



## Experimentation in Software Engineering: Reading Studies



### The Software Engineering Discipline

---



Software techniques, methods, models. etc. need to be  
validated via experimentation  
refined and tailored to the application environment  
logically or physically integrated  
be easily transferred into practice

There is a need to  
understand the relationships between process and product  
learn and evolve our knowledge based upon experience

We need an experimental, evolutionary software development  
framework  
that deals with the symbiotic relationship between research and  
development  
so that learning can take place in a practical way

Experimentation can take many forms



## Research Paradigms



Need research to establish a scientific and engineering basis for software engineering

Required research methods involve the need to build, analyze, and evaluate models of the software processes and products various aspects of the environment, e.g the people, the organization and the interactions of these models.

The goal is to develop the conceptual scientific foundations of software engineering upon which future researchers can build.

This is often a process of  
discovering and validating small but important concepts  
that can be applied in many different ways and  
that can be used to build more complex and advanced ideas  
rather than merely providing a tool or methodology without  
experimental validation of its underlying assumptions or  
careful analysis and verification of its properties



## Research Paradigms Definitions



A **fact** is information obtained through direct observation

A **hypothesis** is an educated guess that precedes an experiment

An **experiment** is

- a test, trial or tentative procedure policy;
- an act or operation for the purpose of discovering something unknown or of testing a principle, supposition, etc.;
- an operation carried out under controlled conditions in order to discover an unknown effect or law, to test or establish a hypothesis, or to illustrate a known law



## Research Paradigms Definitions



A **theory** is a possible explanation based upon many facts and reason

A **law** is a description/observation of behavior used for prediction based upon facts and reason

A **model** is a simplified representation of a system or phenomenon with any hypotheses required to describe the system or explain the phenomenon, often mathematically. A model can be a theory or a law

A **paradigm** is conceptual filter that determines how we perceive/interpret

A **truth** is what really is



## Research Paradigms



Research methods from other disciplines include various forms of experimental or analytic paradigms

Experimental paradigms require an experimental design, observation, data collection and validation on the process or product being studied

The Scientific Method:  
observe the world,  
propose a model or a theory of behavior,  
measure and analyze,  
validate hypotheses of the model or theory,  
and, if possible, repeat the procedure.



## Research Paradigms

---



### The Scientific Method

is an inductive paradigm which can be used to:  
understand the software process, product, people, environment  
extract models from the world that explain underlying phenomena, and  
evaluate if the model is representative of the phenomenon observed

Example: an attempt to understand the way software is developed by an organization to see if their process model can be abstracted or a tool can be built to automate the process

There are two variations of the inductive paradigm which we will call the  
engineering method  
empirical method



## Research Paradigms

---



### The Engineering Method:

observe existing solutions,  
propose better solutions,  
build/develop,  
measure and analyze, and  
repeat the process until no more improvements appear possible.

This version of the paradigm is

an evolutionary improvement oriented approach  
assumes models exist  
modifies model to improve the thing being studied

Example: study improvements to methods or demonstrate that a tool performs better than its predecessor relative to certain characteristics



## Research Paradigms

---



The Empirical Method:

propose a model,  
develop statistical/qualitative methods,  
apply to case studies,  
measure and analyze,  
validate the model and repeat the procedure.

This version of the paradigm is

a revolutionary improvement oriented approach  
proposes a new model  
studies effects of process or product suggested by the new model

Example: proposal of a new method or tool and validation that the model or tool is an advance over current models or tools



## Research Paradigms

---



There must be a rationale for collecting data.

Experiments must be designed to acquire information useful for the building of a suitable description (model or theory) of the systems under study. It is an approach to model/theory/law building.

Experimentation alone is of no value if there is no underlying framework or context where experimental results can be interpreted.

Other issues involved in these inductive, experimental methods include

- the types of experimental design appropriate for different environments,
- whether the experiment is exploratory or confirmatory,
- the validity of the data collected,
- the cost of the experiment,
- the problems of reproducibility, etc.



## Research Paradigms



An analytic paradigm is:

The Mathematical Method:

propose a formal theory or set of axioms,  
develop a theory,  
derive results, and  
if possible compare with empirical observations.

