Using Experiments to Build a Body of Knowledge

Victor R. Basili

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

Outline

Evolving Knowledge through Experiments

Choosing a Problem: The Effects of Process

Studying Reading

Specifying the Problem Space

Defining Specific Reading Techniques Solutions

Selecting an Experimental Framework

Some Experimental Results

Conclusions

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**

Evolving Knowledge
In Software Engineering

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 is 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**

This talk is about experimentation in the software discipline

Evolving Knowledge In Software Engineering
The Quality Improvement Paradigm

Characterize the current project and its environment with respect to models and metrics.

Set quantifiable **goals** for successful project performance and improvement.

Choose the appropriate **process** model and supporting methods and tools for this project.

Execute the **processes,** construct the products, collect,validate, and analyze the data to provide real-time feedback for corrective action.

Analyze the **data** to evaluate the current practices, determine problems, record findings, and make recommendations for future project improvements.

Package the **experience** in the form of updated and refined models and other forms of structured knowledge gained from this and prior projects and save it in an experience base to be reused on future projects.

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 they 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 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 the Problem

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 empirically study the effects of processes?

Can we differentiate their effects, measure the differences?

Can we define techniques with different goals and empirically validate that they satisfy those goals?

What are the **high level hypotheses**?

A particular reading technique detects more of a particular class of defect than another reading technique

Research Agenda for Developing Families of Empirically Based Techniques and Methods

This research program is sponsored by NSF

The main issues are to develop:

Families of techniques and methods
based on empirical evaluation
parameterized for use in different contexts
evaluated for those contexts

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

An expanding **Experience Base** of technology evaluations
accessible by researchers and practitioners
who can append their own experiences

Example Technique: Reading Motivation

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

Example Technique: Reading
What had we learned so far?

Several Experiments were run in the SEL:

Code Reading vs Functional Testing vs Structural Testing

Cleanroom (controlled and case studies)

Results supported the investment in Reading

Reading is effective/efficient

The particular technique/procedure appears to be important

The choice of techniques should be tailored to the defect class

Developers don’t believe reading is better

Using a particular technique motivated developers to read better

Better training needed for reading techniques

Need reading techniques for other documents,

e.g., requirements, design, test plans

Choosing the Problem
Specifying the Problem Space

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

Specifying the Problem Space
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**

Choosing the Problem
Specifying the Problem Space

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 variable set

Choosing the Problem
Specifying the Problem Space

We differentiate two high level 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
...

Choosing the Problem
Specifying the Problem Space

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**, …)

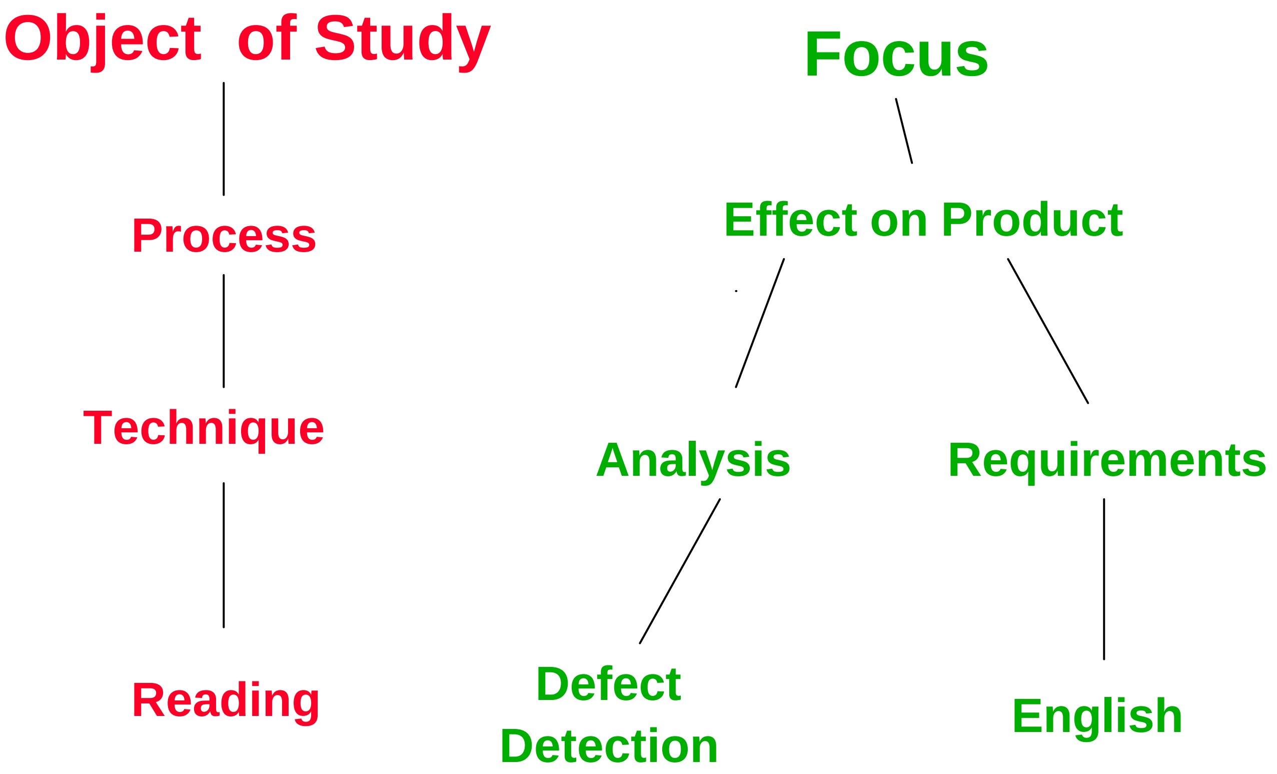
Effectiveness Goal (**Defect Detection**, **Usability**, …

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)

