Better Empirical Science for Software Engineering

How not to get your empirical study rejected: we should have followed this advice

Victor Basili

University of Maryland 1856

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Sebastian Elbaum

University of Nebraska Lincoln

Invited Presentation
International Conference on Software Engineering May 2006

Motivation for this presentation

- □ There is not enough good empirical work appearing in top SE conference venues

- □ Our goal is to help authors and reviewers of top SE venues improve this situation

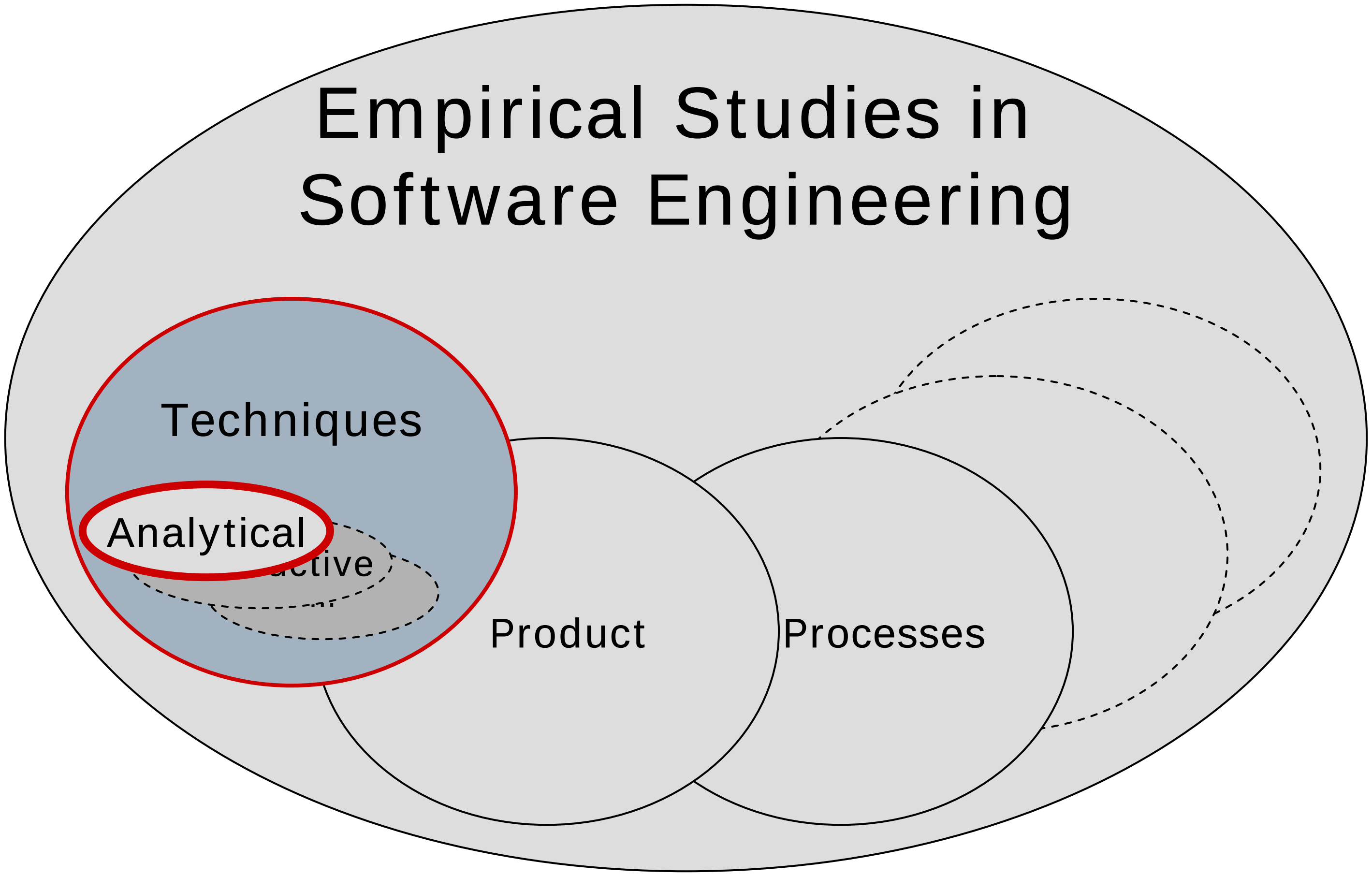
Presentation structure

- □ Discuss the state of the art in empirical studies in software engineering
- □ Debate problems and expectations for papers with empirical components in top SE conference venues

What is an empirical study?

Empirical study in software engineering is the scientific use of quantitative and qualitative data to understand and improve the software product and software development process.

What are we studying?



Why study techniques empirically?

- Aid the technique developer in
  - Demonstrating the feasibility of the technique
  - Identifying bounds and limits
  - Evolving and improving the technique
  - Providing direction for future work
- Aid the user of the technique in
  - Gaining confidence of its maturity for context
  - Knowing when, why and how to use it
- To learn and build knowledge

How to study a technique?

1. Identify interesting problem
2. Characterize and scope problem (stakeholders, context, impact, ...)
3. Select, develop, or tailor techniques to solve a part of problem
4. Perform studies to assess technique on a given artifact (feasibility, effectiveness, limits,...)
5. Evolve the studies (vary context, artifacts, ... and aggregate)

Repeat steps as necessary and disseminate results!

Why is repetition necessary?