This is a deductive analytical model which

does not require experimental design in the statistical sense, but  
provides a framework for  
developing models and understanding their boundaries  
based upon manipulation of model itself

Example: the treatment of programs as mathematical objects and the  
analysis of the mathematical object or its relationship to the program



## Research Paradigms



These paradigms serve as a basis for distinguishing research activities  
from development activities

If one of these paradigms is not being used in some form, the study is  
most likely not a research project

Many projects that claim to be research are simply developments, e.g.,  
building a system or tool alone is development, not research

Research involves gaining understanding about how and why a certain  
type of tool might be useful, and by validating that a tool has certain  
properties or effects by carefully designing an experiment to measure  
the properties or to compare it with alternatives

The scientific method can be used to understand the effects of a particular  
tool in some environment and to validate hypotheses about how  
software development can best be accomplished



## The Experimental Discipline



### Experimental Classifications

#### Level of variable relationship

**Descriptive:** there may be patterns in the data but the relationship among the variables has not been examined

**Correlational:** the variation in the dependent variable(s) is related to the variation of the independent variable (s)

**Cause-effect:** the treatment variable(s) is the only possible cause of variation in the dependent variable(s)



## The Experimental Discipline



### Experimental Classifications

#### Experience of Subjects

**novice:** students or individuals not experienced in domain

**experts:** practitioners or people with experience in domain

#### Experimental Setting

**In vivo:** in the field under normal conditions

**In vitro:** in the laboratory under controlled conditions

#### Type of Study

**Experiment:** at least one treatment or controlled variable

**Observational study:** no treatment or controlled variables



## The Experimental Discipline



### Experimental Classifications

#### Types of Analysis

##### Quantitative Analysis

- obtrusive controlled measurement
- objective
- verification oriented

##### Qualitative Analysis

- naturalistic and uncontrolled observation
- subjective
- discovery oriented



## The Experimental Discipline



### Experimental Classifications

#### Study

- an act to discover something unknown or of testing a hypothesis
- can include all forms of quantitative and qualitative analysis

#### Studies can be

- **experimental**
  - driven by hypotheses; quantitative analysis
  - controlled experiments
  - quasi-experiments or pre-experimental designs
- **observational**
  - driven by understanding; qualitative analysis dominates
  - qualitative/quantitative study
  - pure qualitative study



## The Experimental Discipline



### Experimental Study Classifications

**Experiments** can be

- controlled experiments
- quasi-experiments or pre-experimental designs

**Controlled experiments**, typically:

- small object of study
- in vitro
- a mix of both novices (mostly) and expert treatments

Sometimes, novice subjects used to “debug” the experimental design

**Quasi-experiments or Pre-experimental design**, typically:

- large projects
- in vivo
- with experts

These latter experiments tend to involve a qualitative analysis component, including at least some form of interviewing



## Experimental and Quasi-Experimental Designs



Experimentation is not a panacea, but rather the only available route to cumulative progress

There are a large variety of experimental and quasi-experimental designs

These are represented in what follows, using the notation:

Let X represent the exposure of a group to an experimental variable or event, the effects of which are to be measured

Let O refer to some process of observation or measurement

Assume the X's and O's in the same line are given to the same specific persons

Let R represent the random assignment to separate groups

– Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## Factors Jeopardizing Validity



- There are several factors that can jeopardize the validity of an experimental design
- They can be broken into **internal** and **external** validity
- Internal validity is the basic minimum without which an experiment is uninterpretable
  - Did in fact the experimental treatments make any difference in this specific experimental instance?
- External validity deals with the issue of generalizability
  - To what populations, settings, treatment variables, and measurement variables can this effect be generalized?

– Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## Internal Validity



Eight different classes of extraneous variables, which, if not controlled in the experimental design, might produce effects confounded with the effect of the experimental stimulus.

- History - the specific events occurring between the first and second measurement in addition to the experimental value, creating rival hypotheses (O1 X O2)
- Maturation - processes within the respondents operating as a function of the passage of time per se (not specific to the particular events), including growing older, hungrier, more tired, etc.
- Testing - the effects of taking a first test upon the scores of a second testing
- Instrumentation - changes in the calibration of a measuring instrument or changes in the observers or scorers used, may produce changes in the obtained measurements.

– Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## Internal Validity



- Statistical Regression - operating where groups have been selected on the basis of their extreme scores, i.e., tendency toward the mean
- Selection - biases resulting in differential selection of respondents for the comparison groups (X O1, O2)
- Experimental Mortality - differential loss of respondents from the comparison groups
- Selection-Maturation Interaction, etc. - any of the extraneous variables can have a combined effect that can be mistaken for the effect of the experimental variable

- Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## External Validity



The factors jeopardizing external validity or representativeness are:

- **Testing and X: Reactive/Interaction Effect of Testing** - a pretest might increase or decrease the respondent's sensitivity or responsiveness to the experimental variable, thus making the results obtained for a pre-tested population unrepresentative of the effects of the experimental variable for the unpretested universe from which the experimental respondents were selected
- **Selection and X: Interaction** effects of selection biases and the experimental variable

- Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## External Validity



- Reactive Effects of Experimental Arrangements - preclude generalization about the effect of the experimental variable upon persons being exposed to it in non-experimental settings
- Multi-Treatment Interference - likely to occur whenever multiple treatments are applied to the same respondents, because the effects of prior treatments are not usually erasable. This is a particular problem for one-group designs of type 8 or 9.
  - Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## Pre-Experimental Designs



