What's so hard about replication of Software Engineering experiments?

Victor R. Basili

University of Maryland
and
Fraunhofer Center - Maryland

Introduction

- Why replicate experiments:
  - to verify the results from the first experiment
  - to expand our knowledge of the discipline
  - to build models that can be used to predict and be challenged
- What does it mean to replicate an experiment?
- What are the criteria for a replication?
- Is it the same as replicating a physics experiment?

- Over a decade ago, proposed a framework for building a body of software engineering knowledge through replication by
  - identifying key dependent and independent variables and using them to integrate collections of experiments with 'like' hypotheses
  - using context variables as a way to expand upon our knowledge and address the question of results generalization
  - we used it to integrate the results of some reading technique studies

Experimental Research Agenda

- This research was motivated by an NSF research grant to the Experimental Software Engineering Group, University of Maryland

- **Develop families of techniques and methods**
  - based on empirical evaluation
  - parameterized for use in different contexts
  - evaluated for those contexts

- **Evaluate the approaches** and **criteria** to
  - assess methods/techniques in laboratory and industrial settings
  - determine if a method/technique is appropriate for its context

- **Build an Experience Base of technology evaluations**
  - representing an **integrated body of knowledge**
  - accessible by researchers and practitioners
  - who can append their own experiences

Motivation
Evolving Knowledge in a Discipline

- Understanding a discipline involves learning, i.e.,
  - observation
  - reflection, and encapsulation of knowledge
  - model building (application domain, problem solving processes)
  - experimentation
  - model evolution over time
- This is the paradigm that has been used in many fields,
  - e.g., physics, medicine, manufacturing.
- The differences among the fields are
  - **how models are built and analyzed**
  - **how experimentation gets done**

Motivation: Evolving Knowledge In Software Engineering

- The study of **Software engineering** is a '**laboratory science**'
- We need to understand the nature of the processes, products and the relationship between the two in the context of the system
- All software environments are not the same
  - there are a large number of variables that cause differences
  - their effects need to be understood and studied
- Currently,
  - **insufficient set of models** to reason about the discipline
  - **lack of recognition of the limits** of technologies for the context
  - there is **insufficient analysis and experimentation**

Motivation: Evolving Bodies of Knowledge from Experiments

- Many categories: from controlled experiments to case studies
- Performed for many purposes: to study process effects, product characteristics, environmental constraints (cost or schedule).
- Typically we are looking for a relationship between two variables, such as the relationship between process characteristics and product characteristics
- **Problems** with experiments (controlled)
  - the **large number of variables** that cause differences
  - deal with **low level issues**, **microcosm of reality**, **small set of variables**
- **=> Combining different kinds of experiments** is necessary to build a body of knowledge that is useful to the discipline

Criteria for building comprehensive bodies of knowledge in Software Engineering

- Sets of **high level hypotheses**
  - address interest of the software engineering community
  - identify sets of dependent and independent variables
  - provide options for the selecting detailed hypotheses
- Sets of **detailed hypotheses**
  - written in a context that allow for a well defined experiment
  - combinable to support high level hypotheses
- **Context variables** that can be changed to allow for
  - experimental design variation (make up for validity threats)
  - specifics of the process context;
- **Sufficient documentation** for replication and combination
- **Community of researchers** willing to collaborate and replicate.

Choosing a High Level Focus

- General Interest to the community
  - **Analyzing the Effects of a SE Process on a Product**
- What are the high level **questions of interest**?
  - Can we effectively design and study techniques that are procedurally defined, document and notation specific, goal driven, and empirically validated for use?
  - Can we demonstrate that a procedural approach to a software engineering task could be more effective than a less procedural one under certain conditions?
- What are the **high level hypotheses**?
  - A reading technique that is procedurally defined, document and notation specific, and goal driven for use is more effective than one that does not have these characteristics
  - A procedural approach to reading based upon specific goals will find different defects than one based upon different goals

Example: Understanding for Use
Motivation for Reading

Why pick reading?

- Reading is a **key technical activity** for analyzing and constructing software documents and products
- 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

Choosing a High Level Focus

- How do we build a framework for combining hypotheses from individual experiments, isolating out individual variables?
- Consider using the **Goal/Question/Metrics Paradigm**
- Goal Template:
  - Analyze an **object of study** in order to **purpose** with respect to **focus** from the point of view of **who** in the context of **environment**
- Consider decomposing each of the variables to identify and classify the independent, dependent, and context variables

The Experience Factory Organization
Goal/Question/Metric Paradigm

A mechanism for defining and interpreting operational, measurable goals

It uses four parameters:

a model of an **object of study,**
e.g., a process, product, or any other experience model

a model of one or more **focuses,**
e.g., models that view the object of study for particular characteristics

a **point of view,**
e.g., the perspective of the person needing the information

a **purpose,**
e.g., how the results will be used

to generate a **GQM model** relative to a **particular environment (context variables)**

Choosing a High Level Focus

- Analyzing the Effects of SE Processes on Products
  - Analyze **processes** to evaluate their effectiveness on a product from the point of view of the knowledge builder in the context of (variable set)

- Characterize the object of study:
  - Object of Study (**Process**, Product, …)
  - Process Class (Life Cycle Model, Method, **Technique**, Tool, …)
  - Technique Class (**Reading**, Testing, Designing, …)

- Analyze **reading techniques** to evaluate their effectiveness on a product from the point of view of the knowledge builder in the context of some variable set