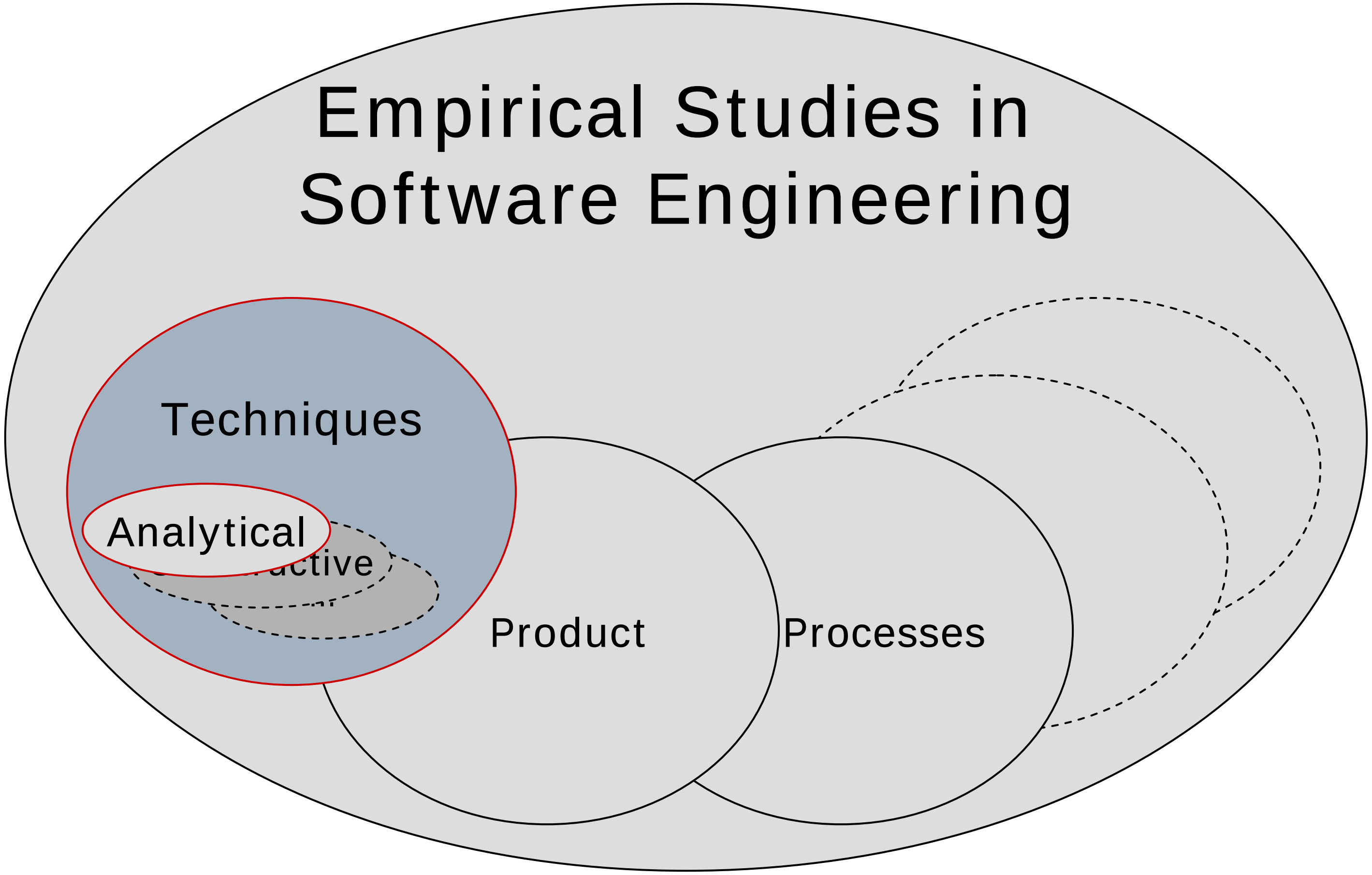
- Need accumulative evidence
  - Each study is limited by goals, context, controls, ...
  - Families of studies are required
    - Varying goals, context, approaches, types of studies, ...
    - Increase confidence, grow knowledge over time

- Need to disseminate studies
  - Each paper is limited by length, scope, audience, ...
  - Families of papers are required
    - Gain confidence through replications across community
    - Move faster or more meaningfully by leveraging existing work to drive future research