### Design 1: The One Shot Case Study

X O

Absence of control, almost no scientific value, opportunity for qualitative analysis or technique evolution

Rival Hypotheses: history, maturation, selection, mortality,

### Design 2: The One Group Pretest Posttest design

O<sub>1</sub> X O<sub>2</sub>

Rival hypotheses: history, maturation, testing, instrumentation, statistical regression (?),

### Design 3: The Static Group Comparison

X O<sub>1</sub>

O<sub>2</sub>

Rival hypotheses: selection, mortality,

- Campbell & Stanley,
- Experimental and Quasi-experimental Designs for Research



## True Experimental Designs



### Design 4: The pretest post test Control Group design

R O<sub>1</sub> X O<sub>2</sub>  
R O<sub>3</sub> O<sub>4</sub>

### Design 5: The Solomon Four group design

R O<sub>1</sub> X O<sub>2</sub>  
R O<sub>3</sub> O<sub>4</sub>  
R X O<sub>5</sub>  
R O<sub>6</sub>

### Design 6: Posttest Only Control Group Design

R X O<sub>1</sub>  
R O<sub>2</sub>

– Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## True Experimental Designs



### Factorial Designs: Several treatments (ala Design 6)

R X<sub>1</sub> O<sub>1</sub>  
R X<sub>2</sub> O<sub>2</sub>  
R X<sub>3</sub> O<sub>3</sub>  
...  
R X<sub>n</sub> O<sub>n</sub>

Can be done with Design 4 and 5 also  
Can be done with a control group as well

– Campbell & Stanley, Experimental and Quasi-experimental Designs for Research



## Quasi-Experimental Designs



When the experimenter lacks full control over the scheduling of experimental stimuli, something like an experimental design can be introduced

### Time Series Design

$O_1 O_2 O_3 O_4 X O_5 O_6 O_7 O_8$

### Equivalent Time Samples Design

$X_1 O_1, X_2 O_2, X_1 O_3, X_2 O_4, \dots$

### Non-Equivalent Control Group Design

$O X O$   
 $O O$

Campbell & Stanley,  
Experimental and Quasi-experimental Designs for Research



## The Experimental Discipline



### How do we combine experiments?

There are several different approaches to experimenting in the software domain

The approaches vary in

- Level of variable relationship
- Level of confidence in results, insights gained
- Experience of subjects
- Environmental setting
- Balance between quantitative/qualitative research
- Cost



## The Experimental Discipline



### Classes of Experimental Studies

#### Experiment Classes

		#Projects	
		One	More than one
# of Teams	One	Single Project	Multi-Project Variation
per Project	More than one	Replicated Project	Blocked Subject-Project



## The Experimental Discipline



### How do we combine experiments?

**Controlled experiments**, typically:

costs high, projects must be small, but better basis for quantitative analysis and generation of stronger statistical confidence in the conclusions

**Quasi-experiments or Pre-experimental design**, typically:

costs reasonable, projects can be large, and better able to simulate the effects of the treatment variables in a realistic environment

Larger projects tend to involve more qualitative analysis along with some more primitive quantitative analysis

One approach to experimentation is to combine treatments:

- one controlled experiment treatment to demonstrate feasibility in the small
- one quasi-experiment treatments to analyze if the results scale up
  - a major problem in software engineering research



## The Experimental Discipline



### Experimental Study Classifications

#### Observational studies

- qualitative/quantitative study
- pure qualitative study

**Qualitative/quantitative analysis:** observer has identified, a priori, a set of variables for observation

There are a large number of case studies and some field studies

- in vivo
- descriptive
- experts

**Pure qualitative analysis:** no variables isolated a priori, open observation

- deductions made using non-mathematical formal logic  
e.g., verbal propositions

Found only one pure qualitative study, a Field Qualitative Study, in vivo, descriptive, experts



## The Experimental Discipline



### Classes of Observational Studies

#### Observational Studies

		Variable Scopes	
		A priori defined variables	No a priori defined variables
# of Sites	One	Case Study	Case Qualitative Study
	More than One	Field Study	Field Qualitative Study



## The Experimental Discipline



Sign of maturity in a field:

**level of sophistication** of the goals of an experiment  
**understanding interesting things** about the discipline

For software engineering that might mean:

Can we build models that allow use to measure and differentiate processes and products?

Can we measure the effect of a change in a particular process variable on the product variable?

Can we predict the characteristics of a product (values of product variable) based upon the model of the process (values of the process variables), within a particular context?

Can we control for product effects, based upon goals, given a particular set of context variables?



## The Experimental Discipline



Sign of maturity in a field:

a **pattern of knowledge** built from a **series of experiments**

Does the discipline build on prior (knowledge, models, experiments).

Was the study an isolated event?