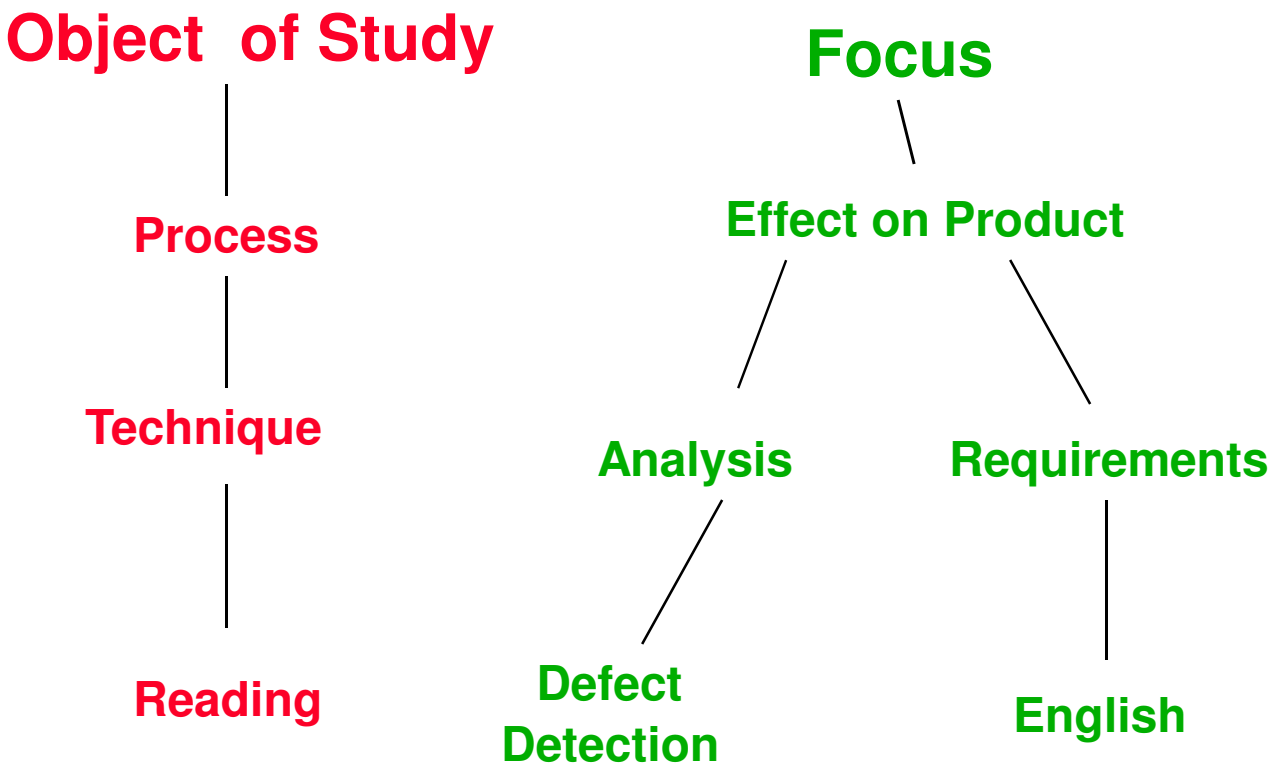
Choosing a High Level Focus

- Analyze reading techniques to evaluate their **effectiveness on products** from the point of view of the knowledge builder in the context of variable set (G1)

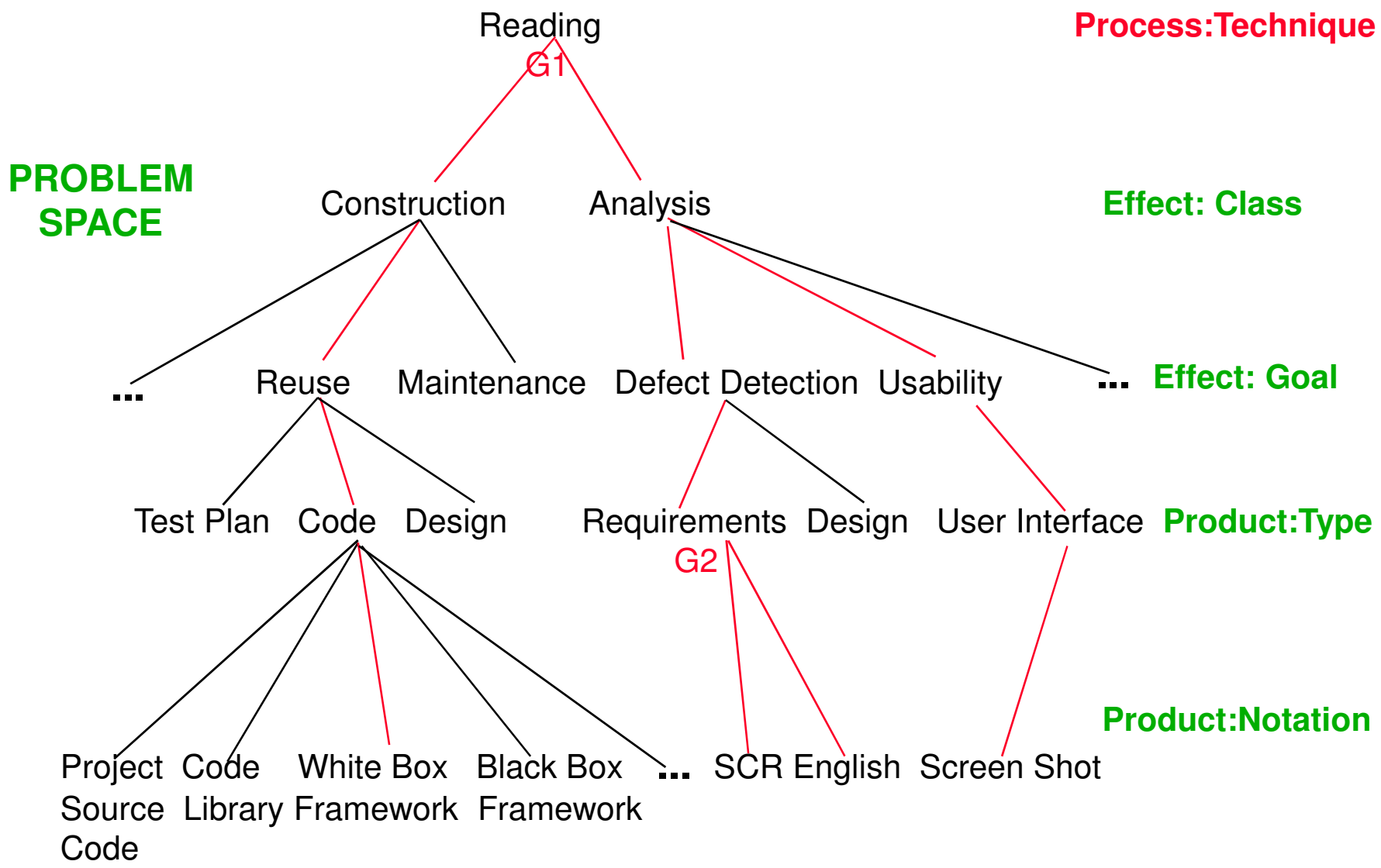
- Characterize the focus: **Effectiveness on a Product**
  - Effectiveness Class (Construction, **Analysis**, …)
  - Effectiveness Goal (**Defect Detection**, **Usability**, …
  - Product Type (**Requirements**, Design, Test Plan, **User Interface**, …
  - Product Notation (**English**, **SCR**, Mathematics, **Screen Shot**, …

- Example Goal: Analyze reading techniques to evaluate their **ability to detect defects in a Requirements Document** from the point of view of the knowledge builder in the context of variable set (G2)

