The Role of Experimentation in Software Engineering: Past, Present, and Future

Victor R. Basili

Experimental Software Engineering Group
Institute for Advanced Computer Studies
and
Department of Computer Science
University of Maryland
USA

Evolving Knowledge
Model Building, Experimenting, and Learning

Understanding a discipline involves **building models,**
e.g., application domain, problem solving processes

Checking our understanding is correct involves
- testing our models
- **experimentation**

Analyzing the results of the experiment involves **learning,** the **encapsulation of knowledge** and the ability to change and refine our models over time

The understanding of a discipline evolves over time

Knowledge encapsulation allows us to deal with higher levels of abstraction

This is the paradigm that has been used in many fields,
e.g., physics, medicine, manufacturing.

Evolving Knowledge
Model Building, Experimenting, and Learning

Outline

Motivation: Evolving knowledge through experimentation
Nature of the Software Engineering Discipline
- Early Observation
- Available Research Paradigms

Status of Model Building
Status of the Experimental Discipline
Maturing of the Experimental Discipline
- Evolution of Knowledge over time
- Reading Technology Experiments

Vision of the future

Evolving Knowledge
Model Building, Experimenting, and Learning

What do these fields have in common?

They evolved as disciplines when they began applying the cycle of model building, experimenting, and learning

Began with observation and the recording of what was observed

Evolved to manipulating the variables and studying the effects of change in the variables

What are the differences of these fields?

Differences are in the objects they study, the properties of those object, the properties of the system that contain them, the relationship of the object to the system, and the culture of the discipline

This effects
how the models are built
how the experimentation gets done

Evolving Knowledge
Model Building, Experimenting, and Learning

Physics

- understand and predict the behavior of the physical universe
- researchers: theorists and experimentalists
- has progressed because of the interplay between the groups

Theorists build models to explain the universe
- predict the results of events that can be measured
- models based on
  - theory about the essential variables and their interaction
  - data from prior experiments

Experimentalists observe, measure, experiment to
- test or disprove a hypothesis or theory
- explore a new domain

But at whatever point the cycle is entered there is a modeling, experimenting, learning and remodeling pattern

Early experimentalists only observed, did not manipulate the objects
Modern physicists have learned to manipulate the physical universe, e.g. particle physicists.

Evolving Knowledge
Model Building, Experimenting, and Learning

Medicine
- researcher and practitioner
- clear relationship between the two
- knowledge built by feedback from practitioner to researcher

Researcher aims at understanding the workings of the human body to predict effects of various procedures and drugs

Practitioner applies knowledge by manipulating processes on the body for the purpose of curing it

Medicine began as an art form
- evolved as a field when it began observation and model building

Experimentation
- from controlled experiments to case studies
- human variance causes problems in interpreting results
- data may be hard to acquire

However, our knowledge of the human body has evolved over time

Evolving Knowledge
Model Building, Experimenting, and Learning

Manufacturing
- domain researcher and manufacturing researcher
- understand the process and the product characteristics
- produce a product to meet a set of specifications

Manufacturing evolved as a discipline when it began process improvement

Relationship between process and product characteristics
- well understood

Process improvement based upon models of
- problem domain and solution space
- evolutionary paradigm of model building, experimenting, and learning
- relationship between the three

Models are built with good predictive capabilities
- same product generated, over and over, based upon a set of processes
- understanding of relationship between process and product

Software Engineering
The Nature of the Discipline

Like other disciplines, software engineering requires the cycle of model building, experimentation, and learning

Software engineering is a **laboratory science**

The **researcher's role** is to understand the nature of the processes, products and the relationship between the two in the context of the system

The **practitioner's role** is to build "improved" systems, using the knowledge available

More than the other disciplines these **roles are symbiotic**

The researcher needs laboratories to observe and manipulate the variables
- they only exist where practitioners build software systems

The practitioner needs to better understand how to build better systems
- the researcher can provide models to help

Software Engineering
The Nature of the Discipline

Software engineering is **development** not production

The technologies of the discipline are **human based**

All software is not the same
- there are a **large number of variables** that cause differences
- their effects need to be understood