Did it lead to other studies that made use of the information obtained from it

Have studies been replicated under similar or differing conditions?

Does the building of knowledge exist in one research group or environment, or has it spread to others - researchers building on each other's experimental work?

For example, inspections, in general, are well studied experimentally

However, there has been very little combining of results, replication, analysis of the differentiating variables



## Studying Process Effects



### Methods and Techniques

A **technique** is a technical procedure for constructing or assessing software, that requires skill, and produces a technical result, e.g., reading, testing

A **method** is a management procedure for applying software techniques, with a set of rules stating how and when to apply and when to start and stop applying the technique (entry and exit criteria), which technique is appropriate, and how to evaluate it (management support), e.g., design inspections, test plans.

We need to understand  
the relationship between techniques and methods  
the dimensions of both  
how to improve them for a particular environment



## Reading Techniques



Reading is a **key technical activity** for analyzing and constructing software artifacts

Reading is **a model for writing**

Reading is **critical for reviews, maintenance, reuse, ...**

What is a reading technique?

a concrete set of instructions given to the reader saying how to read and what to look for in a software product

More Specifically, software reading is

**the individual analysis of a software artifact**

e.g., requirements, design, code, test plans

**to achieve the understanding needed for a particular task**

e.g., defect detection, reuse, maintenance



## Dimensions of a Reading Technique



- Input object: Requirements, specification, design, code, test plan,...
- Output object: Set of anomalies
- Approach: Sequential, path analysis, stepwise abstraction,...
- Formality: Reading, correctness demonstrations,...
- Emphasis: Fault detection, traceability, performance,...
- Method: Walk-throughs, inspections, reviews,...
- Consumers: User, designer, tester, maintainer,...
- Product qualities: Correctness, reliability, efficiency, portability,...
- Process qualities: Adherence to method, integration into process,...
- Quality view: Assurance, control,...



## Reading Techniques



Early experiments (Hetzel, Meyers) showed very little difference between reading and testing

But reading was simply reading, without a technological base

We discuss a series of experiments at the University of Maryland and at NASA used to learn about, evaluate, and evolve reading techniques

This example

- shows **multiple experimental designs**
- provides a combination of **evaluation approaches**
- offers insight into the **effects of different variables** on reading

The experiments start with the early reading vs. testing experiments to various Cleanroom experiments to the scenario based reading techniques currently under study



## EXPERIMENTAL LEARNING MECHANISMS



### Series of Studies

		# Projects	
		One	More than one
# of Teams per Project	One	<b>3. Cleanroom (SEL Project 1)</b>	<b>4. Cleanroom (SEL Projects, 2,3,4,...)</b>
	More than one	<b>2. Cleanroom at Maryland</b>	<b>1. Reading vs. Testing 5. Scenario reading vs. ...</b>



## EVALUATION OF A PROCESS



When introducing any form of process, method or tool, the organization needs to evaluate its effectiveness

That effectiveness of a process can be measured by

- higher than normal quality
- cheaper development costs
- improved cycle time to delivery
- improved product functionality
- more predictable behavior

...

It is important to understand the relationship between the process and the product

It is important to have a data as a basis of comparison



## Blocked Subject Project Study



### Testing/Reading Strategies Comparison

**Goals:**

Analyze code reading, functional testing and structural testing to evaluate and compare them with respect to their effect on fault detection effectiveness, fault detection cost and classes of faults detected from the viewpoint of quality assurance

**Environment:**

NASA/CSC and the University of Maryland  
Text formatter, plotter, abstract data type, database  
Seeded with software faults (9, 6, 7, 12)  
145 - 365 LOC

**Experimental design:**

Blocked subject-project: Fractional factorial design  
Three applications  
74 subjects: 32 NASA/CSC, 42 UM



## Blocked Subject Project Study Testing/Reading Strategies Comparison



### Technique Definition

**Code Reading:** Reading by Stepwise Abstraction

read a sequence of statements and abstract the function they compute repeat until the function of the entire program has been abstracted and can be compared with the specification

**Functional Testing:** Boundary Value Equivalence Partition Testing

divide the requirements into valid and in valid equivalence classes and make up tests that check the boundaries of the classes

**Structural Testing:** Achieving 100% statement coverage

make up a set of tests that guarantee that 100% of the statement in the program have been executed



## Blocked Subject Project Study Testing/Reading Strategies Comparison



### Fractional Factorial Design

Each Subject applies each of the treatments (techniques) on a project

The techniques are the **independent** variable (the item being studied)

The effects on the product (number and type of faults identified, time to identify the faults, ...) are the **dependent** variables

The **context** variables are the elements of the environment being studied, e.g., the experience of the people applying the techniques

The design allow us to minimize the effect of the individual

- Each subject uses each technique and tests each program
- We can block according to experience level and program tested
  - this allows us to identify the effects of each of those variables