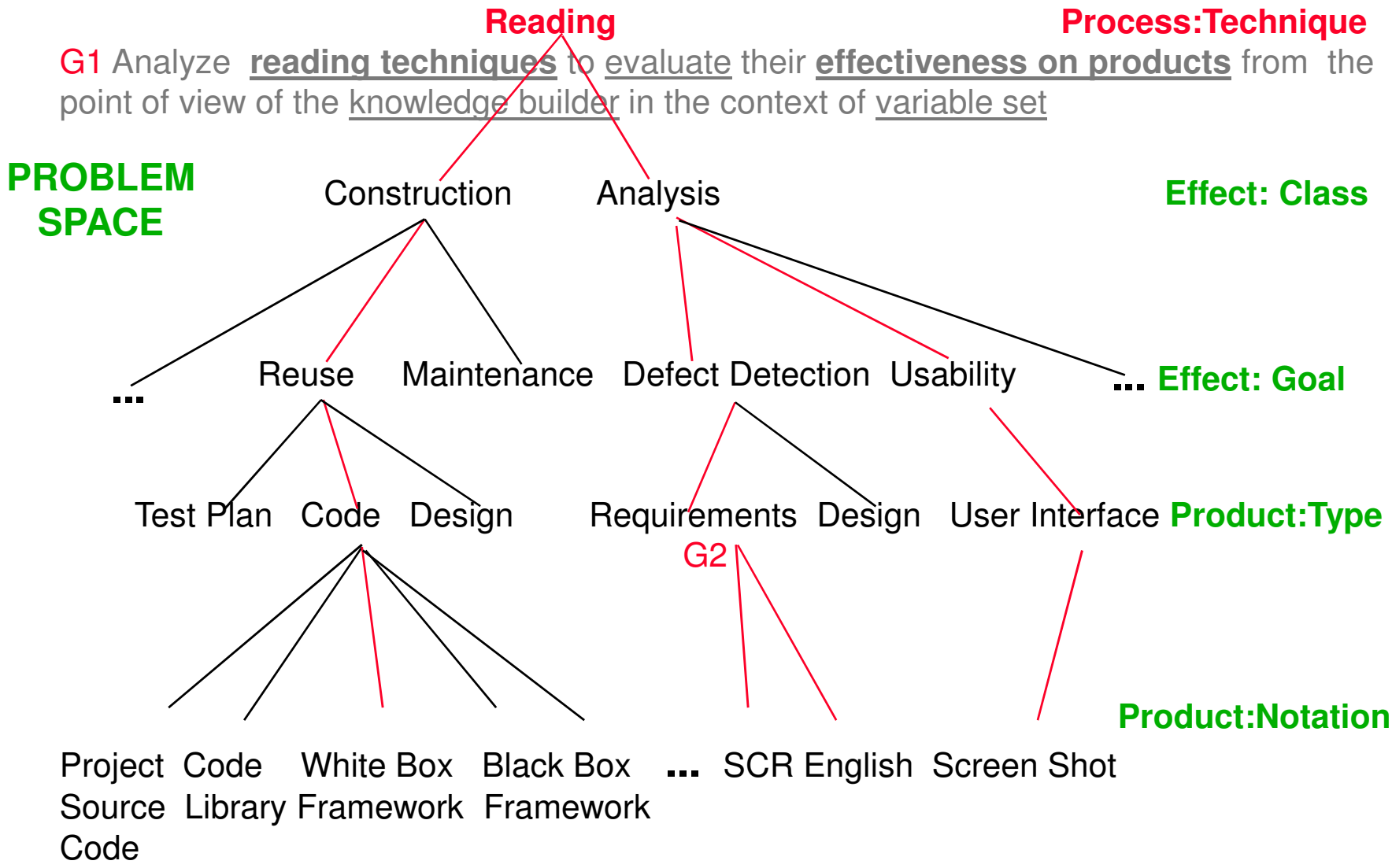
Refining a High Level Focus



Families of Reading Techniques



Families of Reading Techniques



Scenario-Based Reading Definition

- Given this set of characteristics/dimensions, an approach to generating a family of reading techniques, called **operational scenarios**, has been defined

- **Goals:** 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

- These goals defines a set of guidelines/characteristics for a process definition for reading techniques that can be studied experimentally

Choosing a Specific Focus from the Experimental Framework

- Characterize the process:
  - Technique Class (**Reading**, Testing, Designing, …)
  - Technique Characteristics (**goal oriented**, **procedurally based**, **coverage focussed**, **documentation and notation specific**, …)

- Analyze a **set of goal-oriented, procedurally-based, coverage focussed, document and notation specific reading techniques** to evaluate their effectiveness on a product from the point of view of the knowledge builder in the context of (variable set)

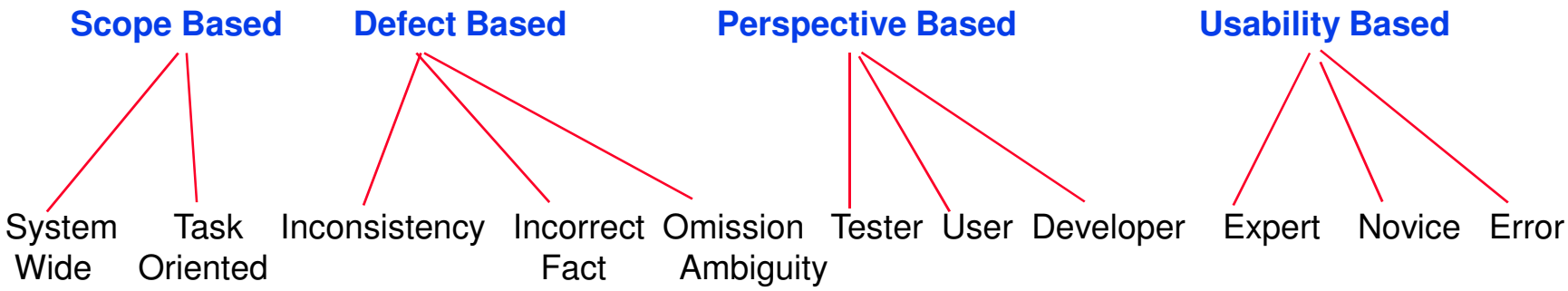
- Analyze a set of scenario based reading techniques to evaluate their effectiveness on products from the point of view of the knowledge builder in the context of (variable set)

- Attempts to satisfy the high level hypotheses and provide a frameworks for individual experiments

Choosing a Specific Focus from the Experimental Framework

- Analyze a set of scenario based reading techniques to evaluate their effectiveness on products from the point of view of the knowledge builder in the context of (variable set)

- We had developed four families of reading techniques at the time
  - parameterized for use in different contexts and
  - evaluated experimentally in those contexts