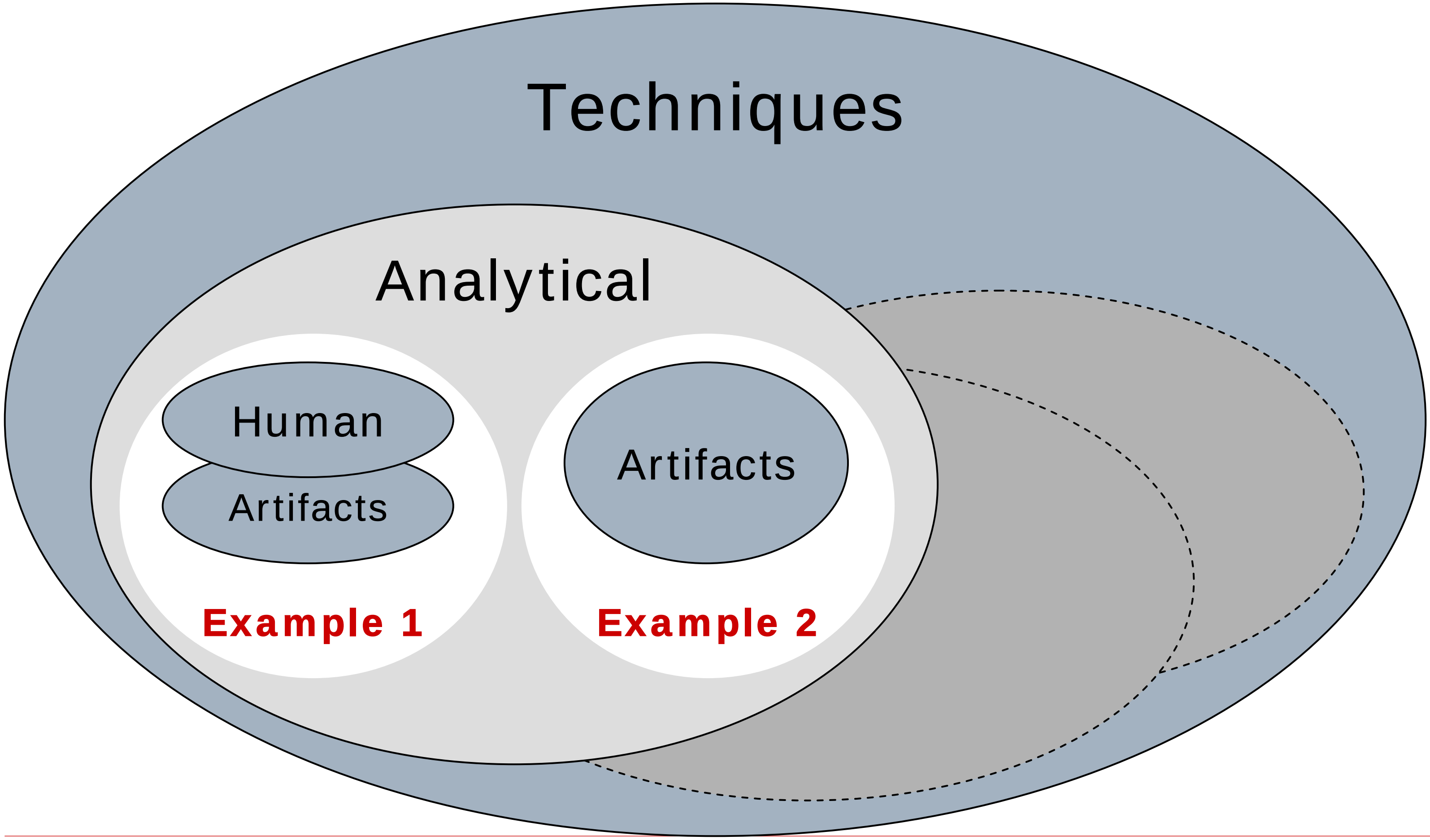
Studies of Techniques
Large variation across community

- Is the human part of the study?
- What are the bounds on sample size?
- What is the cost per sample?
- What are the interests, levels of abstraction, model building techniques?
- What types of studies are used, e.g., qualitative, quantitative, quasi-experiments, controlled experiments?
- How mature is the area?

Studies of Techniques
Two Examples



Studies of Techniques
Two Examples



Example 1: Human Based Study on an analytic technique

Evaluating a code reading technique

Initial version: rejected for ICSE 1984

Invited Talk: American Statistical Association Conference, July 1984

Published TSE 1987 (after much discussion)

A study with human subjects
Question and Motivation

- Is a particular code reading technique effe
  - Is
  - How does it compare to various testing te
  - W cover?
  - What is the effect of experience, product type, ...?

State clearly what questions the investigation is intended to address and how you will address them, even if the study is exploratory.

Try to design your study so you maximize the number of questions asked in that particular study, if you can.

A study with human subjects
Context and Population

Environment:

NASA/CS

Text forn database

Seeded w

145 - 36

Specify as much context as possible... this is often hard to do so in a short conference paper.

Experiment

Fractiona

Three applications

74 subjects: 32 NASA/CSC, 42 UM

Student studies offer a lot of insights. This led to new questions for professional developers.

A study with human subjects

Variables and Metrics

Independent (the technique)

Code Readin

Given: Spec

Functional Te

Given: Spec and Executables

Structural Testing: % statement coverage

Given: Source, Executables, Coverage tool, then spec

Technique definition and process conformance need to be carefully specified in human studies.

Dependent (effectiveness)

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

A study with human subjects

Controlling Variation

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

The more people you can get to review you design, the better.
It is easy to miss important points.

It is easy to contaminate subjects.
It is hard to compare a new technique against the current technique.

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

A study with human subjects
Quantitative Results (NASA/CSC)

□ Fault Detection Effectiveness
- Code reading > (functional > structural)

Student Study had weaker results but showed similar trends.

□ Fault
- Code reading > (functional ~ structural)

□ Classes of Faults Detected
- Interface:
  - □ code reading > (functional ~ structural)
- Control:
  - □ functional > (code reading ~ structural)

A study with human subjects

Qualitative Results (NASA/CSC)

- Code r ated their perfor
- Particip ing worked best
- When inspections were applied on a live project t, if any
- Threat to Validity:
  - External Validity: Generalization, interaction of er
- Stu
  - 32 professional programmers for 3 days

Empirical studies are important even when you believe the results should be self-evident.

It may be difficult to generalize from in vitro to in vivo.

Human subject studies are expensive. You cannot easily repeat studies.

A study with human subjects

New Ideas (NASA/CSC)

- Reading using a defined technique is more effe ecific test
- Different techniques may be more effect ts
- The re
- The reading technique may be different from the reading method

It is important to make clear the practical importance of results independent of the statistical significance.

