Writing Good Software Engineering Research Papers

Mary Shaw

Carnegie Mellon University

www.cs.cmu.edu/~shaw/

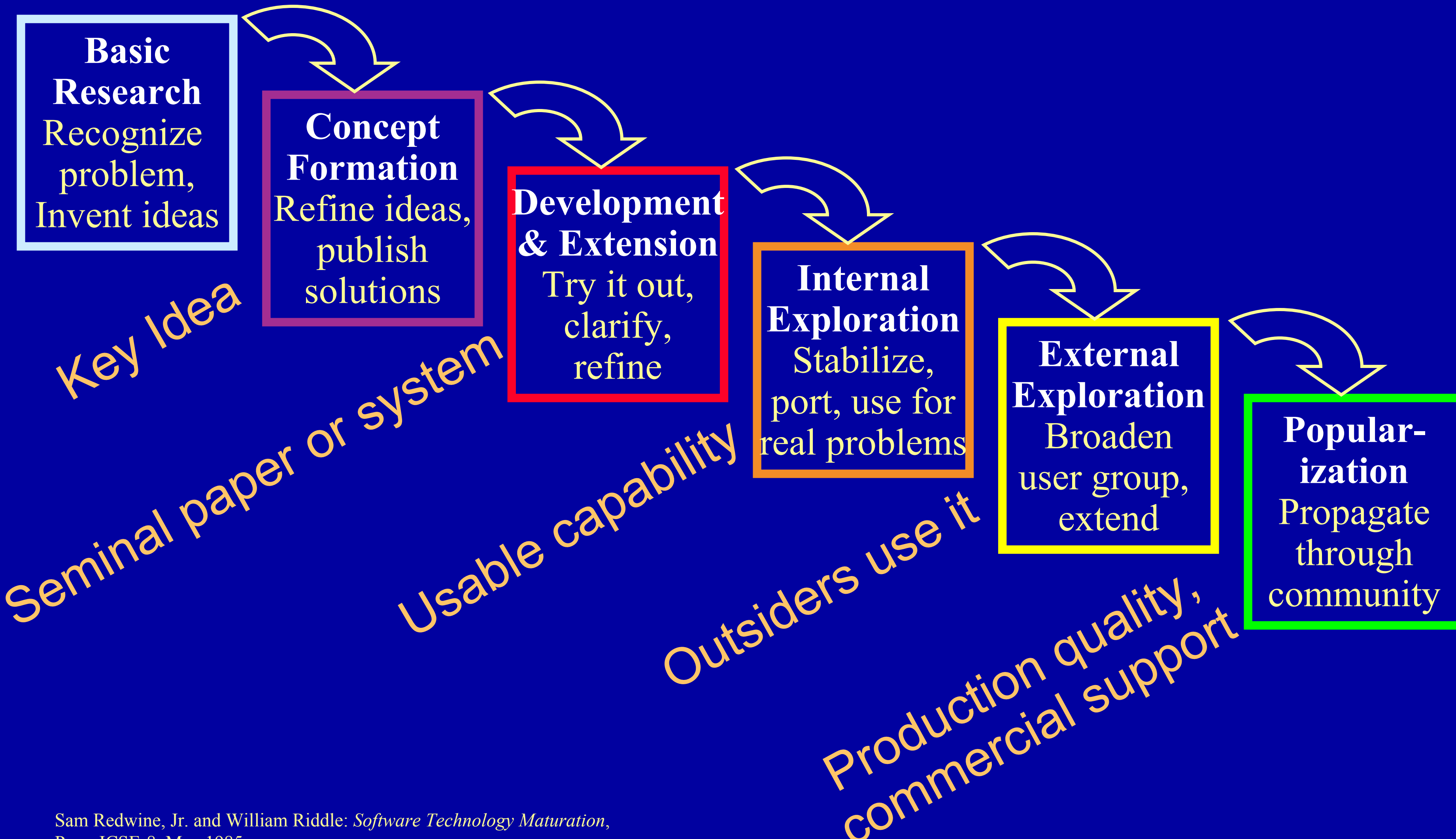
Good Writing Needs Good Content

- Writing a good paper depends on having good research to write about
  - > If the result is not significant, it doesn't matter how good the paper is
  - > If your claims don't match your results, you'll have trouble providing convincing evidence
- It's also hard work, a skill that requires practice. Writing a paper is like designing a system.
- So this minitutorial addresses both your research strategy and how you present the work