Choosing the Problem
Specifying the Reading Problem Space



Object of Study

Process

Technique

Reading

Focus

Effect on Product

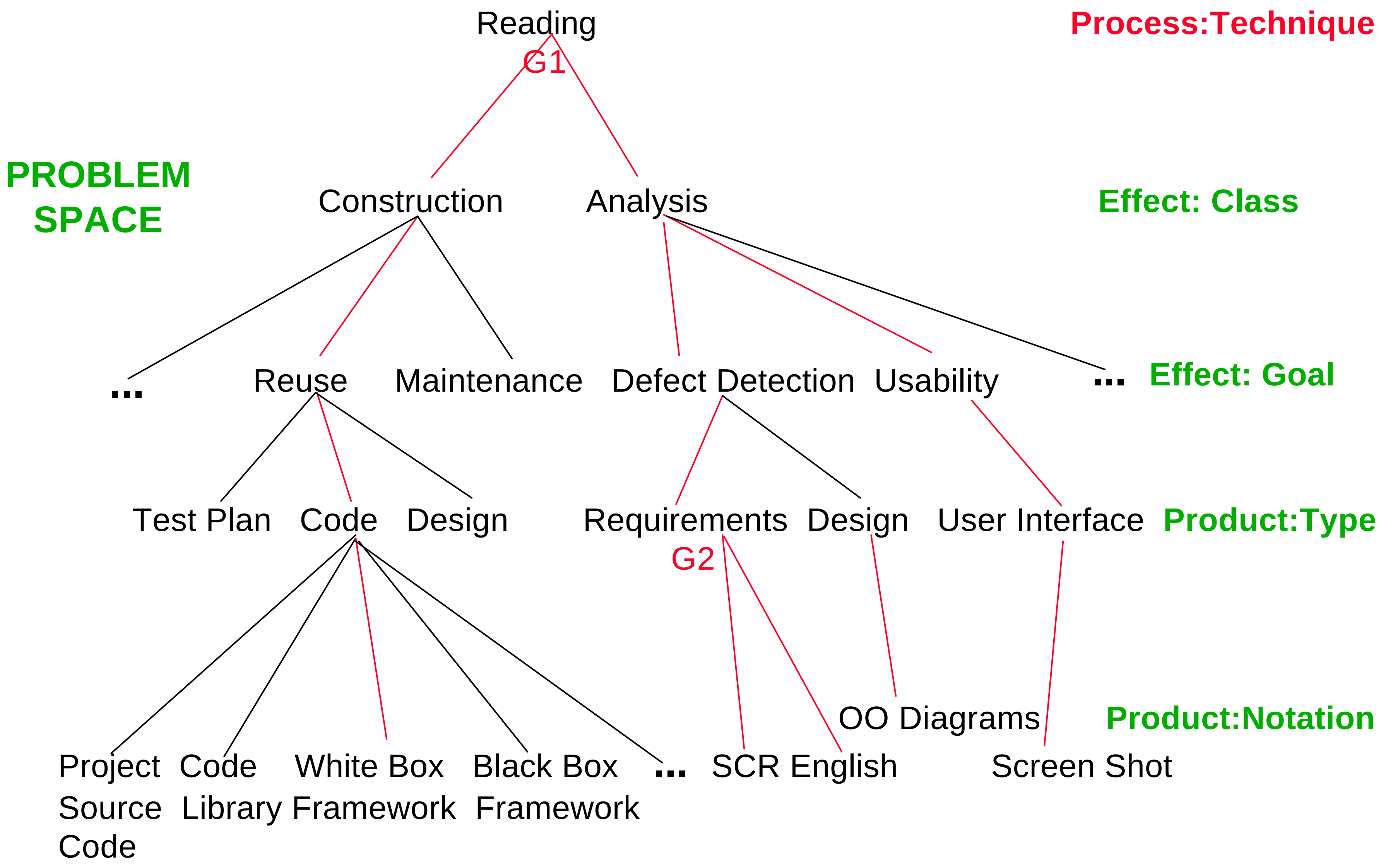
Analysis

Requirements

Defect Detection

English

Specifying the Reading Problem Space



PROBLEM SPACE

Reading G1 — Process:Technique

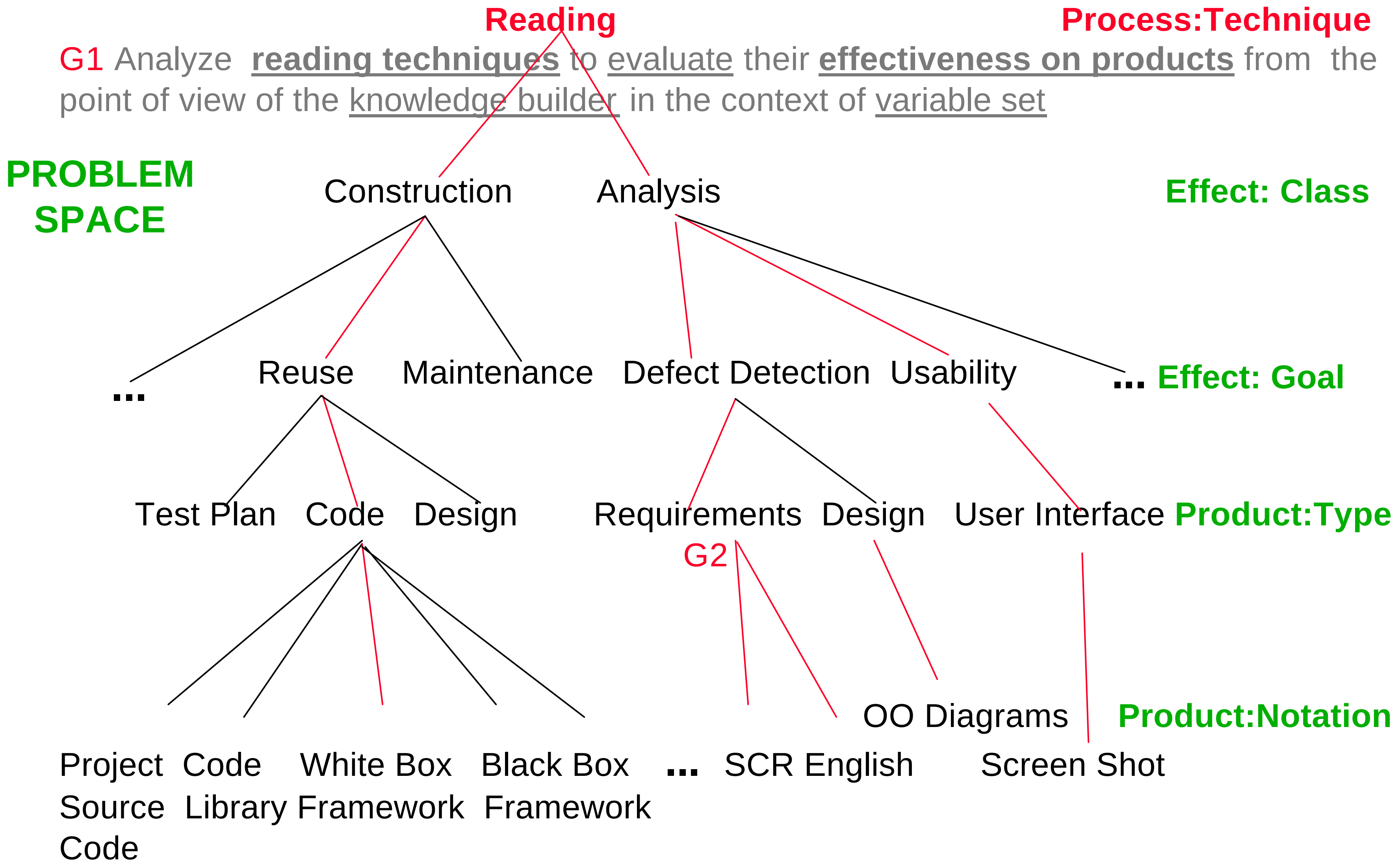
Construction, Analysis — Effect: Class

..., Reuse, Maintenance, Defect Detection, Usability, ... — Effect: Goal

Test Plan, Code, Design, Requirements G2, Design, User Interface — Product:Type

Project Source Code, Code Library, White Box Framework, Black Box Framework, ..., SCR, English, OO Diagrams, Screen Shot — Product:Notation

Specifying the Reading Goals



Reading

Process:Technique

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

PROBLEM SPACE

Construction

Analysis

Effect: Class

...