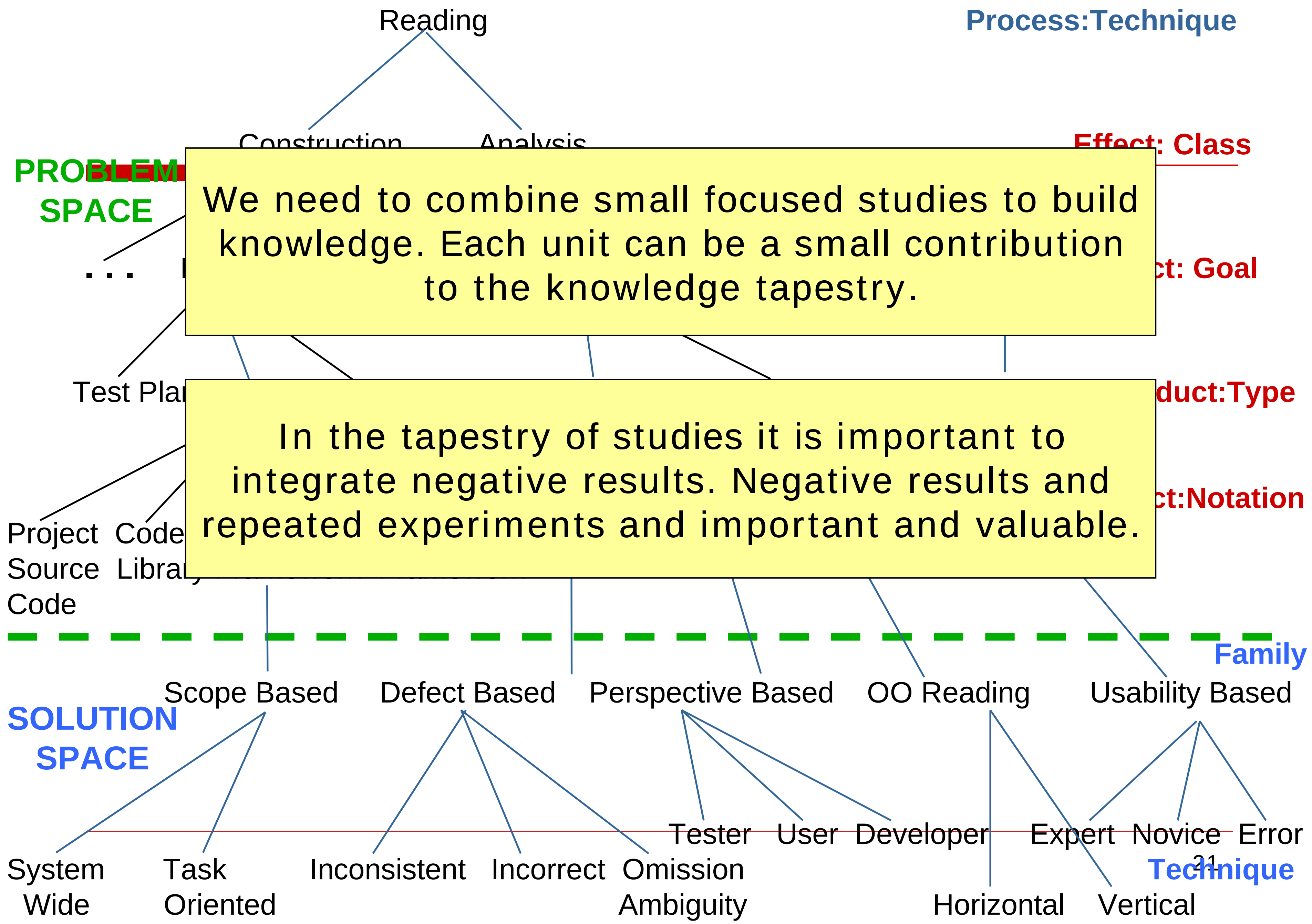
Don’t expect perfection or decisive answers. For example, insights about context variables alone are valuable.

Studies with human subjects
Evolution of Studies

Each study opens new questions. Scaling up is difficult and the empirical methods change.

		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. ...**

Evolution of Studies: Families of Reading Techniques



Reading

Process:Technique

Construction

Analysis

Effect: Class

PROBLEM SPACE

. . .

ct: Goal

Test Pla

duct:Type

ct:Notation

Project Source Code

Code Libra

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

We need to combine small focused studies to build knowledge. Each unit can be a small contribution to the knowledge tapestry.

In the tapestry of studies it is important to integrate negative results. Negative results and repeated experiments and important and valuable.

Example 2:
Artifact Based, Analytic

The Impact of Test Suite Granularity on the
Cost Effectiveness of Regression Testing
(ICSE 2002)

Evaluating the effects of test suite composition
(TOSEM 2004)

A study with artifacts
Question and Motivation

- How do we compose test suites?

Separate believes from knowledge.

- Wh

Experience can help to shape interesting and meaningful conjectures.

Bor s *tests than to do the job with fewer, grander tests.*

Cem Kaner: *Large tests save time if they aren't too complicated; otherwise, simpler tests are more efficient.*

James Bach: *Small tests cause fewer cascading errors, but large tests are better at exposing system level failures involving interactions.*

A study with artifacts

Context and Population

- □ Con
  - ■ D sion)
    - □ What tests should we re-run?
    - □ In what order should we re-run them?

Identify context that is likely to have greatest impact!

- □ Pop
  - ■ T
    - □
  - ■ Seeded faults
    - □ Non-seeded versions were the oracles
  - ■ Test suite
    - □ Original + enhanced

We do not have a good idea of our populations...
but this should not stop us from specifying scope of findings.