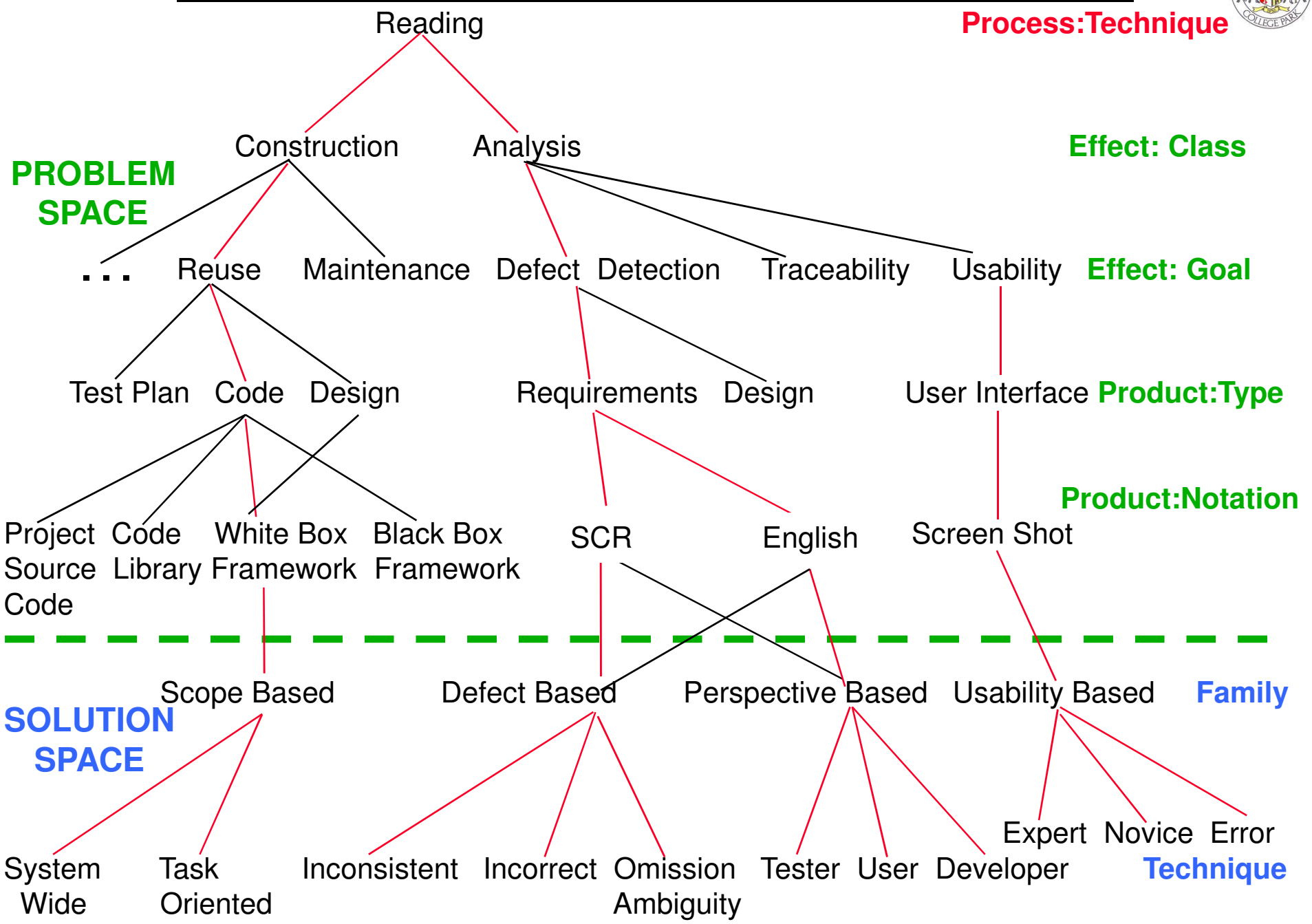
Choosing a Specific Focus from the Experimental Framework

- Analyze a **set of scenario based reading techniques** to evaluate their **ability to detect defects in a Requirements Document** from the point of view of the knowledge builder in the context of (variable set)

- Example: Perspective -Based Reading:
  - Choose perspectives; designer, tester, user
  - Define procedural processes for each perspective
  - Choose experimental treatment
  - Choose defect classes
  - etc.

- Contexts (context variables) can be continually expanded, e.g., NASA/SEL subjects, Professional Software Engineering student, Bosch project personnel

Families of Reading Techniques



Sample Set of Experiments

We have developed four families of reading techniques

parameterized for use in different contexts

evaluated experimentally in those contexts

some involved us as directly as experimenters IE, others did not OE

IE: Scott Green, Filippo Lanubile, Forrest Shull, Marvin Zelkowitz, Zhijun Zhang

Perspective Based Reading

IE: NASA/GSFC and UM Professional SE Course

OE: Germany (Bosch), Norway, Italy, ..Brazil

Usability-Based Reading

IE: Bureau of Census, UM students, UM Professional SE Course

Defect-Based Reading:

IE: UM students, Lucent Bell Laboratories

OE: Sweden, Italy, …

Scope-Based Reading

IE: UM Students

Choosing a Specific Focus from the Experimental Framework

- There are still many questions that need to be covered:
  - Process variable (**Independent variable**) issues:
    - How do we define/specify the process?
    - How do we account for process conformance?
  - Effectiveness of Product (**Dependent variable**) issues:
    - How do we select good criteria for effectiveness?
  - **Context Variables** Issues:
    - What subjects are performing the process?
    - What types of product is it performed on?
    - …(need a list to identify potential effects and generalizing)
- Questions associated with the variables need to be further specified and documented for replication

- Varying the values of these variables allow us to
  - vary the detailed hypotheses
  - support validity of study results