Currently,
- **insufficient set of models** that allow us to reason about the discipline
- **lack of recognition of the limits** of technologies for certain contexts
- there is **insufficient analysis and experimentation**

Software Engineering
Early Observation

Belady & Lehman ('72,'76)
- **observed** the behavior of OS 360 with respect to releases
- posed theories based on observation concerning entropy

The idea
- that you might redesign a system rather than continue to change it
- was a revelation

But, Basili & Turner ('75)
- **observed** that a compiler system
- being developed using an incremental development approach
- gained structure over time, rather than lost it

How can these **seemingly** opposing statements be true?

What were the variables that caused the effects to be different?
Size, methods, nature of the changes, context?

Software Engineering
Early Observation

Walston and Felix ('79) identified **29** variables that had an effect on software productivity in the IBM environment

Boehm ('81) observed that **15** variables seemed sufficient to explain/ predict the cost of a project across several environments

Bailey and Basili ('81) identified **2** composite variables that when combined with size were a good predictor of effort in the SEL environment

There are numerous cost models with different variables

Why were the variables different?

What does the data tell us about the relationship of variables?

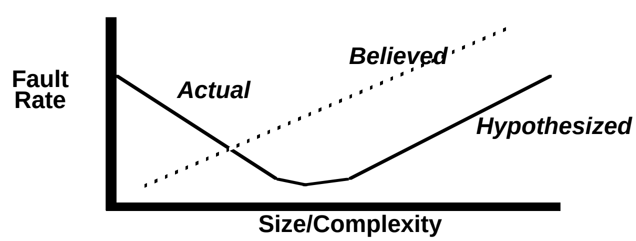
Which variable are relevant for a particular context?

What determines their relevance?

What are the ranges of the values variables and their effects?

Software Engineering
Early Observation

Basili & Perricone ('84) **observed** that the defect rate of modules shrunk as module size and complexity grew in the SEL environment



Seemed counter to folklore that smaller modules were better, **but**

- interface faults dominate
- developer tend to shrink size when they lose control

This result has been observed by numerous other organizations

But **defect rate is only one dependent variable**

What is the effect on other variables? What size minimizes the defect rate?

Available Research Paradigms?

The **analytic paradigm**:
- propose a formal theory or set of axioms
- develop a theory
- derive results and
- if possible, verify the results with empirical observations.

Experimental paradigm:
- observing the world (or existing solutions)
- proposing a model or a theory of behavior (or better solutions)
- measuring and analyzing
- validating hypotheses of the model or theory (or invalidate
- repeating the procedure evolving our knowledge base

The experimental paradigms involve
- experimental design
- observation
- quantitative or qualitative analysis
- data collection and validation on the process or product being studied

Available Research Paradigms?

Quantitative Analysis
- obtrusive controlled measurement
- objective
- verification oriented

Qualitative Analysis
- naturalistic and uncontrolled observation
- subjective
- discovery oriented

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 Status of Model Building

Modeling research

- software product
  - mathematical models of the program function
  - product characteristics, such as reliability models
- variety of process notations
- cost models, defect models

Little experimentation

- implementation yes, experimentation no

Why? Model builders

- theorists, expect the experimentalists to test the theories
- view their "models" as self evident, not needing to be tested

For any technology

- Can it be applied by a practitioner?
- Under what conditions its application is cost effective?
- What kind of training is needed for its successful use?

What is the effect of the technique on product reliability, given an environment of expert programmers in a new domain, with tight schedule constraints, etc.?

The Status of the Experimental Discipline

Where are we in the spectrum of model building, experimentation, and learning in the software engineering discipline?

These have been formulated as three questions

What are the components and goals of the software engineering studies?

- what we are studying and why

What kinds of experiment have been performed?

- the types and characteristics of the experiments run

How is software engineering experimentation maturing?

- judgements against some criteria and examples

The Status of the Experimental Discipline

What are the components of the studies?