## Blocked Subject Project Study Testing/Reading Strategies Comparison



### Fractional Factorial Design

		Code Reading			Functional Testing			Structural Testing			
		P1	P2	P3	P1	P2	P3	P1	P2	P3	
Advanced Subjects	S1			X		X		X			
	S2		X		X					X	
	⋮										
		S8	X			X			X		
		<hr/>									
Intermediate Subjects	S9			X		X		X			
	S10		X		X					X	
	⋮										
		S19	X			X			X		
		<hr/>									
Junior Subjects	S20			X		X		X			
	S21		X		X					X	
	⋮										
		S32	X			X			X		

Blocking according to experience level and program tested  
Each subject uses each technique and tests each program

NASA/CSC



## Blocked Subject Project Study Testing/Reading Strategies Comparison



### Major Conclusions (NASA/CSC)

#### Fault Detection Effectiveness

4.0 Faults found on average (SD = 1.9, 50.0%)  
Code reading > (functional > structural)

#### Fault Detection Rate

1.82 faults/hour average (SD = 1.80)  
Code reading > (functional ≈ structural)

#### Total Fault Detection Time

3.3 hours testing on average (SD = 2.19)  
No difference in techniques

#### Classes of Faults Detected

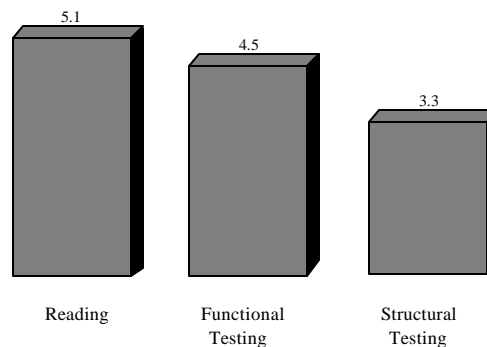
Omission: (code reading ≈ functional) > structural  
Initialization: (code reading ≈ functional) > structural  
Interface: code reading > (functional ≈ structural)  
Computation: code reading > structural  
Control: functional > (code reading ≈ structural)



## Blocked Subject Project Study Testing/Reading Strategies Comparison



### Major Conclusions (NASA/CSC) Fault Detection Effectiveness (Mean)

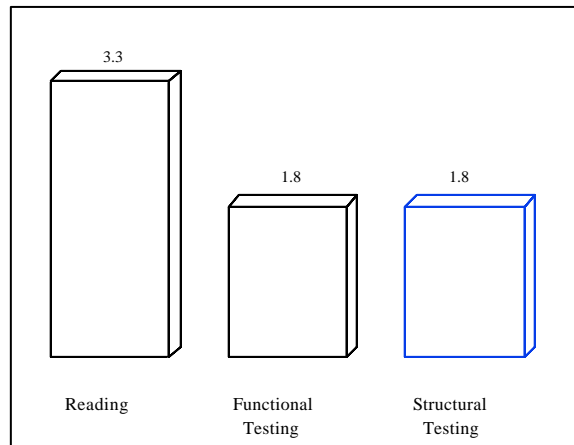




## Blocked Subject Project Study Testing/Reading Strategies Comparison



### Major Conclusions (NASA/CSC) Fault Detection Rate (Faults/hour)



## Blocked Subject Project Study Testing/Reading Strategies Comparison



### Major Conclusions

**Self-estimates during study:**

code reading > structural > functional

**After completion of study:**

over 90% of the participants thought functional testing worked best

**Based upon this study**

reading was implemented as part of the SEL development process

**But -** reading appeared to have very little effect

**Hypothesis 1:** People did not read as well as they should have as they believed that testing would make up for their mistakes

**Experiment:** If you read and cannot test you do a more effective job of reading than if you read and know you can test.

**Hypothesis 2:** There is a confusion between reading and the method in which it is embedded, e.g., inspections



## Blocked Subject Project Study Testing/Reading Strategies Comparison

---



### Lessons Learned

Reading using a particular technique is more effective and cost effective than specific testing techniques

**The reading technique is important**

**but**

**Different approaches may be more effective for different types of defects**

Reader needs to be motivated to read better

**The reading motivation is important**

We may need to better support the reading process

**The reading technique may be different from the reading method**

The Cleanroom approach seemed to cover a couple of these issues so we tried a controlled experiment at the University of Maryland



## Reading-Based Life Cycle Model Cleanroom Process

---



Key components:

- Mathematically-based design methodology
  - Function specification for programs
  - State machine specification for modules
  - Reading by stepwise abstraction
  - Correctness demonstrations when needed
- Top-down development