Designing Detailed Experiments to Increase Knowledge

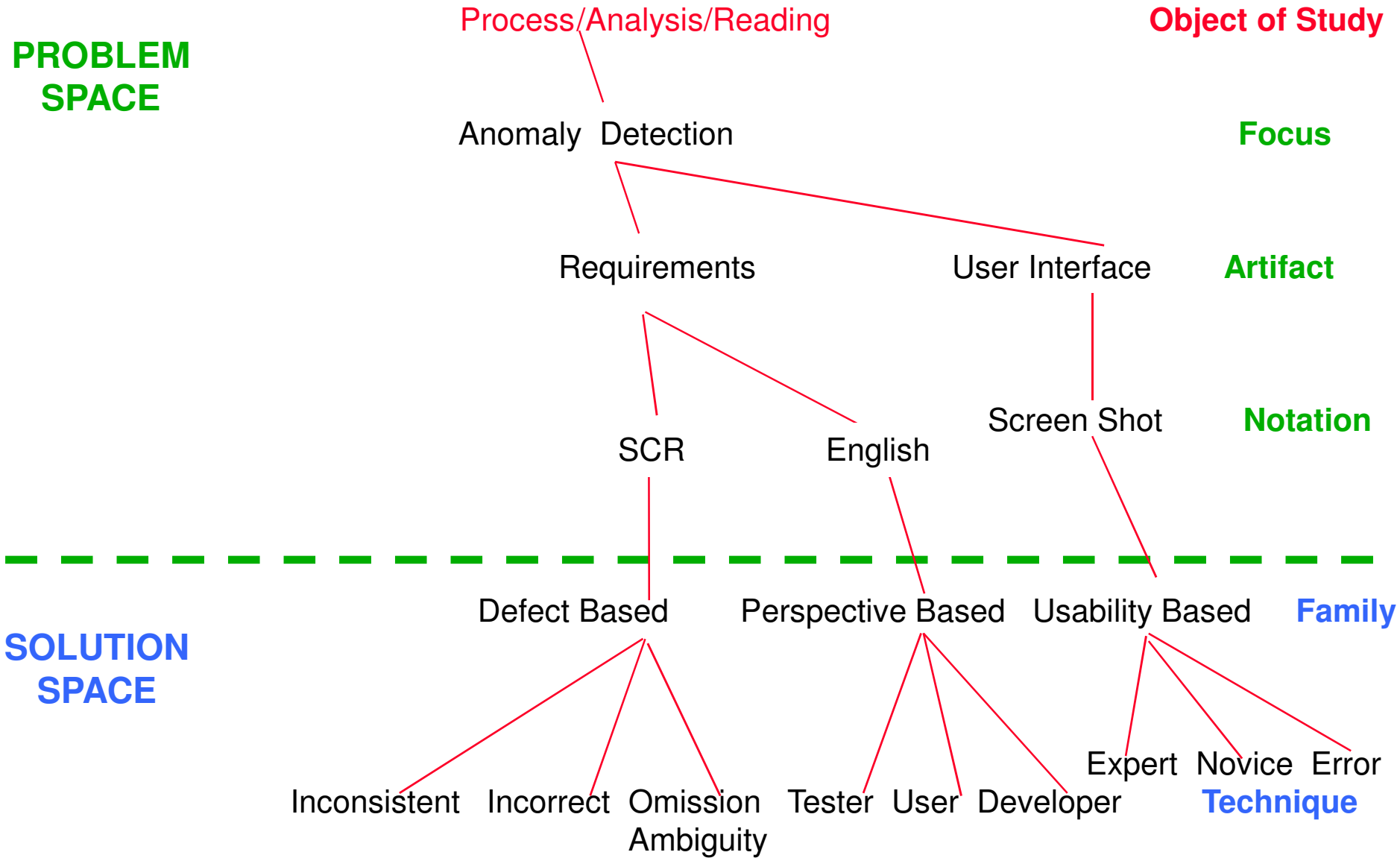
- We can build up knowledge by **replicating** detailed experiments, keeping the same hypothesis, combining results

- Varying **Context Variables** **(is this still a replication?)**
  - subject experience
  - context (classroom, toy, off-line, in project)
  - variability among subjects
  - Type of product involved
  - Vary order of events and activities

- Allows us to balance threats to validity
  - interaction of experience and treatment
  - spontaneous migration of subjects across treatments
  - replicating to counterbalance

Focused Families of Analysis Techniques

G3 Analyze a **set of processes focused to provide a particular coverage of an artifact** to evaluate their **ability to detect anomalies** from the point of view of the knowledge builder in the context of (variable set)



Conclusions from Experiments

- Able to **combine** the **results** of several experiments and **build up** our **knowledge** about software processes
  - We can **effectively design and study techniques** that are procedurally defined, document and notation specific, goal driven, and empirically validated for use
  - We can demonstrate that a **procedural approach** to a software engineering task could be more effective than a less procedural one under certain conditions (e.g., depends on experience)
  - A procedural approach to reading based upon **specific goals** will find defects related to those goals, so reading can tailored to the environment
  - et. al.

Conclusions about Knowledge Building Experimental Framework

- Benefit to Researchers:
  - ability to **increase the effectiveness** of individual experiments
  - offers a **framework** for building relevant practical SE knowledge
  - provides a way to develop and integrate **laboratory manuals**
  - generate a **community** of experimenters

- Benefits to Practitioners:
  - offers some relevant **practical SE knowledge**
  - provides a better basis for making judgements about **selecting process**
  - shows importance of and ability to tailor "**best practices**"
  - provides support for defining and **documenting processes**
  - allows organizations to **integrate their experiences** with processes

Combining Evaluation Approaches

Going past changing context in a controlled experiment

- **Controlled experiments** have limits
  - don't scale up
  - done in classroom/training situations
  - in vitro
  - face a variety of threats to validity
  - are high risk
- What specific problems do these cause? How can we deal with these problems? How do we balance the various threats to validity?
- One approach is to run **multiple studies**, mixing controlled experiments and case studies, building our knowledge in pieces
- Another approach is to apply **multiple evaluation methods** to the same study,
  - e.g., a mix of quantitative and qualitative analysis

Combining Evaluation Approaches

Running Multiple Studies

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

Running Multiple Studies to Learn

Reading Technique Experiments

- 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
- Early experiments (Hetzel, Meyers) showed very little difference between reading and testing
- **But reading was simply reading, without a technological base**

Running Multiple Studies to Learn

Reading Technique Experiments

Series of Studies

		# Projects	
		One	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

Code Reading vs **Functional Testing** vs **Structural Testing**

Study: fault detection effectiveness, cost, classes of faults detected

Experimental design: Fractional factorial design at NASA/CSC

Some Results

Code reading (by stepwise abstraction) more
- **effective** than functional testing (equivalence partitioning)
- **efficient** than functional or structural testing (100%stmt coverage)

Different techniques more effective **for different defect classes**

Developers don't believe reading is better, not motivated to read

Blocked Subject Project Study

Testing 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	
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

EXPERIMENT

Replicated Project Study

Cleanroom Study

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

Study: effects on the process, product, developers

Experimental design: 15 three-person teams at UMD

Some Results

Cleanroom developers were motivated to read better

Reading by step-wise abstraction more effective and efficient

Does Cleanroom scale up? Will it work on a real project?