A study with artifacts

Type of Study

- Family of controlled experiments
  - Conjectures should lead to more formal and (likely more constrained) hypotheses.

    ng
  - Measure effects on
  - Carefully identify and explain dependent, independent, and fixed variables.
    - *Does granularity and grouping matter?*

- High levels of controls
  - Process, execution, replicability

A study with artifacts

Controlling sources of variation

- Controlled manipulation
  - *of tests*

Controlling is not just about the chosen experimental design, is also about controlling noise so that we really measure the desired variables.

1. Start with a given test suite
2. Partition in test grains
3. To generate test suite of granularity **k**

   Select **k** grains from pool

A study with artifacts

Controlling sources of variation

- Experimental designs
  - R
  - M ity, g els, ...

Once automated, application of treatment to units is inexpensive. We can get many observations quickly and inexpensively.

Test Case Selection

Test Case Prioritization

All

Feedback

Granula

Granularity

G1 G2 G4

G1 G2 G4 G8 G16

Provide detailed definition of data collection process, including costs and constrains that justify choices.

Empire
(10 versions)

Bash
(10 versions)

A study with artifacts

Analysis and Results

- Analysis
  - iation

Richness of results may be in interactions between factors. Question is not really about “does it matter?” but “when does it matter?”

- Results
  - ime
    - Very fine granularity – it enabled better test case selection/prio.
  - Test suite fault detection effectiveness improved at
    - Coarse granularity but only for easy-to-detect faults
    - Fine granularity when faults were detected by single grains

Combine exploratory and formal data analysis.

A study with artifacts

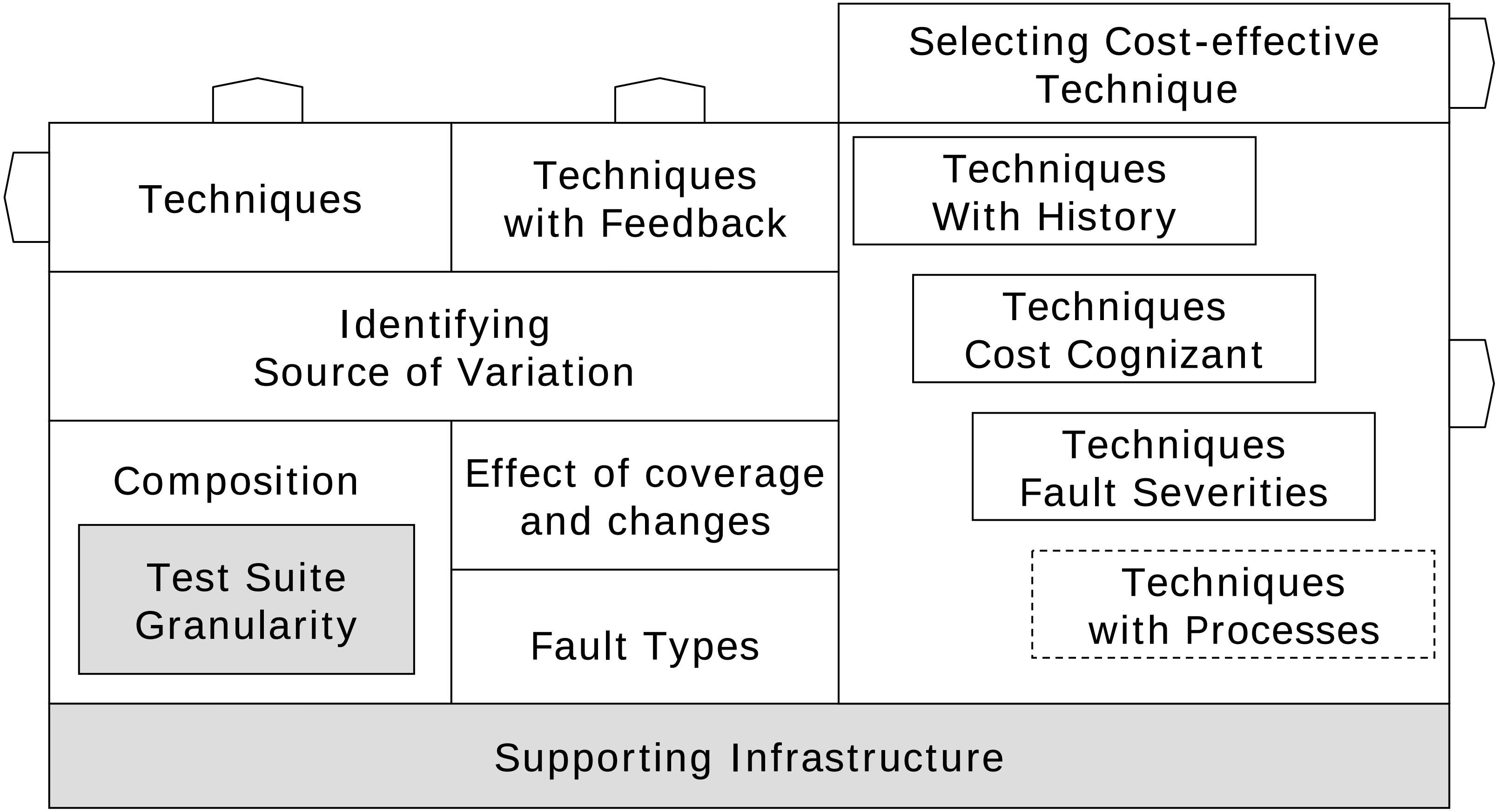
Qualified Implications

- Test suite comp. mattered, specially for extremes
- But
  - Hard-to-detect faults
  - Aggressive test case selection or reduction techniques
- Threats
- Generalizations
  - Early testing, significant program changes: coarser suites
  - Mature stage, stable product: finer granularity

Keep “chain of significance” throughout the paper. Close with “distilled implications”.