Plan

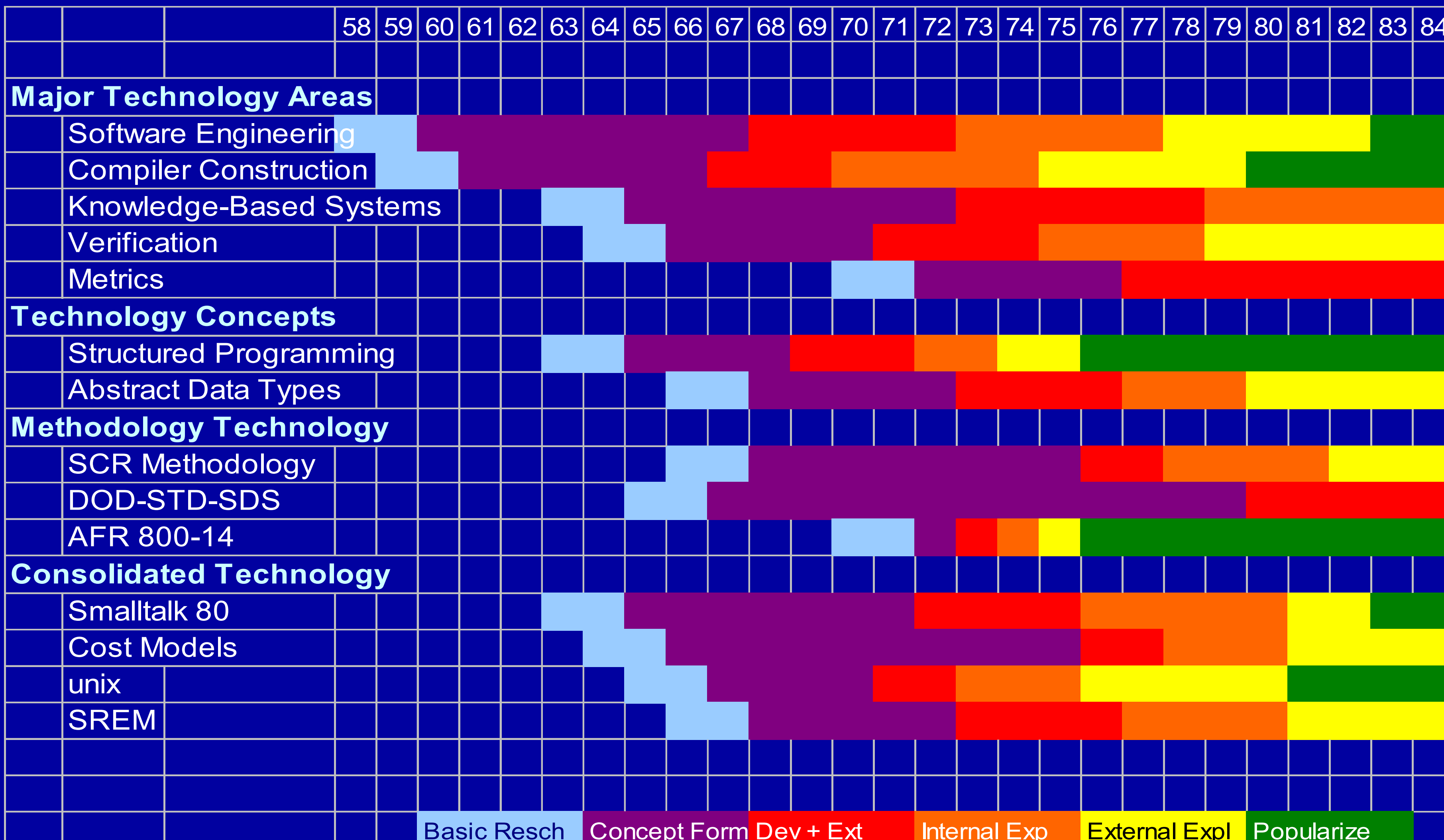
- Life cycle of a technological innovation
  - > Different issues, venues at different stages
- Focus on research papers
  - > Various authors, conference advice
- Elements of a research presentation
  - > Question, result, validation
  - > Data from ICSE 2002, 2003
- Research strategies that work
  - > The logical structure of a project and paper
  - > Examples from ICSE 2003