EXPERIMENT

Single Project Study

Cleanroom in the SEL

Cleanroom process vs. **Standard SEL Approach**

Study: effects on the effort distribution, cost, and reliability

Experimental design: Flight Dynamics project in the SEL

Some Results

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

Better training needed for reading by stepwise abstraction

Will it work again? Can we scale up more?

EXPERIMENT

Multi-Project Analysis Study

Cleanroom in the SEL

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

Study: effects on the effort distribution, cost, and reliability

Experimental design: Three Flight Dynamics projects in the SEL

Some Results

Reading by step-wise abstraction
- effective and efficient in the SEL
- appears to reduce the cost of change

Better training needed for reading by stepwise abstraction

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

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

EXPERIMENTING

Blocked Subject Project Study

Scenario-Based Reading

Perspective-Based Reading (PBR) vs **NASA's reading technique**

Study: fault detection effectiveness in the context of an inspection team

Experimental design: Partial factorial design, replicated twice in SEL

Some Results

Scenario-Based Readers performed better than
- Ad Hoc, Checklist, NASA Approach reading
- especially when they were less familiar with the domain
- benefit higher for teams

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

Would better tailoring and more specificity of PBR
- improve the effects
- stop experts from using a familiar technique

Combining Evaluation Approaches

Backing up a Controlled Experiment

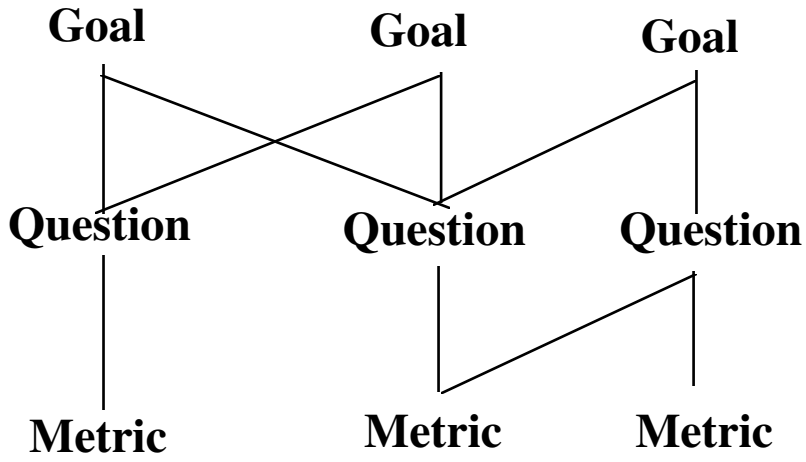
- So we used **multiple studies**, mixing controlled experiments and case studies to
  - provide insights
  - help us build models
  - deal with some of the problems,
    - e.g., scale up, the limitations of in vivo studies, and some threats to validity

- Building our knowledge in pieces ,
  - we have learned that
    - reading is an important process
    - the specific technique matters
    - there is a learning curve and old habits don't die easily
  - we have better understood
    - the difficulty of running controlled experiments
    - the need for better models of cognitive processes when defining reading techniques

Back up Slides

- Reading Studies

GOAL/QUESTION/METRIC PARADIGM
Goal and Model Based Measurement



A Goal links two models: a model of the **object of interest** and a model of the **focus** and develops an integrated GQM model

Goal: Analyze the **final product** to **characterize** it with respect to the **various defect classes** from the point of view of the **organization**

Question: What is the error distribution by phase of entry

Metric: Number of Requirements Errors, Number of Design Errors, ...

High Level Reading Goals

We differentiate two goals for reading techniques:

Reading for analysis:
Given a document,
how do I assess
various qualities
and characteristics?

Assess for
product quality
defect detection
...

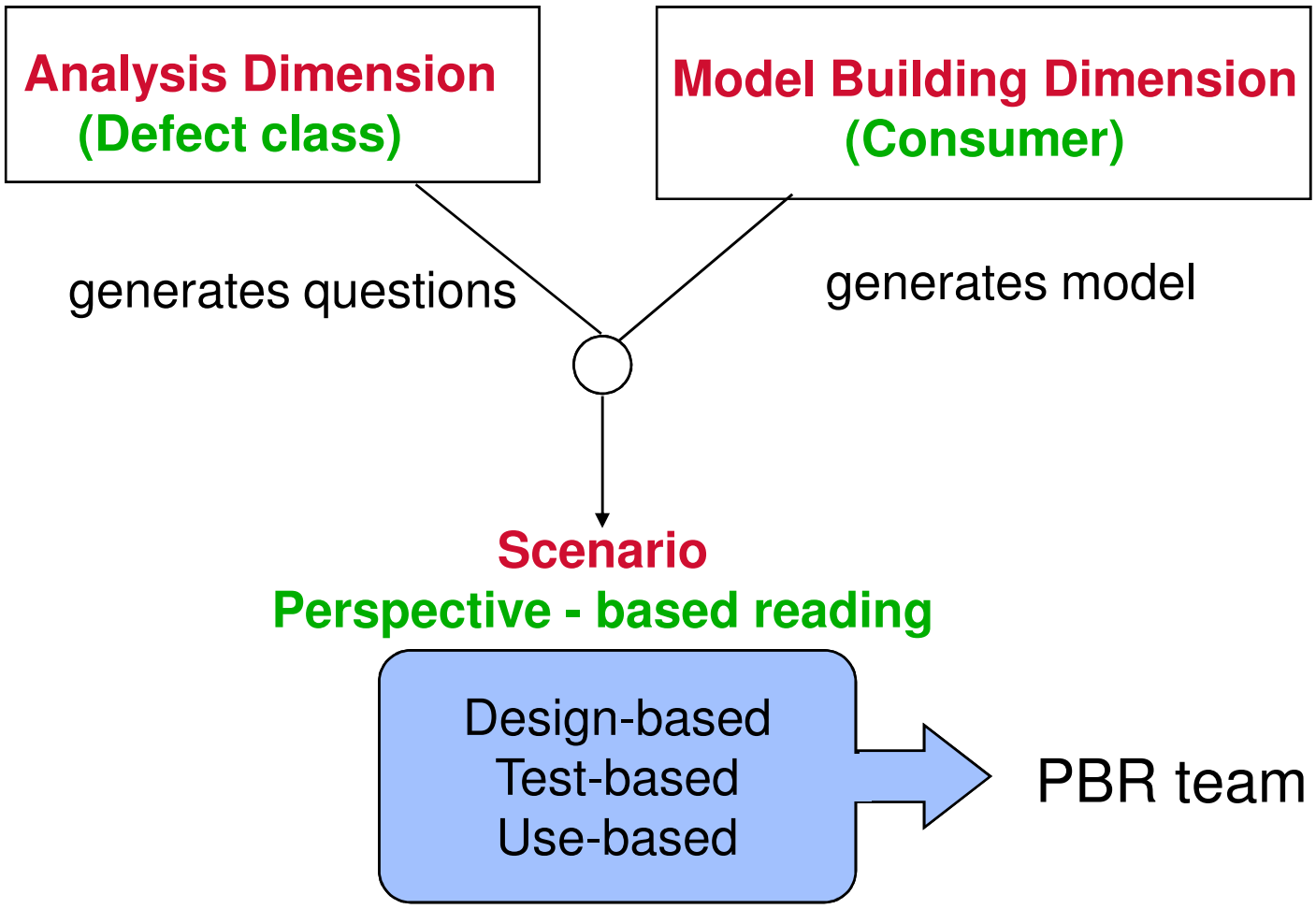
Useful for
quality control,
insights into development
...