A study with artifacts

Building a Family for Regression Test Case Prioritization



Selecting Cost-effective Technique
Techniques
Techniques with Feedback
Techniques With History
Techniques Cost Cognizant
Identifying Source of Variation
Techniques Fault Severities
Composition
Effect of coverage and changes
Techniques with Processes
Test Suite Granularity
Fault Types
Supporting Infrastructure

A 6 year lifespan, over 15 researchers from many institutions, building knowledge incrementally.

Looking at Some Recurring Issues

- What is the target and scope?
- What is representative?
- What is an appropriate sample?
- What are the sources of variation?
- What infrastructure is needed?

Recurring Issues

What is the target and scope?

- With humans
  - Effect of people applying technique
  - Costly. Little margin for error in a single study
  - Hard to replicate, context variables critical
- With artifacts
  - Effect of technique on various artifacts
  - Summative evaluations, confirmatory studies
  - Replicable through infrastructure/automation

Recurring Issues
What is representative?

- **With humans**
  - Participants' ability, experience, motivation, ...
  - Technique type, level of specificity,...
  - Context for technique application
- **With artifacts and humans**
  - Product: domain, complexity, changes, docs, ..
  - Fault: actual or seeded, target, protocols, ...
  - Test Suite: unit or system, original or generated,
  - Specifications: notation, type of properties, ...
  - ...

Recurring Issues

What is an appropriate sample?

- **With humans**: mostly opportunistic
  - Small data samples
  - Learning effect issues
  - Unknown underlying distributions
  - Potentially huge variations in behavior
- **With artifacts**: previously used artifacts/testbeds
  - Reusing “toy” examples to enable comparisons
  - Available test beds for some dynamic analysis
  - Not natural occurring phenomenon

Recurring Issues

What are the sources of variation?

- □ With humans
  - Learning and maturation
  - Motivation and training
  - Process conformance and domain understanding
  - Hawthorn Effect
- □ With artifacts
  - Setup/clean residual effects
  - Perturbations caused by program profiling
  - Non-deterministic behavior

Recurring Issues

How objective can we be?

- □ Comparing a new technique with
  - Current practices is hard without contaminating subjects
  - Other techniques on same test bed can be suspect to “tweaking”
- □ Ideal is not to have a vested interested in techniques we are studying
  - But we are in the best position to identify problems and suggest solutions