Redwine/Riddle Maturation Model

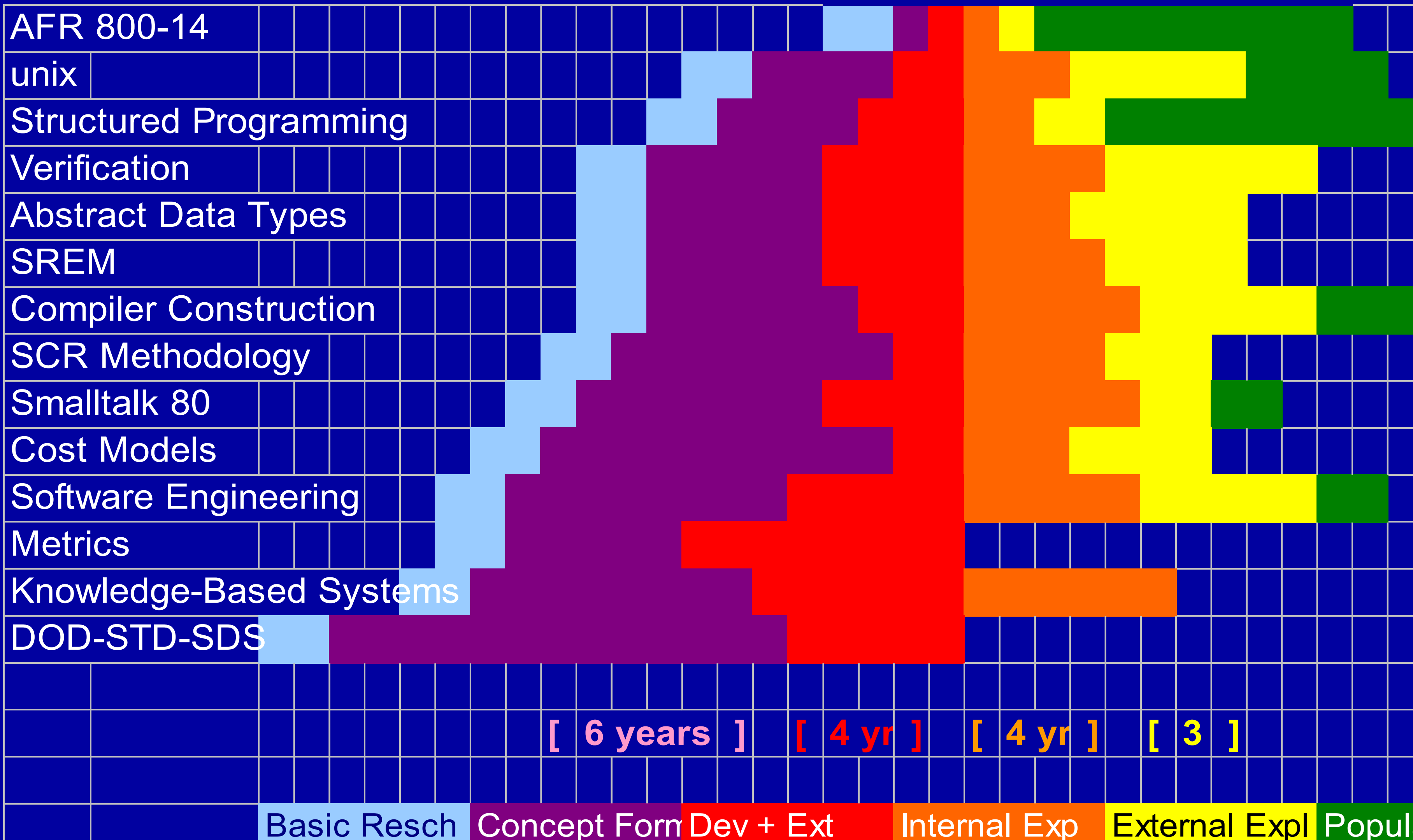


Sam Redwine, Jr. and William Riddle: *Software Technology Maturation*, Proc ICSE-8, May 1985

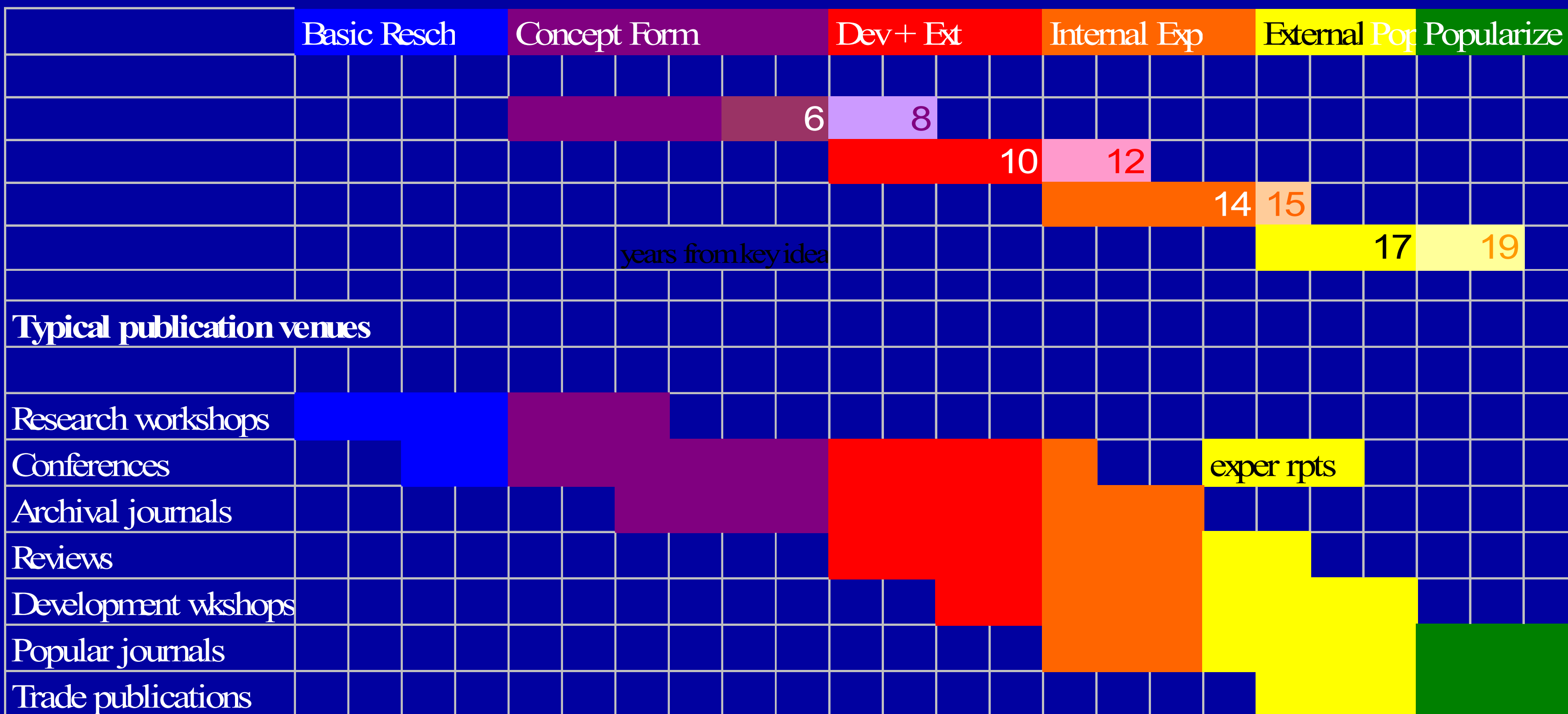
Software Technology Maturation Points



Maturation Times



Phase Times and Publications



Success needs cumulative evidence

- A single paper has limited scope
  - > Conference papers can hold one idea
  - > Journal papers can wrap up individual results
- Results are more convincing if they are confirmed in different ways (triangulation)
- Each promising step justifies investment in next (often more expensive) step

Plan

- Life cycle of a technological innovation
  - > Different issues, venues at different stages
- Focus on research papers
  - > Various authors, conference advice
- Elements of a research presentation
  - > Question, result, validation
  - > Data from ICSE 2002, 2003
- Research strategies that work
  - > The logical structure of a project and paper
  - > Examples from ICSE 2003

Research Styles

- Physics and medicine have well-recognized research styles
  - Hypothesis, controlled experiment, analysis, refutation
  - Double-blind large-scale studies
- Acceptance of results relies on process as well as analysis
- Simplified versions help to explain the field to observers
- Fields can be characterized by identifying what they value:
  - What kinds of questions are "interesting"?
  - What kinds of results help to answer these questions?
    - What research methods can produce these results?
  - What kind of evidence demonstrates the validity of a result?

