiment database with thousands of defects of all types along with pertinent information needed to pinpoint their root causes.

To fully understand the findings of the National Software Quality Experiment, the measurements taken in the lab and the derived metrics are organized along several dimensions including year, software process maturity level, organization type, product type, programming language, global region, and industry type. These dimensions provide a framework for populating an interesting set of analysis bins with appropriate core samples of software product quality.

[1] The National Software Quality Experiment is an entrepreneurial activity

Author: Don O'Neill

Don O'Neill is a seasoned software engineering manager and technologist currently serving as an independent consultant. Following his twenty-seven year career with IBM's Federal Systems Division, Mr. O'Neill completed a three year residency at Carnegie Mellon University's Software Engineering Institute (SEI) under IBM's Technical Academic Career Program. There he developed a blueprint for charting software engineering evolution in the organization including the training architecture and change management strategy needed to transition skills into practice.

As an independent consultant, Mr. O'Neill conducts defined programs for managing strategic software improvement. These include implementing an organizational Software Inspections Process, implementing Software Risk Management, and conducting Global Software Competitiveness Assessments. Each of these programs includes the necessary practitioner and management training.

In his IBM career, Mr. O'Neill completed assignments in management, technical performance, and marketing in a broad range of applications including space systems, submarine systems, military command and control systems, communications systems, and management decision support systems. He was awarded IBM's Outstanding Contribution Award three times:

1. Software Development Manager for the Global Positioning Ground Segment (500,000 source lines of code) and a team of 70 software engineers within a $150M fixed price program.
2. Manager of the FSD Software Engineering Department responsible for the origination of division software engineering strategies, the preparation of software management and engineering practices, and the coordination of these practices throughout the division's software practitioners and managers.
3. Manager of Data Processing for the Trident Submarine Command and Control System Engineering and Integration Project responsible for architecture selections and software development planning (1.2M source lines of code).

Mr. O'Neill served on the Executive Board of the IEEE Software Engineering Technical Committee and as a Distinguished Visitor of the IEEE. He is a founding member of the National Software Council and the Washington DC Software Process Improvement Network (SPIN). He is an active speaker on software engineering topics and has served as the Program Chairman and Program Committee member for several conferences. He has numerous publications to his credit. Mr. O'Neill has a Bachelor of Science degree in mathematics from Dickinson College in Carlisle, Pennsylvania.