Reuse

Maintenance

Defect Detection

Usability

... Effect: Goal

Test Plan

Code

Design

Requirements

Design

User Interface

Product:Type

G2

OO Diagrams

Product:Notation

Project Source Code

Code Library

White Box Framework

Black Box Framework

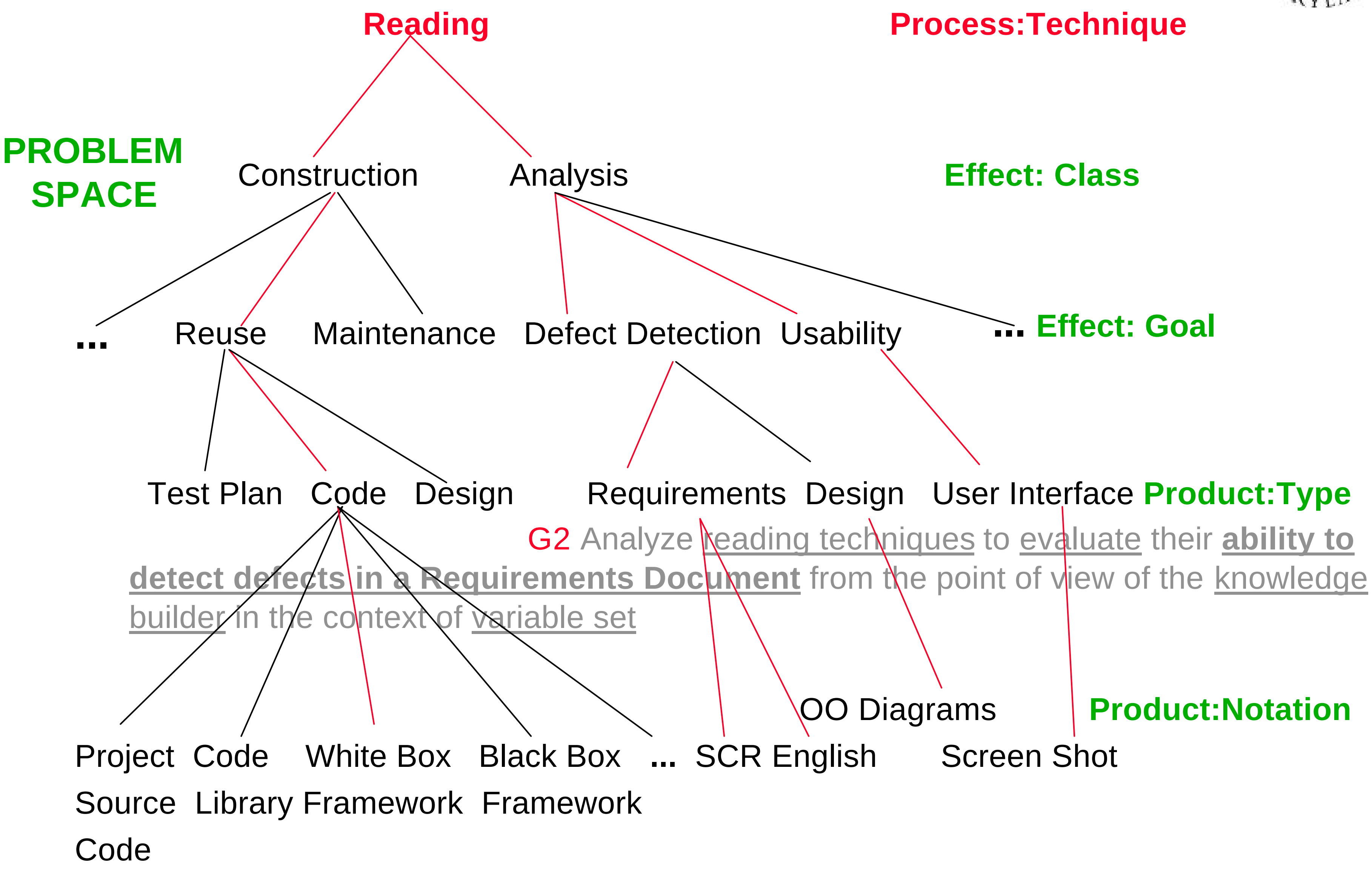
...

SCR

English

Screen Shot

Specifying the Reading Goals



Reading

Process:Technique

PROBLEM SPACE

Construction

Analysis

Effect: Class

...

Reuse

Maintenance

Defect Detection

Usability

... Effect: Goal

Test Plan

Code

Design

Requirements

Design

User Interface

Product:Type

OO Diagrams

Product:Notation

Project Source Code

Code Library

White Box Framework

Black Box Framework

...

SCR English

Screen Shot

G2 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

Specifying a Solution Space
Scenario-Based Reading Techniques

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

Goal: To define families of reading techniques that can are

- document and notation specific
- goal driven
- tailorable to the project and environment
- procedurally defined
- 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

Specifying a Solution Space
Scenario-Based Reading Techniques

Characterize the process:

Technique Class (**Reading**, Testing, Designing, …)

Technique Characteristics (**documentation and notation specific, goal oriented**, **procedurally based**, **coverage focused**, …)

Analyze a **set of goal-oriented, procedurally-based, coverage focused, 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

Specifying a Solution Space
Defining Specific Techniques

So far, we have developed five families of reading techniques
parameterized for use in different contexts and
evaluated experimentally in those contexts

They include:

perspective based reading:
for detecting defects in requirements documents in English

defect based reading:
for detecting defects in requirements documents in SCR

scope based reading:
for constructing designs from OO frameworks

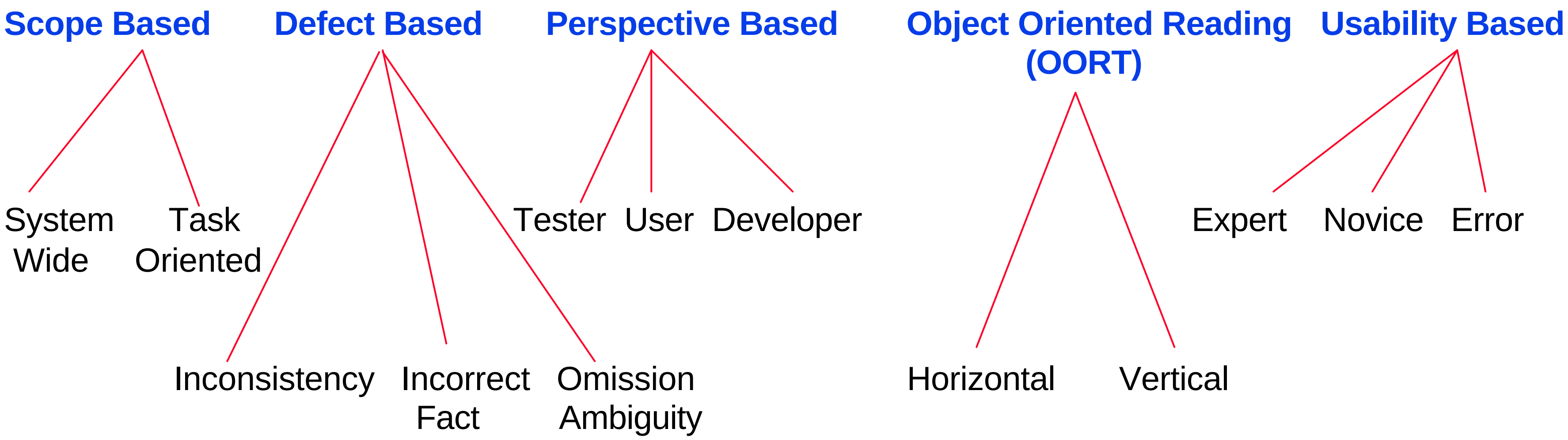
use based reading:
for detecting anomalies in user interface web screens