We use four parameters (based on the GQM template):

object of study: a process, product, any form of model

purpose: characterize (what happens?)
- evaluate (is it good?)
- predict (can I estimate something in the future?)
- control (can I manipulate events?)
- improve (can I improve events?)

focus: the aspect of the object of study that is of interest
- reliability of the product
- defect detection/prevention capability of the process
- accuracy of the cost model

point of view: the person who benefits from the information
- the researcher in understanding something better

Identified two patterns:

human factor studies

project-based studies

The Maturing of the Experimental Discipline

What are the components of the studies?

Human-factor studies
- object of study: a small cognitive task
- focus: some performance measure
- purpose: evaluation
- point of view: researcher

Done by/with cognitive psychologists comfortable with experimentation

Have remained studies in the small

Project-based studies
- object of study: software process, product, ...
- focus: a variety from product reliability and cost to process effect
- purpose: evaluation, some prediction; characterization/understanding
- point of view: the researcher (often a practitioner view)

Done mostly by software engineers, less adept at experimentation

Have evolved from small, specific items,
- like particular programming language features
- to include entire development processes, like Cleanroom

The Status of the Experimental Discipline

What kinds of studies have been performed?

1. Are the study results descriptive, correlational, cause-effect?

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)

Human factor: mostly cause-effect
- Sign of maturity of experimentalists; size nature of problem

Project-based: evolved (?) from correlational to descriptive studies
- Reflects early beliefs that problem was simple and some simple combination of metrics could explain cost, quality, etc.
- Don't have an observational knowledge base

The Status of the Experimental Discipline

What kinds of studies have been performed?

2. Is the study performed on novices or experts or both?
novice: students or individuals not experienced in domain
experts: practitioners or people with experience in domain

Human-Factor: investigate difference between novices and experts
Project-based: more studies with experts, especially descriptive studies of organizations and projects

3. Is the study performed in vivo or in vitro?
In vivo: in the field under normal conditions
In vitro: in the laboratory under controlled conditions

Human-Factor: more in vitro
Project-based: more in vivo

4. Is it an experiment or an observational study?
Experiment: at least one treatment or controlled variable
Observational study: no treatment or controlled variables

The Status of the Experimental Discipline

What kinds of studies have been performed?

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

The Maturing of the Experimental Discipline

What kinds of studies have been performed?

Experiment Classes

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

The Maturing of the Experimental Discipline

What kinds of studies have been performed?

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 Status of the Experimental Discipline

What kinds of studies have been performed?

Observational Studies

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

The Maturing of the Experimental Discipline

How is experimentation maturing?

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 Maturing of the Experimental Discipline

How is experimentation maturing?

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

The Maturing of the Experimental Discipline

How is experimentation maturing?

There is some evidence that researchers appear to be
- asking more **sophisticated questions**
- **studying relationships** between processes/product characteristics
  - using more studies **in the field** than in the controlled laboratory
  - **combining** various **experimental classes** to build knowledge

On such example of evolving knowledge over time,
- based upon experimentation and learning is
- the evolution of the SEL knowledge
- of the effectiveness of reading techniques and methods

Software Engineering Laboratory is a consortium (established in 1976)

- NASA/Goddard Space Flight Center
- University of Maryland
- Computer Sciences Corporation

Evolution of Knowledge over Time
Reading Technology Experiments

This example
- shows the combination of **multiple experimental designs**
- provides insight into the **effects of different variables** on reading
- demonstrates **replication by other researchers**

The experiments start with
the early reading vs. testing experiments
to various Cleanroom experiments
to the development of new reading techniques currently under study
to replications at other groups

The experiments are based upon the ideas that
Reading is a key technical activity
for improving the analysis of all kinds of software documents
and we need to better understand its effect

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

But reading was simply reading, without a technological base

EXPERIMENTAL LEARNING MECHANISMS

Series of Studies

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

EXPERIMENT
Blocked Subject Project Study

Analysis Technique Comparison

Technique Definition:

Code Reading vs **Functional Testing** vs **Structural Testing**
Compare with respect to:
fault detection effectiveness and cost
classes of faults detected