Implementation without any on-line testing by developer

Independent testing

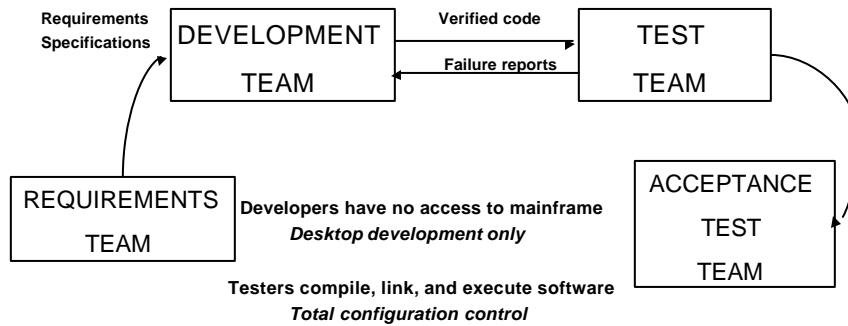
- Statistically based on anticipated operational use
- Quality assurance orientation



## EVALUATION OF A PROCESS Process Definition



### Cleanroom Process



## PROCESS MODEL DEFINITION



### Cleanroom Process

#### Mathematically-based design methodology

Function specification for programs

function:  $[f] = f$   
 sequence:  $[f] = [g;h] = \{(x,y) \mid y = h(g(x))\}$   
 if then:  $[f] = [\text{if } p \text{ then } g] = \{(x,y) \mid (p(x) = \text{True and } y = g(x)) \text{ or } p(x) = \text{False and } y = h(x))\}$   
 if then else:  $[f] = [\text{if } p \text{ then } g \text{ else } h] = \{(x,y) \mid (p(x) = \text{True and } y = g(x)) \text{ or } (p(x) = \text{False and } y = h(x))\}$   
 while do:  $[f] = [\text{while } p \text{ do } g] = [\text{if } p \text{ then } g; f]$



## PROCESS MODEL DEFINITION



### Cleanroom Process

#### Mathematically-based design methodology

##### State machine specification for modules

State machine,  $m$ , a function from an ordered pair of inputs into ordered pair of outputs  
 $(\text{newstate}, \text{output}) = m(\text{current state}, \text{input})$

Interpretation: after machine has operated, the newstate becomes the oldstate for next operation.

User: Only inputs & outputs "observable", so operation may appear non-functional.

Designer: Can observe the state data, so the behavior is completely functional.

A module state machine is the state machine defined by a module

- the input variables and their types define the inputs
- the output variables and their types define the outputs
- the retained variables and their types
- the text of the module programs define the transition rules
- a distinguished input may cause a transition from any state to a known initial state, or the presence of initial data may create an initial state

## PROCESS MODEL DEFINITION

### Cleanroom Process

#### Mathematically-based design methodology

Reading by stepwise abstraction

##### **sequence:**

Read the successive parts and summarize their sequential effect in the final outcome

##### **ifthenelse:**

Read the if test true and the then part, then read the if test false and the else part, and summarize the two outcomes into a single outcome

##### **whiledo:**

Read the while test true and the do part (be sure to verify that the whilettest eventually evaluates to false), then read the whilettest false (no operation), and summarize the outcome at exit





## PROCESS MODEL DEFINITION



### Cleanroom Process

#### Mathematically-based design methodology

Correctness demonstrations when needed

Correctness demonstrations are done, when needed, by building trace tables of the function variables and doing symbolic execution

Because a state machine is a special type of function, we can use the function methodology in the design and verification of modules



## Replicated Project Study



### Cleanroom Study

#### Study Goal:

Analyze the Cleanroom process  
in order to evaluate and compare it to a non-Cleanroom process  
with respect to the effects on the process, product and developers  
from the viewpoint of quality assurance

#### Environment:

University of Maryland  
Electronic message system, ~ 1500 LOC

#### Experimental design:

Replicated project, 15 three-person teams (10 used Cleanroom)  
3 to 5 test submissions

Data collected: Background,  
Attitude survey  
On-line activity  
Testing results



## Replicated Project Study Cleanroom Evaluation



### Major Results

#### Effect on the Software Development Process

##### Cleanroom developers

felt they more effectively applied off-line review techniques, while others focused on functional testing  
spent less time on-line and used fewer computer resources  
tended to make all their scheduled deliveries

#### Effect on the Delivered Product

Static properties: less dense complexity, higher percentage of assignment statements, more global data, more comments

Operational properties: product more completely met requirements, higher percentage of test cases succeeded

#### Effect on Developers

Missed program execution  
Modified their development style  
Would use it again



## Cleanroom in the SEL



### Using the QIP for Cleanroom in the SEL

**Characterize:** What are the relevant models, baselines and measures? What are the existing processes? What is the standard cost, relative effort for activities, reliability? What are the high risk areas?

**Set goals:** What are the expectations, relative to the baselines? What do we hope to learn, gain, e.g., Cleanroom with respect to changing requirements?