Reading for construction:
Given a system,
how do I understand
how to use it as part
of my new system?

Understand
what a system does
what capabilities do and do not exist
...

Useful for
maintenance
building systems from reuse
...

Perspective-based Reading



PBR Example

- **Test-based reading (excerpt):**
- For each requirement/functional specification, generate a test or set of tests that allow you to ensure that an implementation of the system satisfies the requirement/functional specification. Use your standard test approach and technique, and incorporate test criteria in the test suite. In doing so, ask yourself the following questions for each test:

1. Do you have all the information necessary to identify the item being tested and the test criteria? Can you generate a reasonable test case for each item based upon the criteria? Can you be sure that the tests generated will yield the correct values in the correct units?

2. Can you be sure that the tests generated will yield the correct values in the correct units?

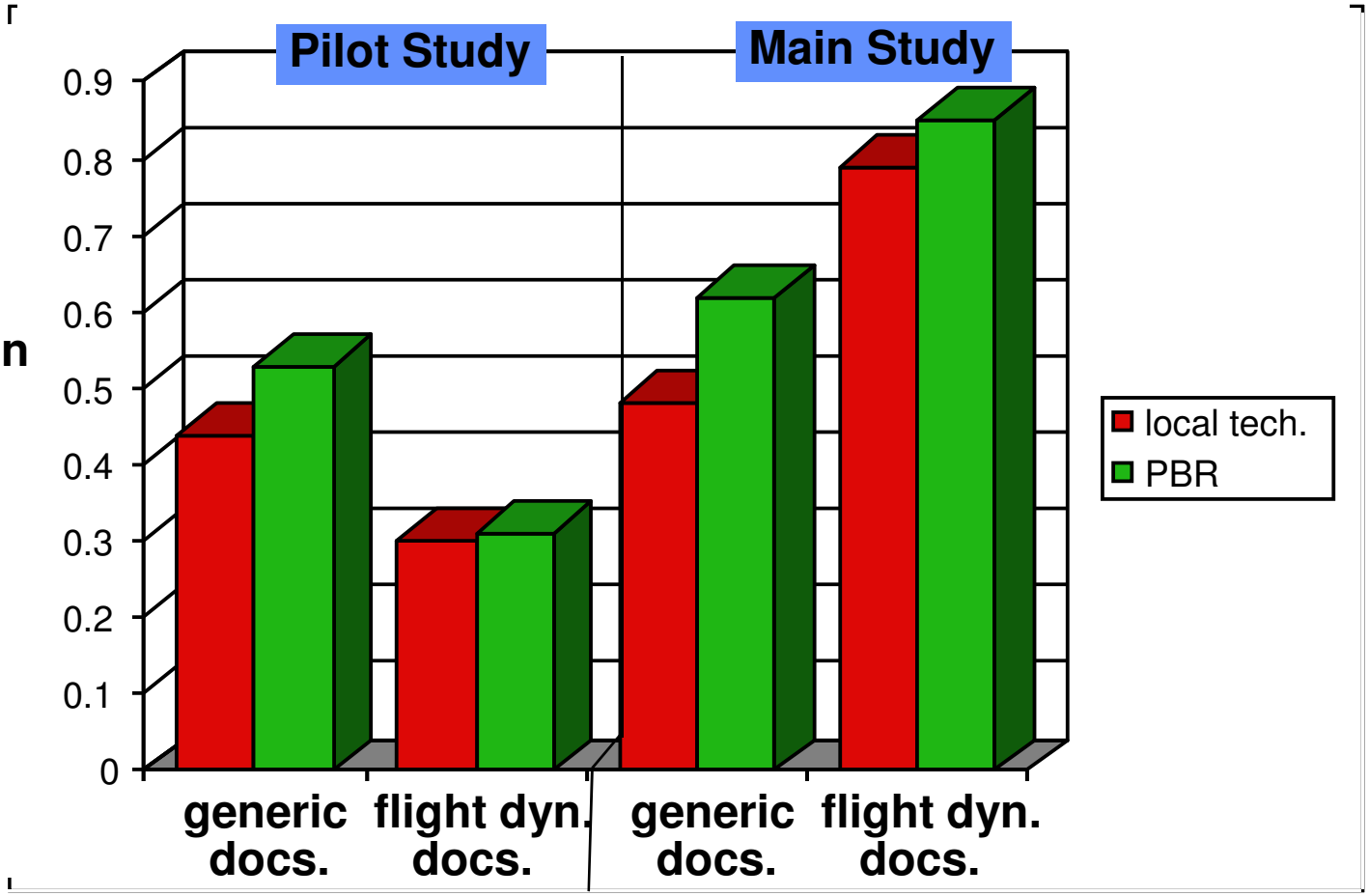
... etc.

Questions for each perspective

Reading for Analysis:
Perspective-Based Reading Experiment

Goal of Perspective-Based Reading (PBR):
detect defects in a requirements document
focus on product consumers