Experimental design:
Fractional factorial design

Environment:
University of Maryland (43) and then NASA/CSC (32)
Module size programs (145 - 365 LOC), seeded with faults
Cause-effect, in vitro, novices and experts

Blocked Subject Project Study
Testing Strategies Comparison

Fractional Factorial Design

		Code Reading P1	Code Reading P2	Code Reading P3	Functional Testing P1	Functional Testing P2	Functional Testing P3	Structural Testing P1	Structural Testing P2	Structural Testing P3
Advanced Subjects	S1			X		X		X		
	S2		X		X					X
	:									
	S8	X					X		X	
Intermediate Subjects	S9			X		X		X		
	S10		X		X					X
	:									
	S19	X					X		X	
Junior Subjects	S20			X		X		X		
	S21		X		X					X
	:									
	S32	X					X		X	

Blocking by experience level and program tested

NASA/CSC

Blocked Subject Project Study
Analysis Technique Comparison

Some Results (NASA/CSC)

Code reading more
- **effective** than functional testing
- **efficient** than functional or structural testing

Different techniques more effective for different defect classes
- code reading more effective for interface defects
- functional testing more effective for control flow defects

Code readers assessed the true quality of product better than testers

After completion of study:
- Over 90% of the participants thought functional testing worked best

Some Lessons Learned

Reading is effective/efficient; the particular technique appears important

The choice of techniques should be tailored to the defect classification

Developers don't believe reading is better

Blocked Subject Project Study
Analysis Technique Comparison

Based upon this study

reading was implemented as part of the SEL development process

But - reading appeared to have very little effect

Possible Explanations (NASA/CSC)

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 the reading technique and the reading method

NEXT: Is there an approach with reading motivation and technique?

⇨ Try Cleanroom in a controlled experiment at the University of Maryland

EXPERIMENT
Replicated Project Study

Cleanroom Study

Approaches:

Cleanroom process vs. **non-Cleanroom process**

Compare with respect to:

effects on the process product and developers

Experimental design:

15 three-person teams (10 teams used Cleanroom)

Environment:

University of Maryland

Electronic message system, ~ 1500 LOC

novice, in vitro, cause-effect

Replicated Project Study
Cleanroom Evaluation

Some Results

Cleanroom developers
- more effectively applied off-line review techniques
- spent less time on-line and used fewer computer resources
- made their scheduled deliveries

Cleanroom product
- less complex
- more completely met requirements

Some Lessons Learned

Cleanroom developers were motivated to read better

Cleanroom/Reading by step-wise abstraction was effective and efficient

NEXT: Does Cleanroom scales up? Will it work on a real project?
Can it work with changing requirements?
⇨ Try Cleanroom in the SEL

EXPERIMENT
Single Project Study

First Cleanroom in the SEL

Approaches:

Cleanroom process vs. **Standard SEL Approach**

Compare with respect to:
effects on the effort distribution, cost, and reliability

Experimental design:
Apply to a live flight dynamics domain project in the SEL

Environment:
NASA/ SEL
40 KLOC Ground Support System
in vivo, experts, descriptive

Single Project Study
First Cleanroom in the SEL

Some Results

Cleanroom was
- effective for 40KLOC
  - failure rate reduced by 25%
  - productivity increased by 30%
  - less computer use by a factor of 5
- usable with changing requirements
  - rework effort reduced
    - 5% as opposed to 42% took > 1 hour to change

Some Lessons Learned

Cleanroom/Reading by step-wise abstraction was effective and efficient
Reading appears to reduce the cost of change

Better training needed for reading methods and techniques

NEXT: Will it work again? Can we scale up more? Can we contract it out?
⇨ Try on larger projects, contracted projects

EXPERIMENT
Multi-Project Analysis Study

Cleanroom in the SEL

Approaches:
Revised Cleanroom process vs. **Standard SEL Approach**

Compare with respect to:
effects on the effort distribution, cost, and reliability

Experimental design:
Apply to three more flight dynamics domain projects in the SEL