Critiques of Experimental CS/SE

"Computer scientists publish relatively few papers with experimentally validated results … The low ratio of validated results appears to be a serious weakness in CS research. This weakness should be rectified"

- Studies over past few years criticize computer science for failure to collect, report, analyze experimental data
- They start with the premise that data *must* be collected, then analyze papers and find data lacking
- I ask a different question:
  What are the characteristics of software engineering research that the field recognizes as quality research?

W. F. Tichy &al. "Experimental evaluation in computer science: A quantitative study." *Journal of Systems Software,* Vol. 28, No. 1, 1995, pp. 9-18.
Walter F. Tichy. "Should computer scientists experiment more? 16 reasons to avoid experimentation." *IEEE Computer*, Vol. 31, No. 5, May 1998.
M. Zelkowitz & D. Wallace. "Experimental models for validating technology." Computer (IEEE), Vol. 31, No. 5, 1998, pp.23-31.

Newman: Pro Forma Abstracts

- Asked, “To what extent is HCI an engineering discipline”?
- Characterized engineering research products
- Created three pro forma abstracts, templates describing research
- 90% of papers in engineering research fit these templates

William Newman. "A preliminary analysis of the products of HCI research, using pro forma abstracts." Proceedings of the CHI '84, pp.278-284

Newman’s Pro Forma Templates for Engineering

EM: Enhanced model

Existing model-type models are deficient in dealing with properties of solution strategy. An enhanced model-type is described, capable of providing more accurate analyses / predictions of properties in solution strategy designs. The model has been tested by comparing analyses / predictions with empirically measured values of properties.

ES: Enhanced solution

Studies of existing artifact-type have shown deficiencies on property. An enhanced design for an artifact-type is described, based on solution strategy. In comparison with existing solutions, it offers enhanced levels of property, according to analyses based on model-type. These improvements have been confirmed / demonstrated in tests of a working artifact-type based on the design.

ET: Enhanced tool

The effectiveness of model-type / solution strategy in supporting the design of artifact-type has been demonstrated. An enhanced tool / method is described for the design of artifact-type based on model-type / solution strategy. Examples are provided confirming the effectiveness of its support for model-type / solution strategy in design.

Newman: Pro Forma Abstracts

- Only 25-30% of HCI papers fit
- Created 2 more pro forma abstracts (arguably engineering)
- Now 95% of HCI papers fit
- Notes
  - > Preliminary study, e.g., no check on inter-rater reliability
  - > Found this a useful device for reading papers
  - > Influenced refereeing in CHI

William Newman. "A preliminary analysis of the products of HCI research, using pro forma abstracts." Proceedings of the CHI '84, pp.278-284.

Additional Pro Forma Templates for HCI

RS: Radical solution

A radical solution to the problem of problem definition is described, based on solution strategy. In comparison with existing normal solutions it offers advantages, which have been demonstrated in preliminary tests, but it leaves a number of side effects to be addressed including list of side effects. Strategies are suggested for addressing these side effects.

XH: Experience and/or Heuristic

Studies reported here of application supported by supporting technology generate a number of findings concerning issues, including list-of-findings. They indicate that requirement is / is not met by design-heuristic.

Brooks: Kinds of Research Results

Brooks proposed recognizing three kinds of results, with individual criteria for quality:

> findings -- well-established scientific truths -- judged by truthfulness and rigor

> observations -- reports on actual phenomena -- judged by interestingness

> rules-of-thumb -- generalizations, signed by an author (but perhaps not fully supported by data) -- judged by usefulness

with freshness as criterion for all

Frederick P. Brooks, Jr. Grasping Reality Through Illusion -- Interactive Graphics Serving Science. Proc ACM SIGCHI Human Factors in Computer Systems Conference, May 1988, pp. 1-11.

Conference-specific advice

- There's lots of "how to write a paper" advice
  - > OOPSLA, POPL, PLDI, SOSP, SIGCOMM, SIGGRAPH
  - > Links on my writing advice web site
    - » www.cs.cmu.edu/~shaw > Education > WordWright
    - » Under Resources > CS Advice
- HCI community does better
  - > Newman analysis above
  - > Analysis of regional differences in acceptance rates

Plan

- Life cycle of a technological innovation
  - > Different issues, venues at different stages
- Focus on research papers
  - > Various authors, conference advice
- Elements of a research presentation
  - > Question, result, validation
  - > Data from ICSE 2002, 2003
- Research strategies that work
  - > The logical structure of a project and paper
  - > Examples from ICSE 2003