horizontal/vertical reading:
for detecting defects in object oriented design in UML

Specifying a Solution Space
Defining Specific Techniques

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)

Each family has multiple focuses, procedurally defined, and is tailorable to the context



Scope Based

System Wide

Task Oriented

Defect Based

Inconsistency

Incorrect Fact

Omission Ambiguity

Perspective Based

Tester

User

Developer

Object Oriented Reading (OORT)

Horizontal

Vertical

Usability Based

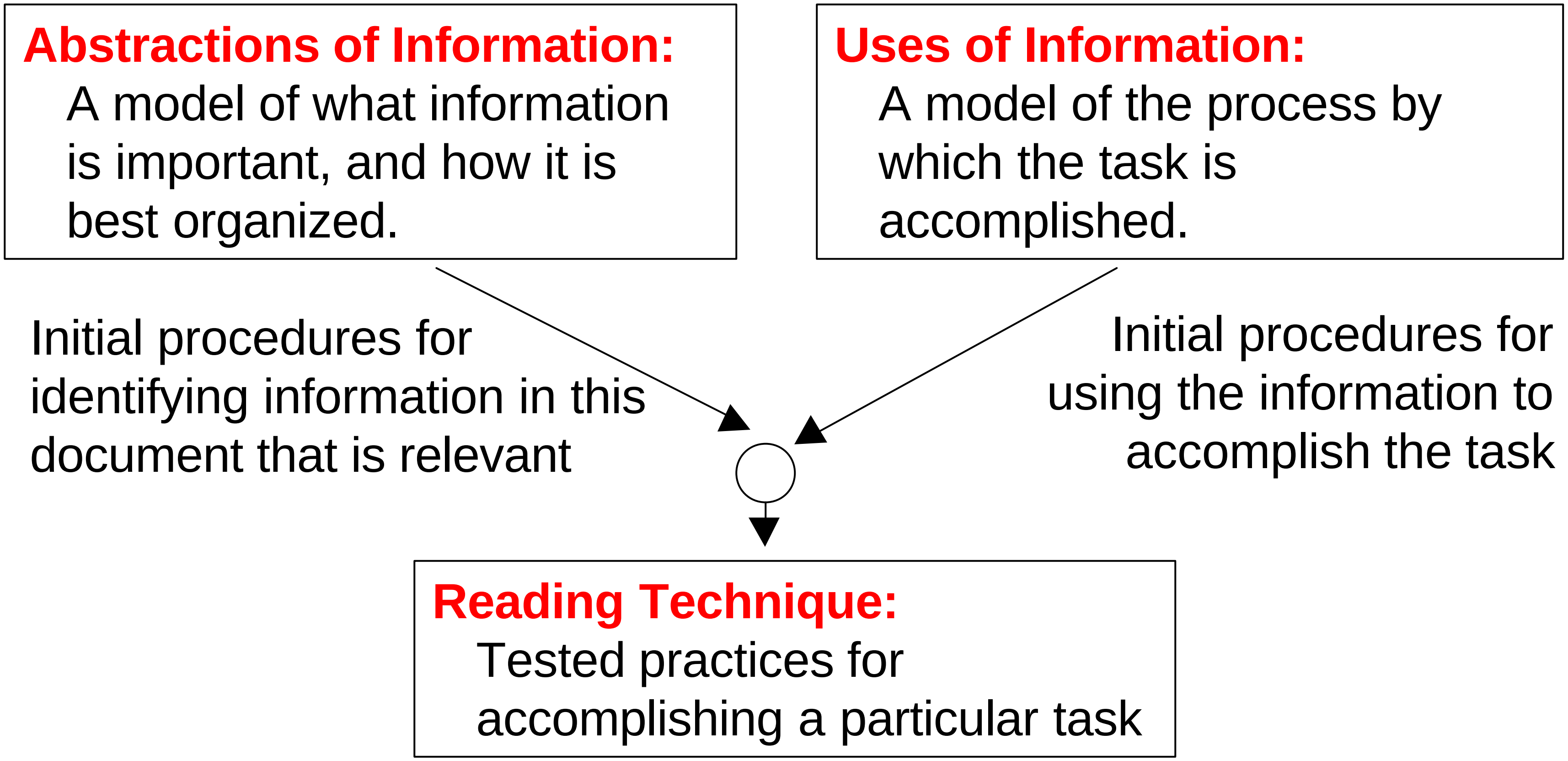
Expert

Novice

Error

Defining a Specific Technique
Mapping Models to a Reading Technique

Need to characterize the “model of use”: how the information in a document is used for a particular task in a particular environment.



Abstractions of Information:
A model of what information is important, and how it is best organized.