**Choose process:** How should the Cleanroom process be modified and tailored relative to the environment? E.g., formal methods hard to apply, require skill; may have insufficient data to measure reliability. Allow back-out options for unit testing certain modules.

**Execute:** Collect and analyze data based upon the goals, making changes to the process in real time

**Analyze data:** Try to characterize and understand what happened relative to the goals; write lessons learned

**Package experience:** modify the process for future use



## Single Project Study



### Cleanroom in the SEL

**Study Goal:**

Analyze the Cleanroom process in order to evaluate and compare it to a standard SEL development process with respect to the effects on the effort distribution, cost, and reliability from the viewpoint of quality assurance

**Environment:**

NASA/ SEL  
40 KLOC Ground Support System

**Experimental design:**

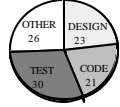
Case Study  
Data collected: effort distribution, change profile, productivity, level of rework, impact of spec changes, error rate, error distribution, error source



## Single Project Study Cleanroom in the SEL



### Sample Measures, Baselines, and Expectations

	Sample Measures	Sample Baseline	Sample Expectation
<b>PROCESS</b>	Effort distribution		Increased design % due to emphasis on peer review process
	Change profile		
<b>COST</b>	Productivity	Historically, 26 DLOC per day	No degradation from current level
	Level of rework		
	Impact of spec changes		
<b>RELIABILITY</b>	Error rate	Historically, 7 errors per KDLOC	Decreased error rate
	Error distribution		
	Error source		



## Single Project Study Cleanroom in the SEL



### Major Results

Can use for up to 40KLOC  
Can use with changing requirements  
Failure rate during test reduced by 25%  
Reduction in rework effort: 95% as opposed to 58% took < 1 hour to fix  
Productivity increased by about 30%

Effort distribution changes: more time in design  
50% of code time spent reading

Code appears in library: later than normal, more like a step function

Less computer use by a factor of 5

Concerns: Only 26% of faults found by both readers  
Formal Methods not applied effectively  
No payoff in reliability modeling

**Evolution:** Better training needed for methods and techniques  
Better mechanisms needed for uploading code to testers  
To allow testers to add requirements for output analysis of code

**Side effect:** Caused more emphasis on requirements analysis



## Multi-Project Analysis Study



### Cleanroom in the SEL

**Study Goal:**

Analyze the Cleanroom process  
in order to evaluate and compare it to a standard SEL development process with respect to the effects on the effort distribution, cost, and reliability

**Changes:**

Better training for methods and techniques - use box structure approach  
Fix uploading problem  
Allow clean compile  
Allow testers to add requirements for output analysis of code

**Environment:**

Project 2: 22 KLOC Flight Dynamics System (in-house)  
Project 3: 160 KLOC Flight Dynamics System (contractor)  
Project 4: 140 KLOC Flight Dynamics System (contractor)

**Experimental design:**

Multi-Project Study  
Data collected: effort distribution, change profile, productivity, level of rework, impact of spec changes, error rate, error distribution, error source



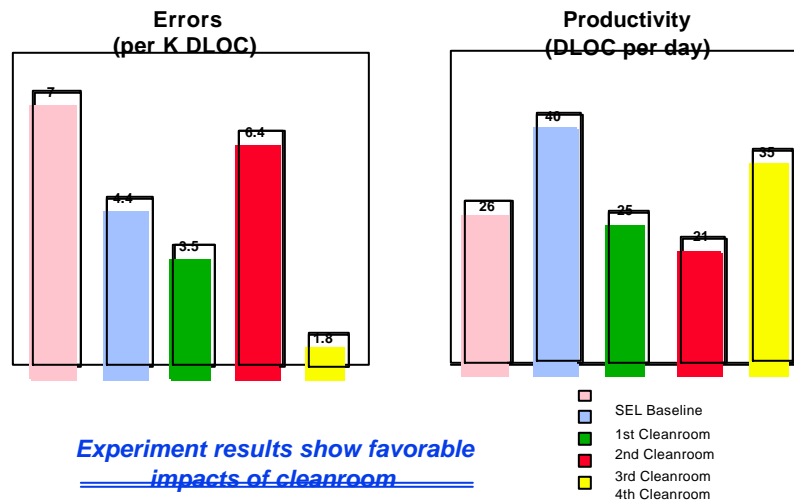
## Multi-Project Analysis Study Cleanroom in the SEL



	1st Cleanroom	2nd Cleanroom	3rd Cleanroom	4th Cleanroom
Project	ACME	SAMPEX	WIND/POLAR	SOHO AGSS
Size (developed lines)	40 KSLOC	23 KSLOC	160 KSLOC	141 ICSLOC
Size (total lines)	60 KLOC	39 KLOC	201 KLOC	485 KLOC
Dates	1988-90	1990-91	1990-92	1993-1995
Function	Attitude determination (Math)	Telemetry processing (data processing)	Full attitude (two missions)	Full attitude