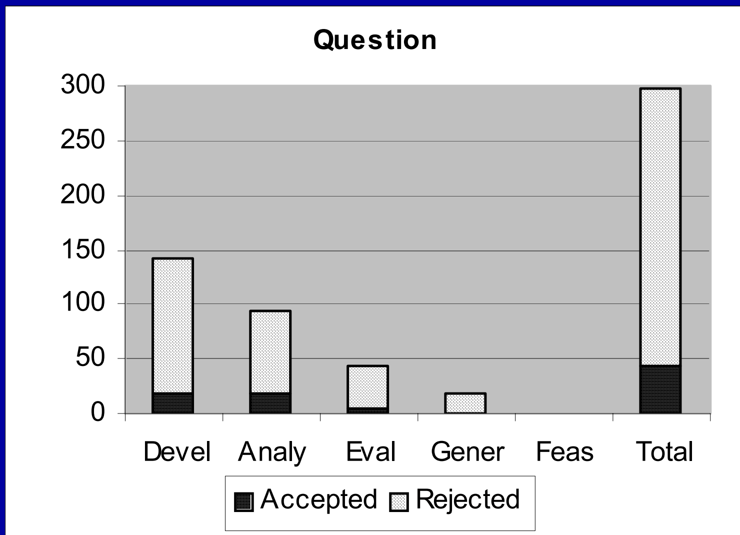
Research Objectives

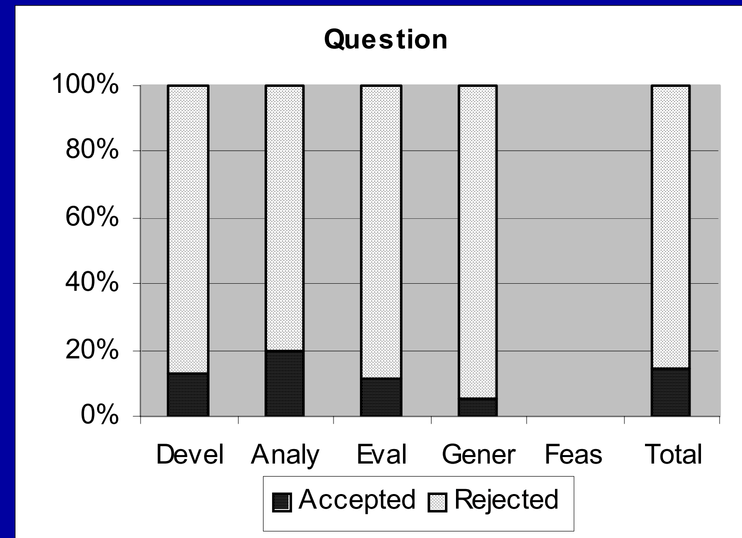
- Key objectives
  - > Quality -- utility as well as functional correctness
  - > Cost -- both of development and of use
  - > Timeliness -- good-enough result, when it’s needed
- Address problems that affect practical software

Types of Research Questions

Method/means of development	How can we do/create/automate X ? What is a better way to do/create X ?
Method for analysis	How can I evaluate the quality of X ? How do I choose between X and Y ?
Evaluation / analysis of an instance	What is property X of artifact/method Y ? How does X compare to Y ? What is the current state of X / practice of Y ?
Generalization / characterization	Is X always true of Y ? Given X, what is Y ? What, exactly, do we mean by X ? Is Y a good formal/empirical model for X ? What are the types of X, how are they related ?
Feasibility	Does X exist, and what is it ? Is it possible to do X at all ?

ICSE 2002 submissions





Type of question	Submitted	Accepted	2003	Ratio Acc/Sub
Method or means of development	142(48%)	18(42%)	13	(13%)
Method for analysis or evaluation	95(32%)	19(44%)	18	(20%)
Design, evaluation, or analysis of a particular instance	43(14%)	5 (12%)	4	(12%)
Generalization or characterization	18(6%)	1 (2%)	7	(6%)
Feasibility study or exploration	0 (0%)	0 (0 %)	0	(0%)
TOTAL	298(100.0%)	43(100.0%)	42	(14%)

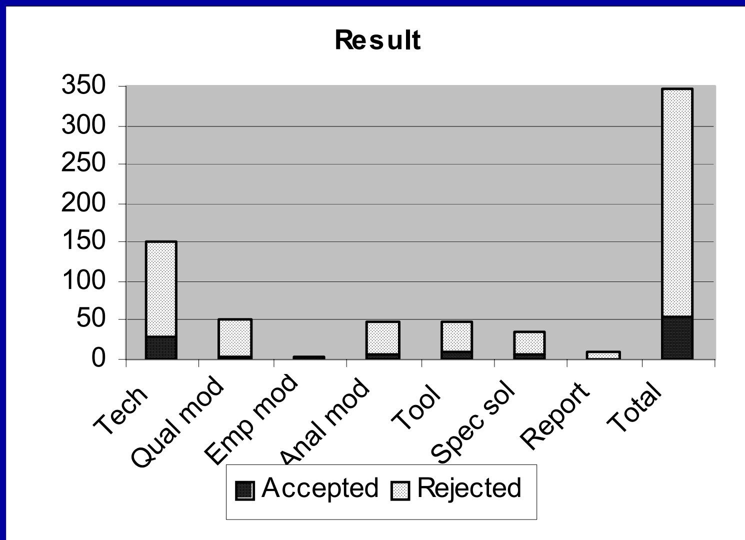
What do PCs look for?

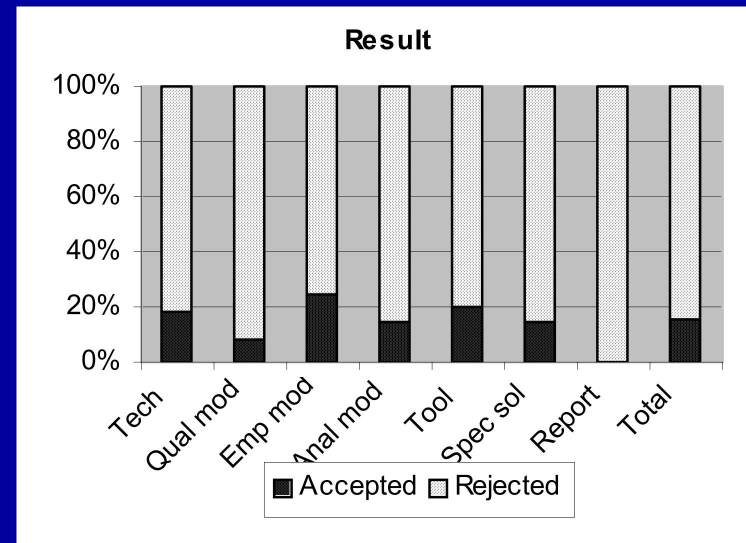
- Clear statement of the question you answered
  - > that is, the problem about software you answered
- Explanation of why the problem matters