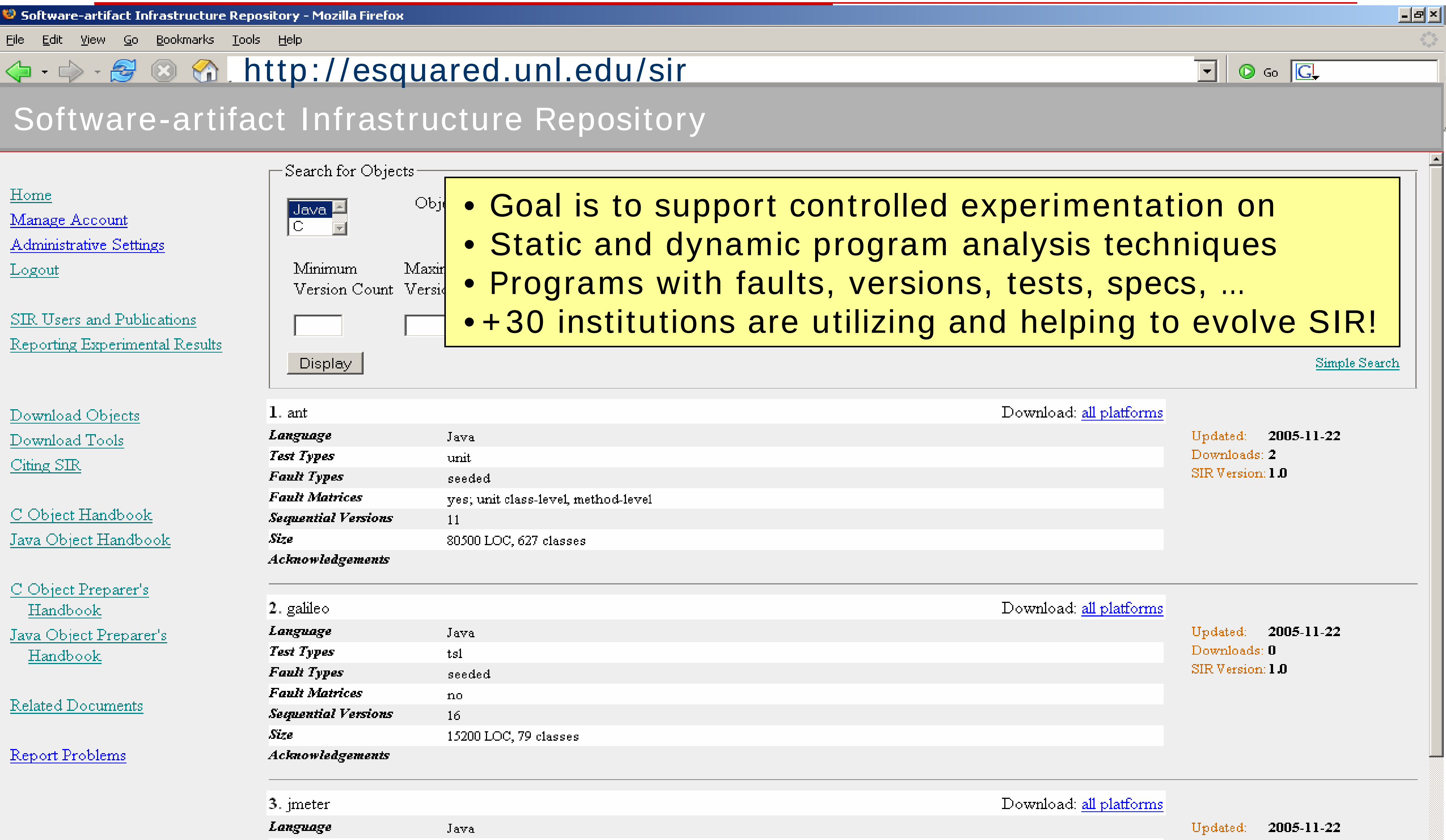
Recurring Issues

How do we support empirical studies?

- Need for infrastructure
  - Test beds are set of artifacts and support for running experiments
  - Testbeds are applicable to limited classes of techniques → need many testbeds
  - Costly but necessary
  - How do we share and evolve infrastructures?

Success Story

Aiding the Empirical Researcher


Software-artifact Infrastructure Repository - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
Go

http://esquared.unl.edu/sir

Software-artifact Infrastructure Repository

- Goal is to support controlled experimentation on
- Static and dynamic program analysis techniques
- Programs with faults, versions, tests, specs, …
- +30 institutions are utilizing and helping to evolve SIR!

Home
Manage Account
Administrative Settings
Logout

SIR Users and Publications
Reporting Experimental Results

Download Objects
Download Tools
Citing SIR

C Object Handbook
Java Object Handbook

C Object Preparer's Handbook
Java Object Preparer's Handbook

Related Documents

Report Problems

Search for Objects

Java
C

Obj

Minimum Version Count

Maxim Versi

Display

Simple Search

1. ant — Download: all platforms

Language	Java
Test Types	unit
Fault Types	seeded
Fault Matrices	yes; unit class-level, method-level
Sequential Versions	11
Size	80500 LOC, 627 classes
Acknowledgements	

Updated: 2005-11-22
Downloads: 2
SIR Version: 1.0

2. galileo — Download: all platforms

Language	Java
Test Types	tsl
Fault Types	seeded
Fault Matrices	no
Sequential Versions	16
Size	15200 LOC, 79 classes
Acknowledgements	

Updated: 2005-11-22
Downloads: 0
SIR Version: 1.0

3. jmeter — Download: all platforms

Language	Java

Updated: 2005-11-22

Success Story:
Aiding the Technique Developer

- □ **Testbed** : TSAFE -a safety critical air traffic control software component

Trying out a technique on a testbed
- helps identify its bounds and limits
- focuses the improvement opportunities
- provides a context for its interaction with other techniques
- helps build the body of knowledge about the class of technique

- □ **Results:** The experimental study resulted in a
  - Better fault classification
  - Identified strengths and weaknesses of the technology
  - Helped improve the design for verification approach
  - Recognized one type of fault that could not be caught

Success Story:
Aiding the Technique User

- **Testbed** : a variety of class projects for high performance computing artifacts at UM, MIT, USC, UCSB, UCSD, MSU
- **Evalu** odels, e.g., t
- **Resul**
- On ce
  - Op
  - UPC/CAF requires around 5-35% less effort than OpenMP
  - XMT-C requires around 50% less effort than MPI.
- For certain kinds of embarrassingly parallel problems, message-passing requires less effort than threaded.
- The type of communication pattern does not have an impact on the difference in effort across programming models.

It is important to build a body of evidence about a domain, based upon experience, recognizing what works and doesn’t work under what conditions

Motivation for this presentation

- □ Discuss the state of the art in empirical studies in software engineering
- □ Debate problems and expectations for papers with empirical components in top SE conference venues

For the Author:
How do we deal with reviews?

- Like with any other review
  - The reviewer is right
  - The reviewer has misunderstood something
    - We led them astray
    - They went astray by themselves
  - The reviewer is wrong

Review example

"It is well-known that shared memory is easier to program than distributed memory (message passing). So well known is this, that numerous attempts exist to overcome the drawbacks of distributed memory."

- Issue: How do you argue that empirical evidence about known ideas is of value?

Review example

"… it is hard to grasp, from the way the results are presented, what is the practical significance of the results. This is mostly due to the fact that the analysis focuses on statistical significance and leaves practical significance aside. Though this, with substantial effort, can partially be retrieved from tables and figures, this burden should not be put on the reader."

- Issue: analysis/results disconnected from practical goals

Review example

"There are two groups in the study with effective sizes of 13 and 14 observations. As the authors point out, the phenomena under study would need samples of more like 40 to 60 subjects given the variance observed. Thus the preferred approach would have been to either treat this study as a pilot, or to obtain data from other like studies to establish the needed sample size for the power needed."

- Issue: How do you present and justify your empirical strategy?

Review example

"… (The technique) was tried on a single form page on five web applications. This is actually quite a limited experiment. Web sites such as those they mention have thousands of pages, and hundreds of those with forms. Perhaps a more extensive study would have produced more interesting results. "

- Issue: how much evidence is enough?
  - Depends on ideas maturity and sub-community empirical expertise

Review example

"the population of inexperienced programmers make it likely that results may be quite different for expert population or more varied tasks"

- Issue: Are empirical studies of students of value?

Review example

"… It is well-known that the composition of the original test suite has a huge impact on the regression test suite. The authors say that they created test cases using the category partition method. Why was only one suite generated for each program? Perhaps it would be better to generate several test suites, and consider the variances."