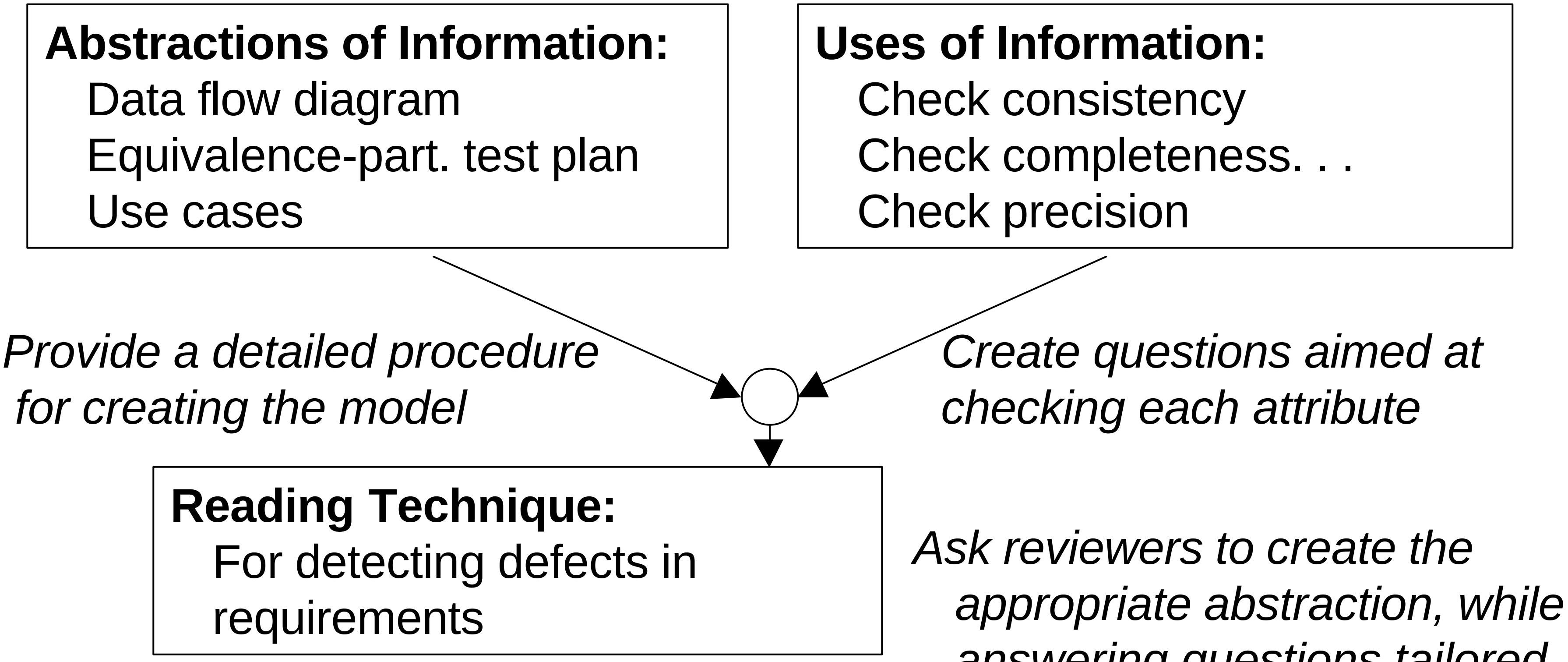
Uses of Information:
A model of the process by which the task is accomplished.

Initial procedures for identifying information in this document that is relevant

Initial procedures for using the information to accomplish the task

Reading Technique:
Tested practices for accomplishing a particular task

Defining a Specific Technique
Perspective-Based Reading



Abstractions of Information:
Data flow diagram
Equivalence-part. test plan
Use cases

Uses of Information:
Check consistency
Check completeness. . .
Check precision

Provide a detailed procedure for creating the model

Create questions aimed at checking each attribute

Reading Technique:
For detecting defects in requirements

Ask reviewers to create the appropriate abstraction, while answering questions tailored to the creation process

Defining a Specific Technique
Perspective-Based Reading

Definition

Various **customers** of a product read it to find out if it satisfies their needs

The reader should find defects and assess the document from their particular point of view.

We used three different perspectives:
test-based reading
use-based reading
designer-based reading

Example: Test-based Reading

For each requirement, make up a test or set of tests that will allow you to ensure that the implementation satisfies the requirement. Use the equivalence partitioning test procedure to make up the test suite.

Reading for Analysis: Perspective-Based Reading

Testing-Based Reading Questions

For each requirement, ask yourself the following questions:

1. Do you have all the information necessary to identify the item being tested and to identify your test criteria? Can you make up reasonable test cases for each item based upon the criteria?

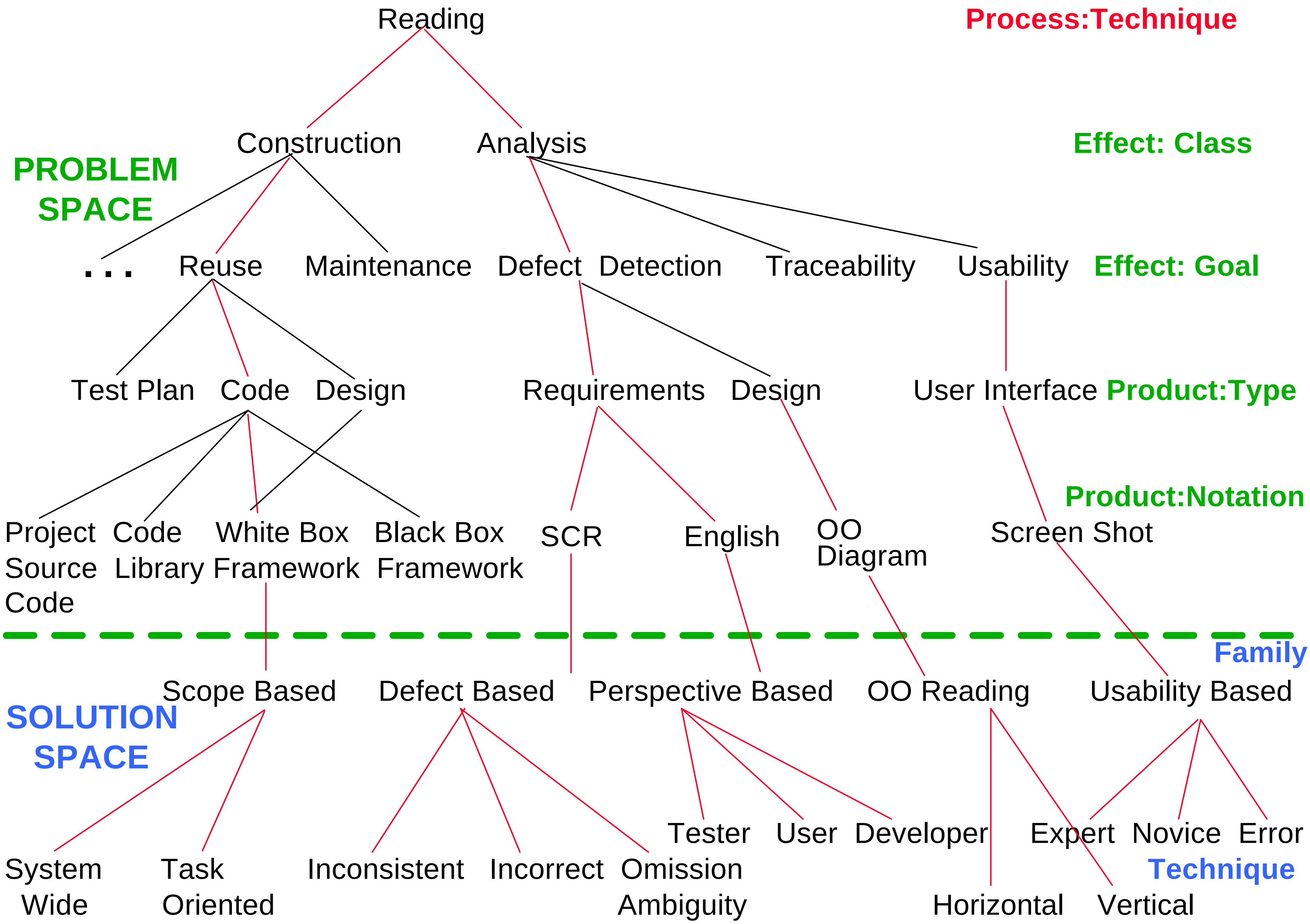
2. Is there another requirement for which you would generate a similar test case but would get a contradictory result?