Types of Research Results

Procedure / technique	New/better ways to do development/analysis tasks; (operational, not just guidelines)
Qualitative or descr. model	Structure/taxonomy for problem area; framework Informal guidance, informal domain analysis
Analytic model	Structural model that permits formal analysis, automation
Empirical model	Empirical predictive models based on real data
Tool / notation	Tool or notation that embodies model or technique
Specific solution	Solution to application problem applying SE principles, or result of specific analysis
Report	Interesting observations, rules of thumb

ICSE 2002 submissions





Type of result	Submitted	Accepted	2003	Ratio Acc/Sub
Procedure or technique	152(44%)	28(51%)	18	18%
Qualitative or descriptive model	50(14%)	4 (7%)	7	8%
Empirical model	4 (1%)	1 (2%)	5	25%
Analytic model	48(14%)	7 (13%)	11	15%
Tool or notation	49(14%)	10(18%)	5	20%
Specific solution, prototype, answer, or judgment	34(10%)	5 (9%)	2	15%
Report	11(3%)	0 (0%)	1	0%
TOTAL	348(100.0%)	55(100.0%)	49	16%

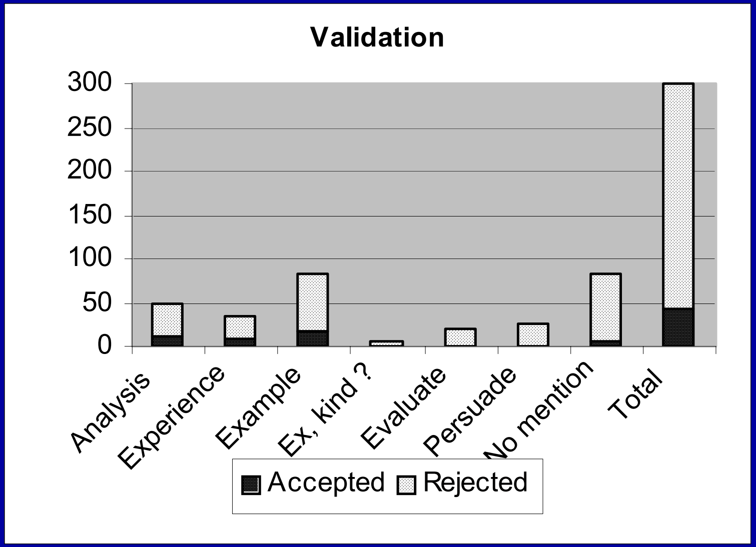
What do PCs look for?

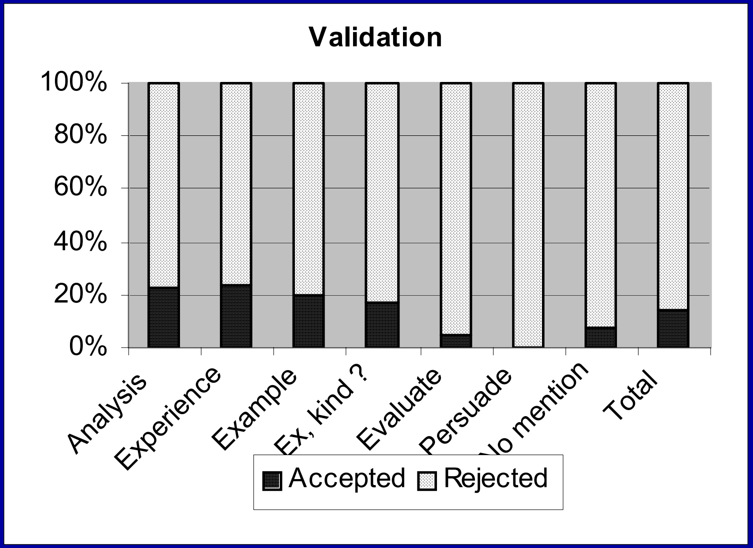
- What's new? How is it related to prior work?
- What, precisely, does the research claim to show?
  - > If it should work on large systems, show it scales
  - > If it's "automatic", don't use manual intervention
  - > If it's "distributed", don't assume central server
  - > If it's a new notation, show why it's better
  - > If it's a new model, be clear about its power
  - > If it's a new design element, treat it as a generalization
  - > If it's a synthesis, say why the synthesis is novel
  - > If an implementation is featured, show its role

Types of Research Validation

Analysis	I have found my result satisfactory through ...
Formal model	rigorous derivation and proof
Empirical model	data on use in controlled situation
Controlled experiment	carefully designed statistical experiment
Experience	My result has actually been used; the evidence is
Qualitative model	narrative
Empirical model, tool	data, usually statistical, on practice
Notation, technique	comparison of systems in actual use
Example	Here's how my result works on a small example
Evaluation	Given these criteria, my result ...
Descriptive model	adequately describes phenomena of interest
Empirical model	is able to predict … because …
Persuasion	I thought hard about this, and I believe...
Blatant assertion	No serious attempt to evaluate result

ICSE 2002 submissions





Type of validation	Submitted	Accepted	2003	Ratio Acc/Sub
Analysis	48(16%)	11(26%)	11	23%
Evaluation	21(7%)	1 (2%)	7	5%
Experience	34(11%)	8 (19%)	7	24%
Example	82(27%)	16(37%)	17	20%
Some example, can't tell whether it's toy or actual use	6 (2%)	1 (2%)	0	17%
Persuasion	25(8%)	0 (0.0%)	0	0%
No mention of validation in abstract	84(28%)	6 (14%)	-	7%
TOTAL	300(100.0%)	43(100.0%)	42	14%

What do PCs look for?