Environment:
NASA/ SEL
Projects: 22 KLOC (in-house)
160 KLOC (contractor)
140 KLOC (contractor)
in vivo, experts, descriptive

Multi-Project Analysis Study
Cleanroom in the SEL

Major Results

Cleanroom was
- effective and efficient for up to ~ 150KLOC
- usable with changing requirements
- took second try to get really effective on contractor, large project

Some Lessons Learned

Cleanroom Reading by step-wise abstraction
- effective and efficient in the SEL
- takes more experience and support on larger, contractor projects
- appears to reduce the cost of change

Unit test benefits need further study

Better training needed for reading techniques

Better techniques for other documents, e.g., requirements, design, test plan

NEXT: Can we improve the reading techniques for requirements and design documents?

⇨ Develop reading techniques and study effects in controlled experiments

Scenario-Based Reading Definition

Goal: To define a set of reading technologies that can be
- document and notation specific
- tailorable to the project and environment
- procedurally defined
- goal driven
- focused to provide a particular coverage of the document
- empirically verified to be effective for its use
- usable in existing methods, such as inspections

An approach to generating a family of reading techniques, called **operational scenarios,** has been defined

So far, two different techniques defined for requirements documents:
defect based reading
perspective based reading

Both techniques have been applied in a series of experiments

EXPERIMENTING
Blocked Subject-Project Study

Scenario-Based Reading

Approaches:
defect-based reading vs **ad-hoc reading** vs **check-list reading**

Compare with respect to:
fault detection effectiveness in the context of an inspection team

Experimental design:
Partial factorial design
Replicated twice
Subjects: 48 subjects in total

Environment:
University of Maryland
Two Requirements documents in SCR notation
Documents seeded with known defects
novice, in vitro, cause-effect

EXPERIMENTING
Blocked Subject Project Study

Scenario-Based Reading

Approaches:
perspective-based reading vs **NASA's reading technique**

Compare with respect to:
fault detection effectiveness in the context of an inspection team

Experimental design:
Partial factorial design
Replicated twice
Subjects: 25 subjects in total

Environment:
NASA/CSC SEL Environment
Requirements documents:
Two NASA Functional Specifications
Two Structured Text Documents
Documents seeded with known defects
expert, in vitro, cause-effect

Blocked Subject Project Study
Scenario-Based Reading

Some Results

Scenario-Based Reading performed better than
Ad Hoc, Checklist, NASA Approach reading
especially when they were less familar with the domain

Scenarios helped reviewers focus on specific fault classes
but were no less effective at detecting other faults

The relative benefit of Scenario-Based Reading is higher for teams

Some Lessons Learned

Need better tailoring of Scenario-Based Reading to the NASA environment in terms of document contents, notation and perspectives

Need better training to stop experts from using their familiar technique

Next: Tailor better for NASA and run a case study at NASA
Replicate these experiments in many different environments
- varying the context

The Maturing of the Experimental Discipline

How is experimentation maturing?

Several of these **experiments have been replicated**
- under the same and differing contexts

The original analysis technique comparison has been replicated
University of Kaiserslautern

Scenario-based reading study variations
University of Bari, Italy
University of New South Wales, Australia
Bell Laboratories, USA
University of Trondheim, Norway
Bosch, Germany
to better understand the reading variable

ISERN
organized explicitly to share knowledge and experiments
has membership in the U.S., Europe, Asia, and Australia
represents both industry and academia
supports the publication of artifacts and laboratory manuals

Its goal is to evolve software engineering kwowledge over time, based upon experimentation and learning

What will our future look like?

Experimentation can provide us with
- better **scientific and engineering basis** for the software engineering
- **better models** of
  - software processes and products
  - various environmental factors, e.g. the people, the organization
- better **understanding of the interactions** of these models

Practitioners will be provided with
- the ability to control and manipulate project solutions
  - based upon the environment and goals set for the project
- knowledge based upon empirical and experimental evidence
  - of what works and does not work and under what conditions

Researchers will be provided laboratories for experimentation

This will require a research plan that will take place over many years
- coordinating experiments
- evolving with new knowledge