3. Can you be sure the test you generated will yield the correct value in the correct units?

4. Are there other interpretations of this requirement that the implementor might make based upon the way the requirement is defined? Will this effect the test you made up?

5. Does the requirement make sense from what you know about the application and from what is specified in the general description?

Families of Reading Techniques



Reading

Process:Technique

Construction Analysis

Effect: Class

PROBLEM SPACE

. . . Reuse Maintenance Defect Detection Traceability Usability

Effect: Goal

Test Plan Code Design Requirements Design User Interface

Product:Type

Product:Notation

Project Source Code

Code Library

White Box Framework

Black Box Framework

SCR

English

OO Diagram

Screen Shot

Family

Scope Based Defect Based Perspective Based OO Reading Usability Based

SOLUTION SPACE

System Wide

Task Oriented

Inconsistent Incorrect Omission Ambiguity

Tester User Developer

Horizontal Vertical

Expert Novice Error

Technique

Defining Specific Techniques Choosing 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 models for each perspective

Choose defect classes

Choose experimental treatment

etc.

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

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?

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**

subject experience

context (classroom, toy, off-line, in project)

variability among subjects

Vary order of events and activities

Allows us to balance threats to validity

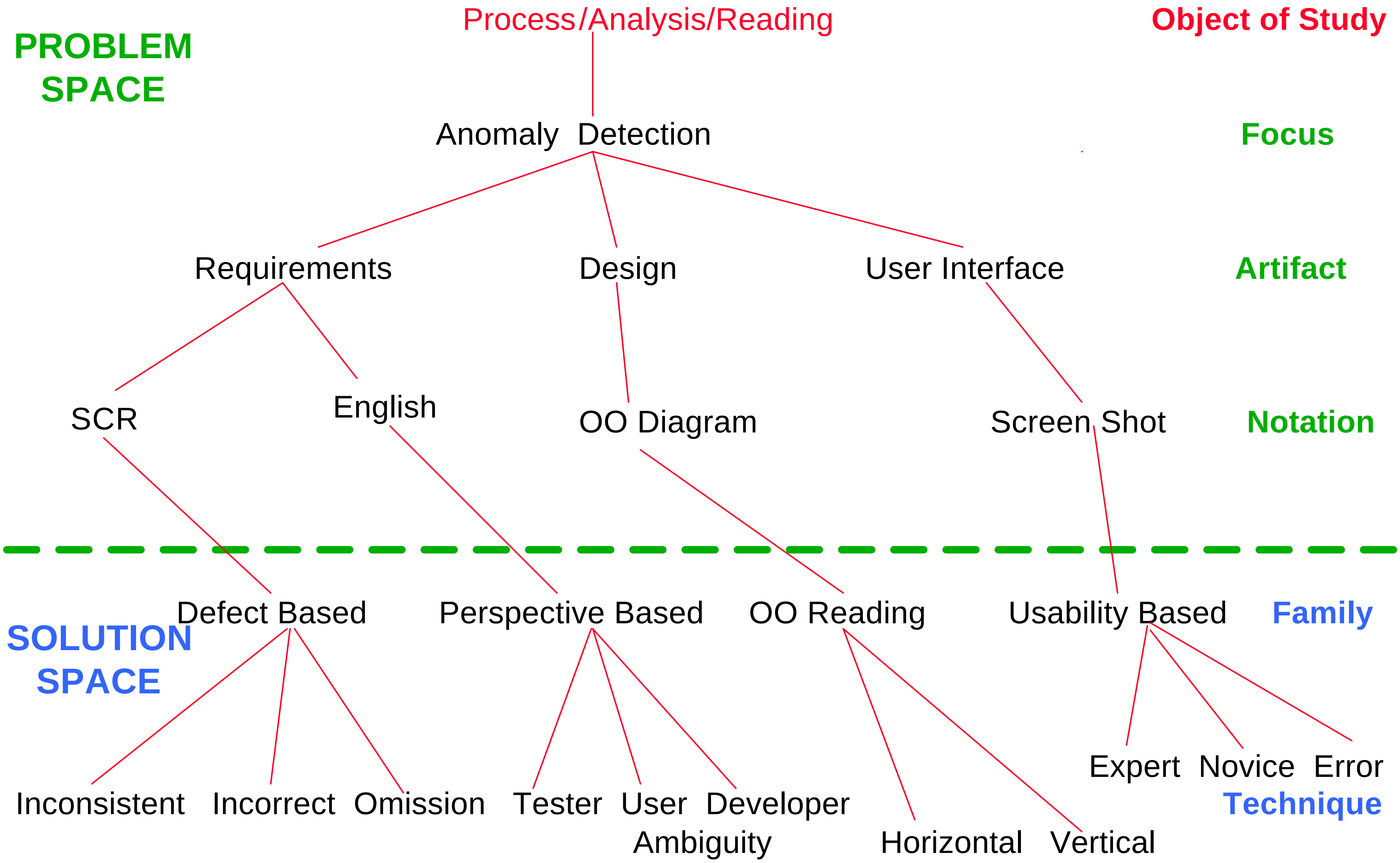
interaction of experience and treatment

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)



Process/Analysis/Reading

Object of Study

PROBLEM SPACE

Anomaly Detection

Focus

Requirements Design User Interface

Artifact

SCR English OO Diagram Screen Shot

Notation

Defect Based Perspective Based OO Reading Usability Based

Family

SOLUTION SPACE

Expert Novice Error

Inconsistent Incorrect Omission Tester User Developer

Technique

Ambiguity Horizontal Vertical

Sample Set of Experiments

We have run several experiments

- on all five families of reading techniques
- parameterized for use in different contexts
- some involved us as directly as experimenters, others did not

Example Contexts: (Government, University, Industry)

- NASA/GSFC (PBR)
- UM Professional SE Course (PBR, UBR)
- UM Students (DBR, UBR, SBR)
- Bureau of Census (UBR)
- Robert Bosch (PBR)
- Lucent (DBR)