- Solid evidence: why the reader should believe result
- Validation related to the claim
  - > If you improve on prior art, do comparison
  - > If you did analysis, follow its rules
  - > If you cite practical experience, separate your effect
- Accurate description of the evidence
  - > “case study” & “experiment” >> data & anecdotes

Commonest Types of ICSE 2002 Papers

- Question
  - Most common: improved method or means of developing software
  - Also fairly common: papers about methods for analysis, principally analysis of correctness (most common in 2003)
- Result
  - Most common: a new procedure or technique for some aspect of software development
  - Not unusual: a new analytic model
- Validation
  - Most common: analysis and experience in practice
  - Also fairly common: example idealized from practice
  - Common in submissions but not acceptances: persuasion

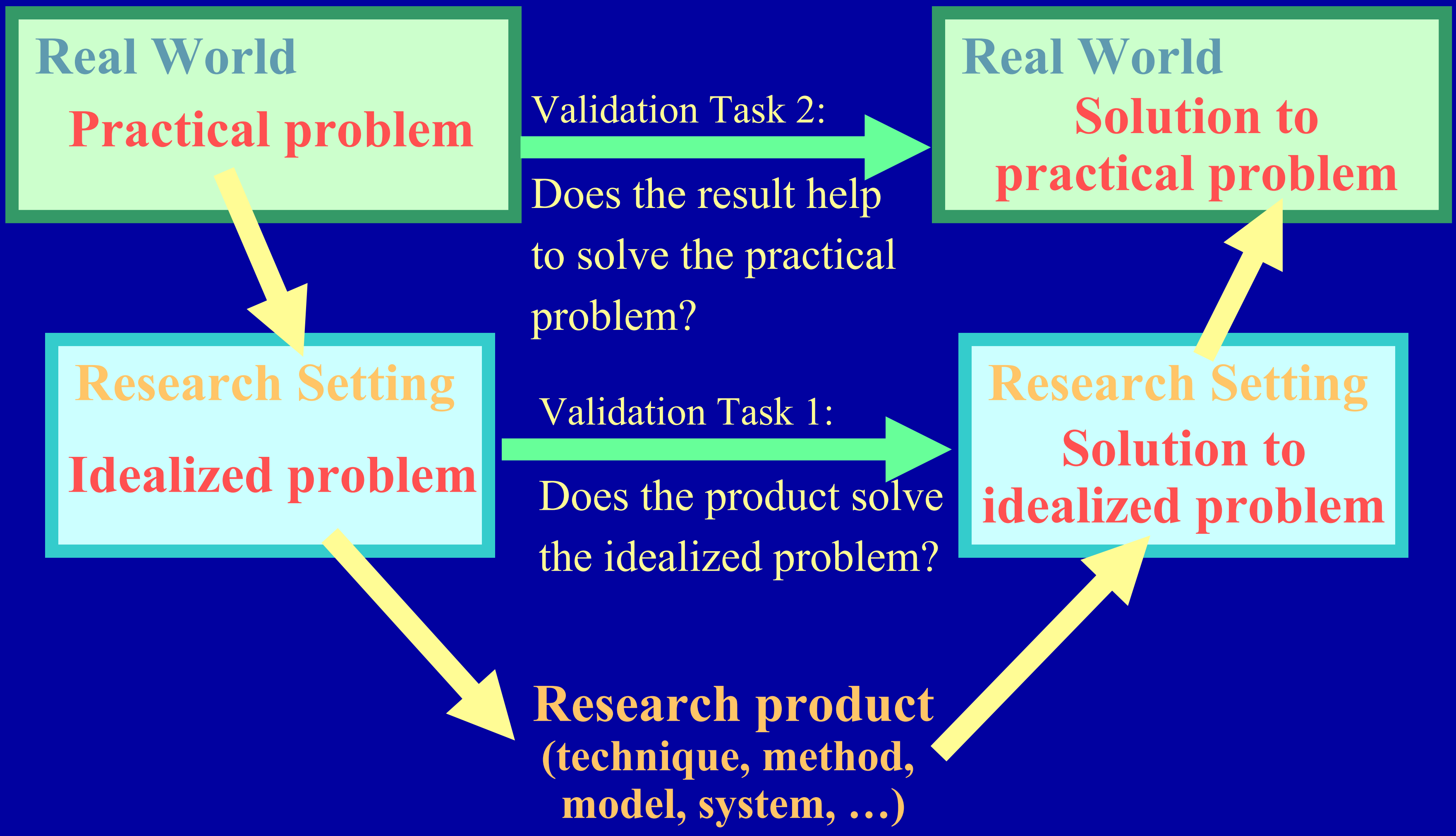
Building Blocks for Research

Question	Strategy/Result	Validation
Devlpmt method	Proc/technique	Analysis
Analysis method	Qual/desc model	Experience
Evaluate instance	Analytic model	Example
Generalization	Empirical model	Evaluation
Feasibility	Tool/notation	Persuasion
	Specific solution	
	Report	

Plan

- Life cycle of a technological innovation
  > Different issues, venues at different stages
- Focus on research papers
  > Various authors, conference advice
- Elements of a research presentation
  > Question, result, validation
  > Data from ICSE 2002, 2003
- Research strategies that work
  > The logical structure of a project and paper
  > Examples from ICSE 2003

Complete Research Result



Two Common Plans

Question	Strategy/Result	Validation
Can X be better?	*New method*	Analysis
Can X tell you Y?	Qual/desc model	*Report actual use*
Evaluate instance	Analytic model	*Careful examples*
Generalization	Empirical model	Evaluation
Feasibility	Tool/notation	Persuasion
	Specific solution	
	Report	

Sagar Chaki, et al. *Modular Verification of Software Components in C*. Proc ICSE 2003 p.385. ICSE 2003 Distinguished Paper

Question (Analysis method): How can we automatically verify that a finite state machine specification is a safe abstraction of a C procedure?