## Multi-Project Analysis Study Cleanroom in the SEL



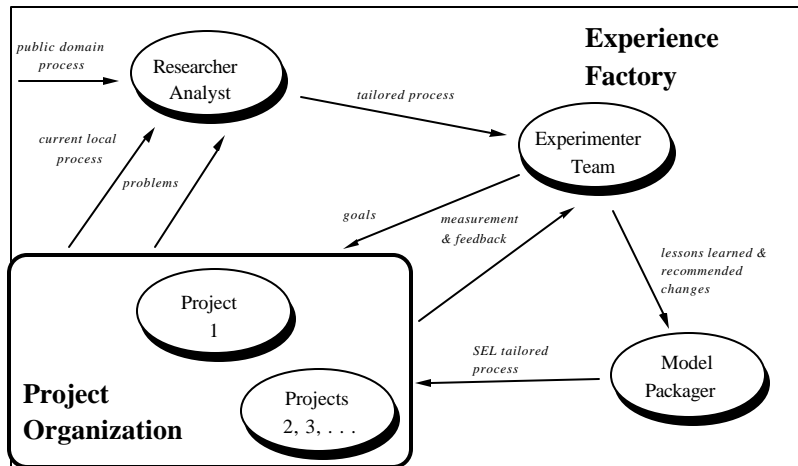
Experiment results show favorable  
impacts of cleanroom



## Multi-Project Analysis Study Improving via the Experience Factory



### Process Evolution/Evaluation



## Multi-Project Analysis Study Cleanroom in the SEL



### Major Results:

In-house, smaller project  
 very effective, reduced defects, slightly lower cost  
 Contractor, larger project  
 not as effective at first, very effective second time  
 Uploading Fixed  
 Formal Methods still not effectively applied  
 There was a duplication of documentation

### Evolution:

Although Cleanroom successful,  
 there are still problems reading and abstracting code formally  
 there are problems reading requirements and other documents  
 Area for study in the continual evolution of Cleanroom include:  
 improve reading techniques for requirements and design documents  
 To improve the existing reading techniques, we first spent time  
 understanding how they were reading the document  
 asked for problems and suggestions  
 Based upon these ideas, we have been working on reading approaches



## References



- V. Basili, The Experimental Paradigm in Software Engineering, Lecture Notes in Computer Science 706, Experimental Software Engineering Issues: Critical Assessment and Future Directives, H.D. Rombach, V. Basili, and R. Selby editors, Proceedings of Dagstuhl-Workshop, September 1992, published by Springer-Verlag, #706, Lecture Notes in Computer Software, August 1993.
- D. Campbell and J. Stanley, Experimental and Quasi-Experimental Designs for Research, Houghton Mifflin Co., Boston: 1963.
- V. Basili, R. Selby, and D. Hutchens, Experimentation in Software Engineering, IEEE Transactions on Software Engineering (invited paper), July 1986.
- R. Selby, V. Basili, and T. Baker, Cleanroom Software Development: An Empirical Evaluation, IEEE Transactions on Software Eng. , pp. 1027-1037, September 1987.
- V. Basili and R. Selby, Comparing the Effectiveness of Software Testing Strategies, IEEE Transactions on Software Engineering, pp. 1278-1296, December 1987.
- V. Basili and R. Selby, Paradigms for Experimentation and Empirical Studies in Software Engineering, Reliability Engineering and System Safety, vol. 32, no. 1-2, pp. 171-193, 1991.



## REFERENCES



- V. Basili and S. Green, Software Process Evolution at the SEL, IEEE Software, pp. 58-66, July 1994.
- A.A. Porter, L.G. Votta, and V. Basili, Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment, IEEE Transactions on Software Engineering, Volume 21, Number 6, pp. 563-575, June 1995.
- V. Basili, S. Green, O. Laitenberger, F. Shull, S. Sorumgaard, and M. Zelkowitz, The Empirical Investigation of Perspective-based Reading, Empirical Software Engineering, An International Journal, Volume 1, Number 2, pp. 133-164, Kluwer Academic Publishers, October 1996.
- V. Basili, Evolving and Packaging Reading Technologies, The Journal of Systems and Software, Volume 38, Number 1, pp. 3-12, July 1997.
- V. Basili and R. Selby, Data Collection & Analysis in Software Research and Management (invited paper), Proceedings of the American Statistical Association, July 1984.
- V. Basili, L. Briand, S. Condon, Y. Kim, W. Melo, and J. Valett, Understanding and Predicting the Process of Software Maintenance Releases, 18th International Conference on Software Eng., (ICSE'18), Berlin, Germany, March 25-29, 1996.