- Issue: what factors can and should be controlled?
  - We cannot control them all.
  - Tradeoffs: cost, control, representativeness

Review example

- “The basic approach suggested in this paper is very labour intensive. There would appear to be other less labour intensive approaches that were not considered … You have not presented a strong argument to confirm that your approach is really necessary.

- Issue: Have the steps been justified against alternatives?

Review example

"… This paper represents a solid contribution, even though the technique is lightweight … 6 of the 10 submitted pages are about results, analysis of the results, discussion … with only a single page required for the authors to describe their approach. Thus, the technique is straightforward and might be construed as lightweight! ."

- Issue: is there such as thing as too much "study" of a straightforward technique?

From our experience

- Ask questions that matter
  - Why do they matter? To Who? When?
- State tradeoffs and threats
  - Control versus exposure
  - Cost versus representativeness
  - Constructs versus variables
- Solicit/share expertise/resources with
  - Authors (as a reviewer)
  - Readers (as an author)
  - Researchers (as a researcher)
- Maintain chain of significance
  - Conjecture, Impact, Results, Impact, Conjecture

For authors and reviewers

Checklists

One example: “Preliminary Guidelines for Empirical Research in Software Engineering” by B. Kitchenham et al. TSE 02

Relevant to previous reviews

- *Differentiate between statistical significance and practical importance.*
- *Be sure to specify as much of the context as possible.*
- *If the research is exploratory, state clearly and, prior to data analysis, what questions the investigation is intended to address, and how it will address them*.
- *If you cannot avoid evaluating your own work, then make explicit any vested interests (including your sources of support), and report what you have done to minimize bias.*
- *Justify the choice of outcome measures in terms of their relevance to the objectives of the empirical study*.

For the Reviewer

Hints for Reviewing SE Empirical Work - Tichy, EMSE 2000

- □ Don’t expect perfection
- □ Don’t expect a chapter of a statistics book
- □ Don’t expect decisive answers
- □ Don’t reject “obvious” results
- □ Don’t be casual about asking authors to redo their experiment
- □ Don’t dismiss a paper merely for using students as subjects (or small programs)
- □ Don’t reject negative results
- □ Don’t reject repetition of experiments

Advice from our studies: About overall design

- State clearly what questions the investigation is intended to address and how you will address them, especially if the study is exploratory
- Justify your methodology and the particular steps
- Justify your selection of dependent variables
- Try to design your study so you maximize the number of questions asked in that particular study
- Make clear the practical importance of the results independent of the statistical significance
- Specify as much context as possible; it is often hard to do so in a short conference paper
- The more people you can get to review you design, the better, it is easy to miss important points.

Advice from our studies: About scope, sample, representation

- Student studies can show trends that are of real value
- Student studies offer a lot of insights leading to improved questions for professional developers
- It is easy to contaminate subjects in human studies
- It is hard to compare a new technique against the current technique
- Technique definition and process conformance need to be carefully specified in human studies
- Human subject studies are expensive. You cannot easily repeat studies.
- Don’t expect perfection of decisive answers, for example, insights about context variables alone are valuable

Advice from our studies: About building a body of knowledge

- Empirical studies are important even when you believe the results should be self-evident
- It may be difficult to generalize from in vitro to in vivo It is important to make clear the practical importance of the results independent of the statistical significance
- Each study open new questions scaling up is difficult and the empirical methods change
- We need to combine small focused studies to build knowledge, each unit can be a small contribution to the knowledge tapestry
- In the tapestry of studies it is important to integrate negative results; negative results and repeated experiments and important and valuable

Improving the odds of getting a paper accepted at a conference

- Define a complete story (motivation, design, analysis, results, practical relevance)
- Achieve a balance among the
  - Control on the context
  - Generalization of the findings
  - Level of detail in a 10 page paper
- Get as many reviews beforehand as possible

Better Empirical Science for Software Engineering

How not to get your empirical study rejected: we should have followed this advice

Victor Basili

University of Maryland 1856

Fraunhofer USA, Inc
Center for Experimental Software Engineering Maryland

Sebastian Elbaum

University of Nebraska Lincoln

References

- V. Basili, "Evolving and Packaging Reading Technologies“, Journal of Systems and Software, 38 (1): 3-12, July 1997.
- V. Basili, F. Shull, and F. Lanubile, "Building Knowledge through Families of Experiments“, IEEE Transactions on Software Engineering, 25(4): 456-473, July 1999.
- S. Elbaum, A. Malishevsky, and G. Rothermel, "Test Case Prioritization: A Family of Empirical Studies", IEEE Transactions on Software Engineering, 28-2:159-182, 2002.
- H. Do, S. Elbaum, and G. Rothermel, “Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact”, Empirical Software Engineering: An International Journal, 10(4):405-435, 2005.
- G. Rothermel, S. Elbaum, A. Malishevsky, P. Kallakuri and X. Qiu, "On Test Suite Composition and Cost-Effective Regression Testing", ACM Transactions of Software Engineering and Methodologies, 13(3):277-331, July 2004.
- B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin, K. Emam and J. Rosenberg, “Preliminary Guidelines for Empirical Research in Software Engineering”, IEEE Transactions on Software Engineering, 28(8):721--734, 2002.
- R. Selby, V. Basili, and T. Baker, “Cleanroom Software Development: An Empirical Evaluation,” IEEE Transactions on Software Engineering, 13(9): 1027-1037, September 1987.
- W. Tichy, “Hints for Reviewing Empirical Work in Software Engineering”, Empirical Software Engineering: An International Journal 5(4): 309-312, December 2000.