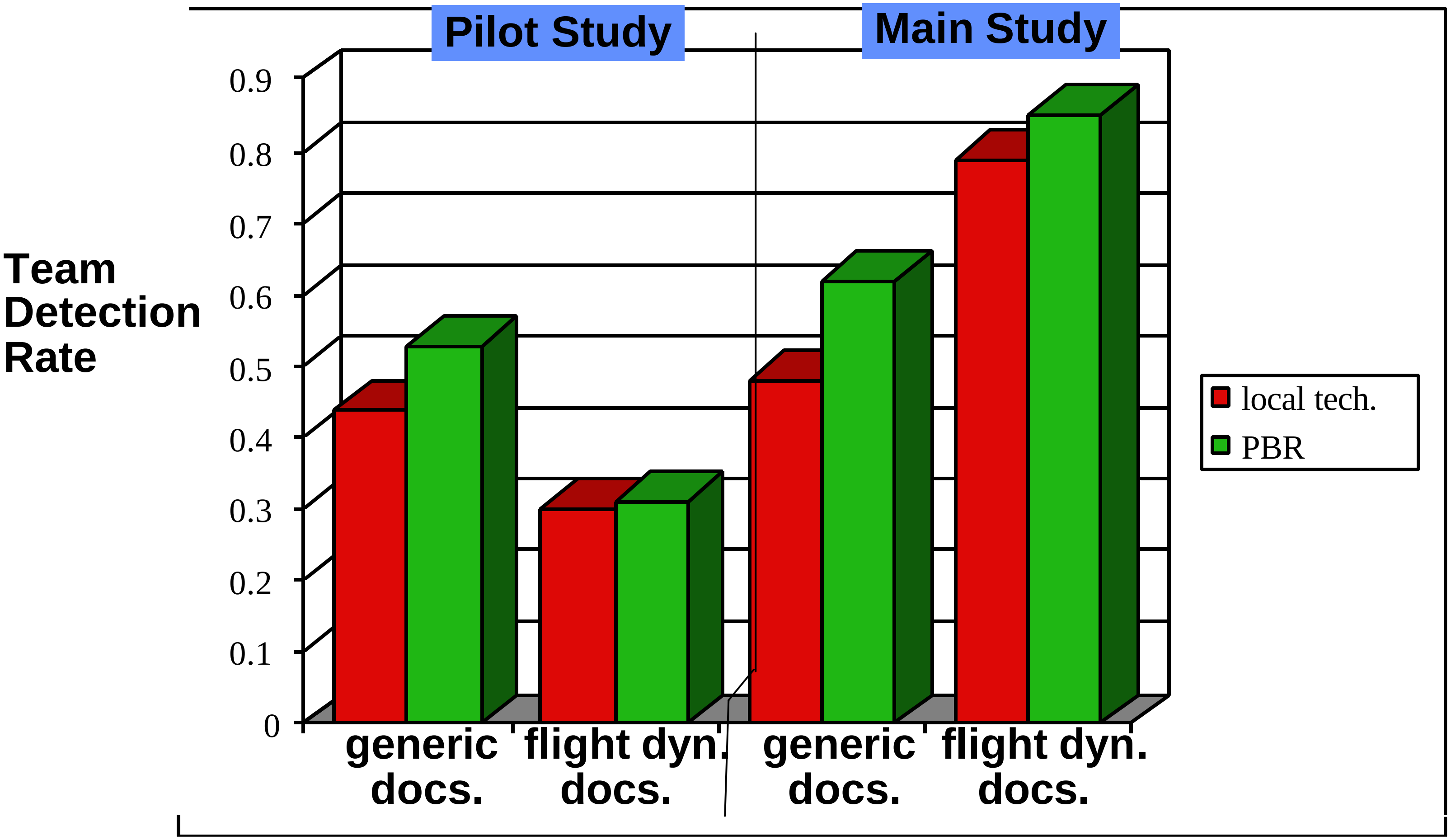
Example Countries: (U.S., Germany, Italy, Sweden, Scotland, Norway,...)

Reading for Analysis: Perspective-Based Reading (PBR) Experiment

PBR: Technique to detect defects in a requirements document in English

Goal of Experiment: Compare PBR with existing reading technique

Controlled experiment run twice with NASA professionals:

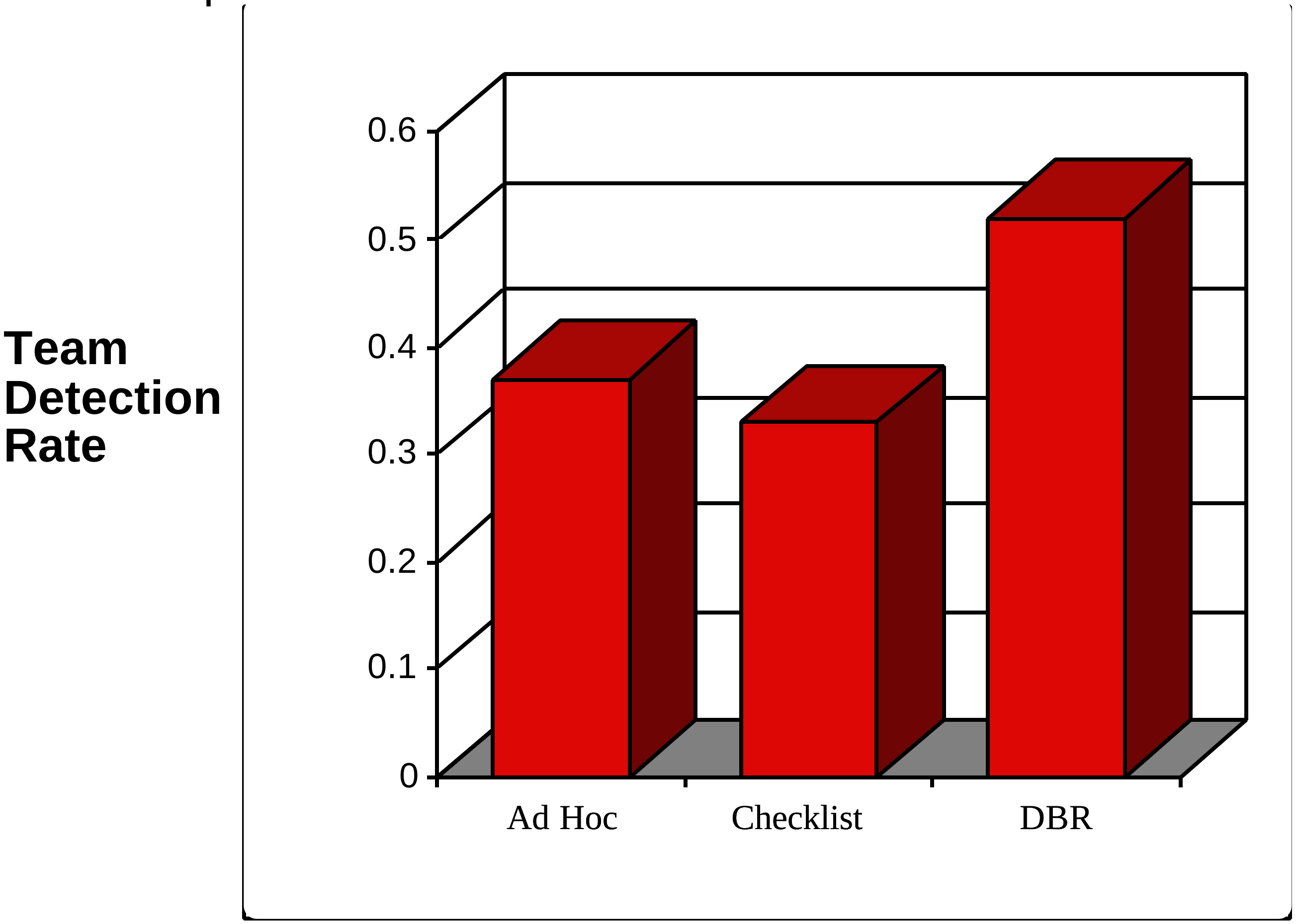


Reading for Analysis: Defect-Based Reading (DBR) Experiment

DBR: Technique to detect defects in a requirements document in SCR notation

Goal of Experiment: Compare DBR with ad hoc and checklist reading

Controlled experiment run twice with UMD graduate students:

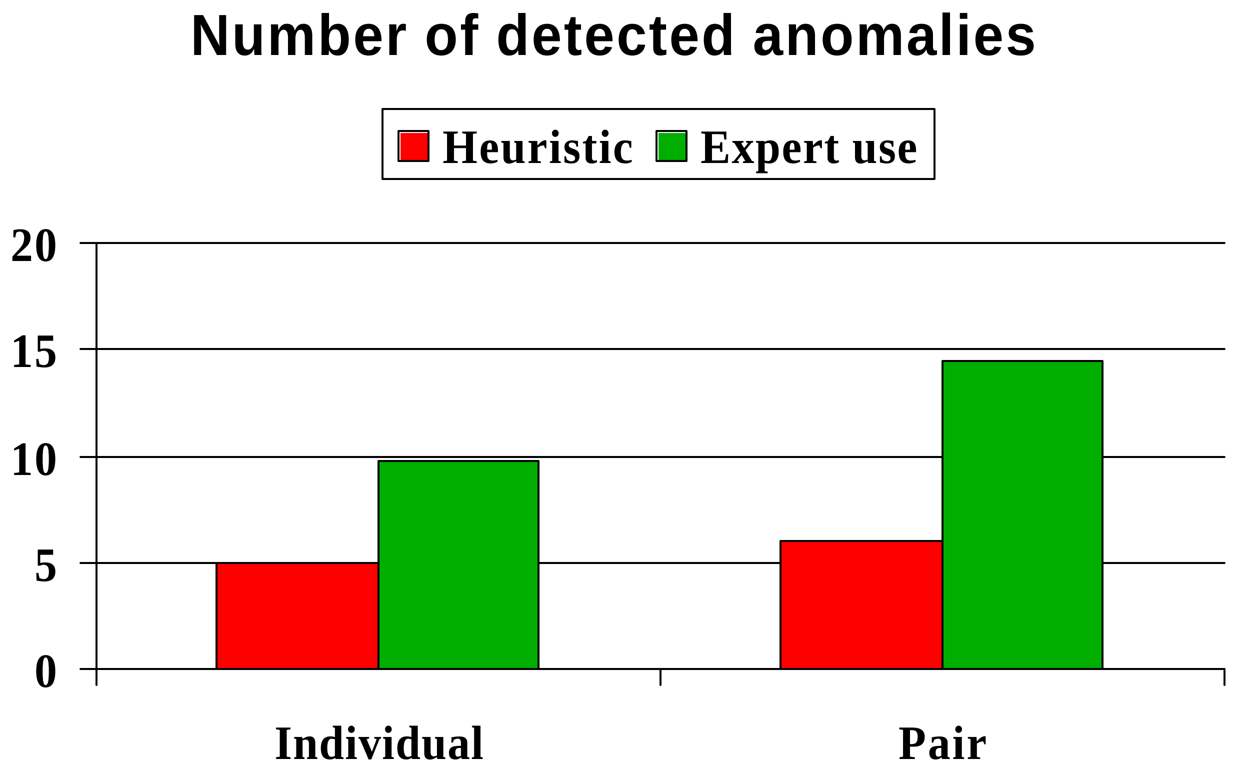


Reading for Analysis: Use-Based Reading (UBR) Experiment

UBR: Technique to detect anomalies in a user interface in SCR notation

Goal of Experiment: Compare UBR with heuristic inspection alone and in pairs

Controlled experiment run twice with UMD graduate students and Bureau of Census:



Reading for Analysis: Use-Based Reading Experiment

Major Results

Use-based (Perspective-based) reading, compared to heuristic evaluation

- did not require more inspection time
- got a better or equivalent preference rating by performers
- was more effective in detecting related anomalies,
- overall found more anomalies
  - at individual level,
  - for paired application of the procedures, and
  - for simulated teams

The effectiveness was consistently shown with different subjects, interfaces, and time constraints

Paired teams performed better than individuals

Reading for Construction
Defining a Specific Technique

Define reading techniques

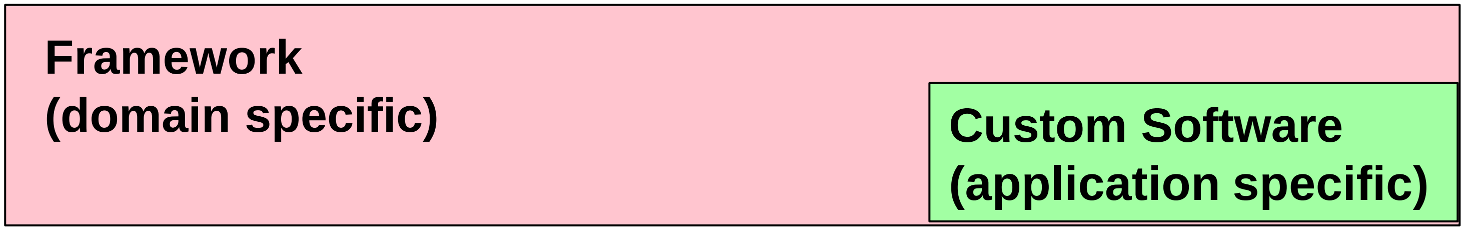
to minimize the effort to learn a new tool or existing system for a specific application development

Framework

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

to reuse domain concepts

to encapsulate implementation details



Framework (domain specific)

Custom Software (application specific)

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

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.

Current work in Reading Techniques

Other perspective-based techniques, for requirements, specification, code

- e.g., **mutation testing perspective**
- Do these perspectives find defects not caught by other perspectives?

Object oriented design reading techniques

- Various models based upon horizontal and vertical reading rules
- Can this be a guide for better OO design?

Comparing **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?

Current work in Reading Techniques

Other ways of defining the process, i.e., **process specification**

- Do more specific procedures provide better results?
- Should we define constraints rather than procedures?

Other ways of assuring **process conformance**

- Does combining a guide/observer and a doer create more effective processes?

How do you modify reading techniques to evaluate **evolving artifacts**?

We are using experiments to help evolve the techniques

What support tools can be used?

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

Contributors to This Work

Jeff Carver, University of Maryland

Scott Green, NASA Goddard Space Flight Center

Oliver Laitenberger, Fraunhofer Institute, Kaiserslautern

Filippo Lanubile, University of Bari

Adam Porter, University of Maryland

Forrest Shull, Fraunhfoer Center, Maryland

Sivert Sorumgaard, Norwegian University of Science and Technology

Guilherme Travassos, University of Maryland

Larry Votta, Lucent Bell Laboratories

Marvin Zelkowitz, University of Maryland

Zhijun Zhang, University of Maryland

Epitaph

I survived EWSPT 2000