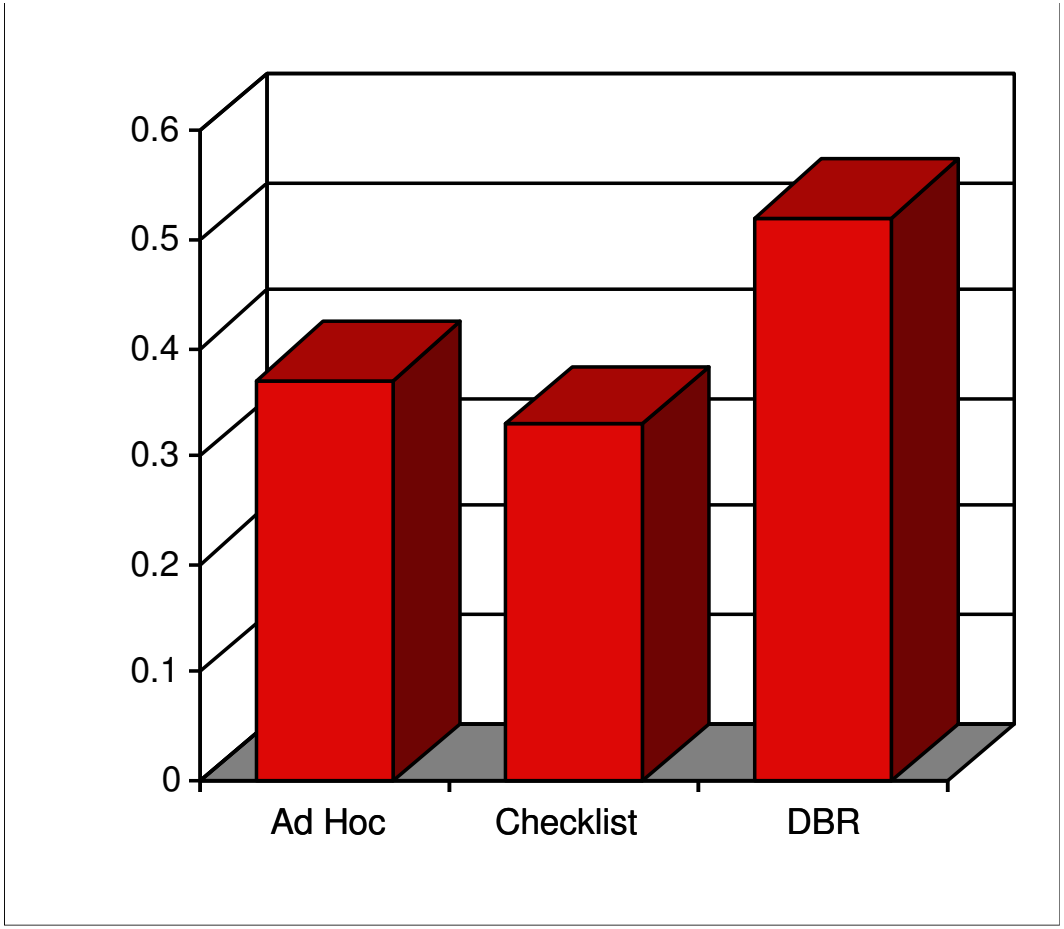
Controlled experiment run twice with NASA professionals:



Reading for Analysis: Defect-Based Reading Experiment

Goal of Defect-Based Reading (DBR):
detect defects in a requirements document
focus on defect classes

Controlled experiment run twice with UMD graduate students:



Reading for Construction

Interested in reading techniques

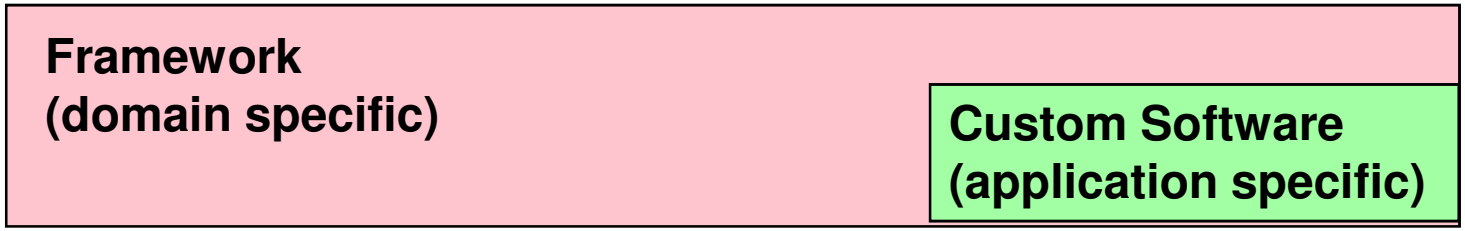
to minimize the effort to learn a new tool or existing system for a specific application development

Framework

A set of classes augmented with a built-in model for defining how classes interact

to reuse domain concepts

to encapsulate implementation details



Two approaches:

White-box frameworks - extend and modify classes

Black-box frameworks - select and configure ready-made classes

Experiments with Reading for Construction

White-Box Frameworks

We proposed two reading techniques for frameworks:

Given the object model of your application and the OO framework

System-wide technique:

- Find the **class** in the **framework hierarchy** that best matches the functionality you are seeking
- Determine **how to parameterize that class** and how to implement it as part of your application

Task-oriented technique:

- Find the **example** in the **example set** that best matches the functionality you are seeking
- Determine **which piece of the example is relevant** and how to implement it as part of your application

Controlled Experiment with UMD students

Experiments with Reading for Construction

Some Results: White-Box Framework Experiment

The effectiveness of an example-based technique is heavily dependent on the quality and breadth of the example set provided.

Example-based techniques are well-suited to use by beginning learners.

A hierarchy-focused technique is not well-suited to use by beginners.

Teams who began their implementation using an existing example for guidance seemed more effective than those who began implementing from scratch.

Teams who were able to stay close to their original object model of the system during implementation seemed more effective.

Next Steps in Developing Techniques and Methods

Study other perspective-based techniques, e.g., **use-case driven perspective**

- Does this perspective find defects not caught by other perspectives?
- Do better defined PBR procedures provide better results?

Study **object oriented design** reading techniques

- scenarios based upon defect classes (UMD)
- scenarios based upon perspectives (Fraunhofer IESE)

Can use-case driven reading technique be used in the context of a product line to help generate **generic use cases** for the product line?

What support processes and tools are necessary?

What other **families of techniques** can we develop

- based on empirical evaluation
- parameterized for use in different contexts

Conclusion

In this research we expect results along several broad directions:

(1) We hope to deliver several families of technologies, starting with analysis technologies, to understand and build different software artifacts;

(2) We hope to deliver better criteria for evaluating software development techniques and methods that can be used by other researchers

(3) We hope to deliver experimental designs for software engineering,a template for storing and interrelating sets of experimental results and an integrated and body of software engineering knowledge

To build a body of useful software engineering knowledge

Building Laboratory Manuals

Laboratory manuals can be used to
- document processes
- provide artifacts
- offer a mix of experimental designs and analysis techniques
- provide a basis for balancing threats to validity
- support meta-analysis

Several Laboratory Manuals already exist
- Reading vs. Testing
- Defect Based Reading
- Perspective Based Reading
- Framework Reading

Several experiments have been replicated
- under the same and differing contexts
- using these manuals

Some progress has been made in doing meta-analysis

Building Laboratory Manuals

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

It can be used to

- help define and replicate studies and techniques
- support the development of evaluation approaches for software engineering
- contribute to the laboratory manuals.