Result (Technique, supported by tool):

Extract finite model from C source code (using predicate abstraction and theorem proving); show conformance via weak simulation.

Decompose verification to match software design so results compose.

Tool interfaces with public theorem provers

Validation (Examples):

Use examples whose correct outcome is known

Compare performance with various public provers incorporated

Verify OpenSSL handshake

Two Common Plans

Question	Strategy/Result	Validation
Can X be better?	*New method*	Analysis
Can X tell you Y?	Qual/desc model	*Report actual use*
Evaluate instance	Analytic model	*Careful examples*
Generalization	Empirical model	Evaluation
Feasibility	Tool/notation	Persuasion
	Specific solution	
	Report	

Roope Kylmäkoski. *Efficient Authoring of Software Documentation Using RaPiD7*. Proc ICSE 2003 p.255.

Question (Development method): How can we improve on the traditional approach to document authoring?

Result (Technique):

Document authored by team in series of workshops

Workshops are highly structured around concrete issues

Validation (Experience):

In use in Nokia since 2000

Self-assessment by survey in 2001, good results

reduces calendar time for document

improves communication

reduces defects

Empirical Validation

Question

- Devlpmt method
- *Can we predict cost?*
- Evaluate instance
- Generalization
- Feasibility

Strategy/Result

- *Cost est method*
- Qual/desc model
- Analytic model
- Empirical model
- Tool/notation
- Specific solution
- Report

Validation

- *Statistical comparison*
- Experience
- Example
- Evaluation
- Persuasion

M Ruhe, R Jeffery, I Wieczorek. *Cost Estimation for Web Applications*. Proc ICSE 2003 p.285.

Question (Anaysis method): Can we estimate costs of developing web applications?

Result (Technique):

Tailor existing COBRA method for web applications

Get data set from web development company

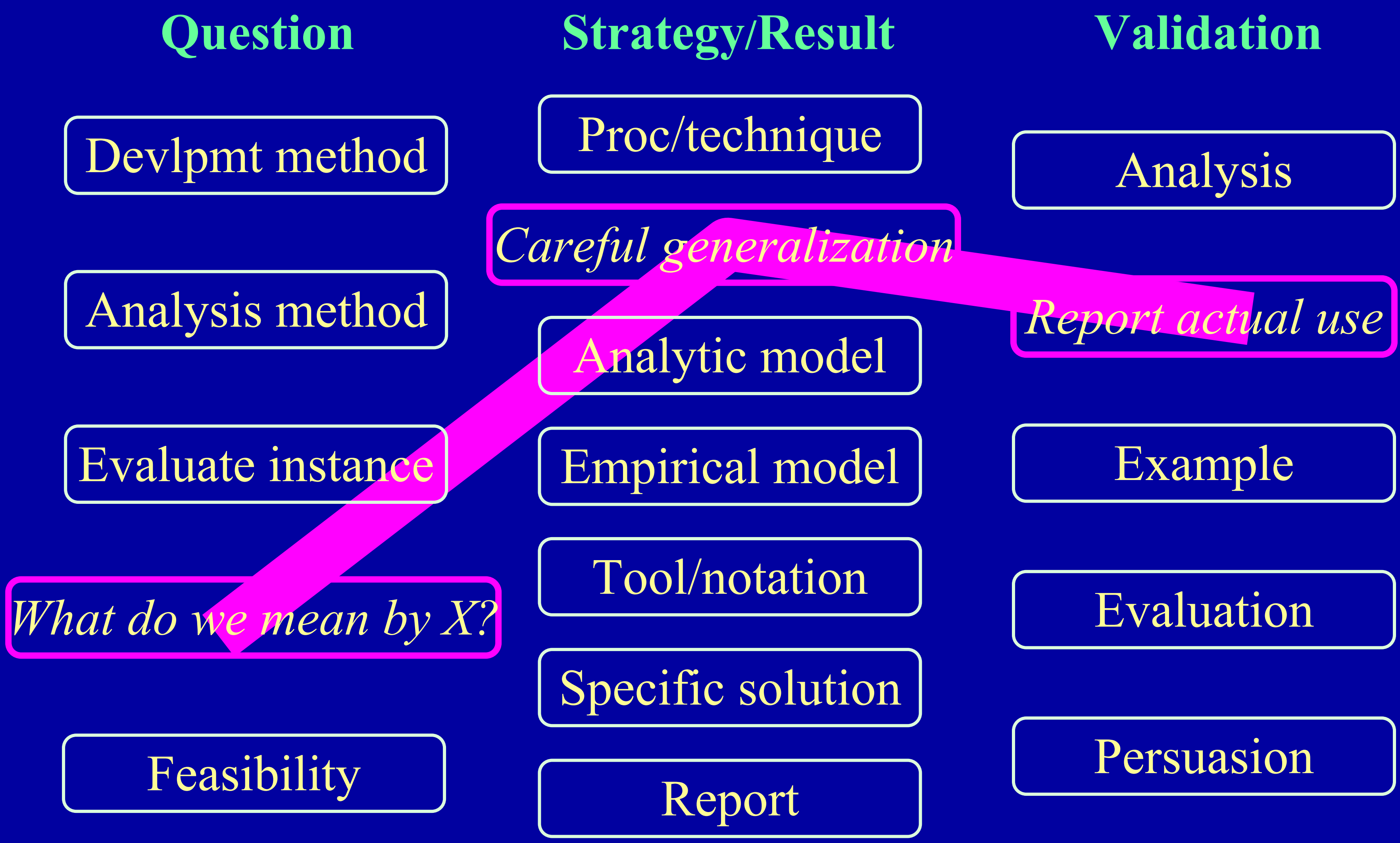
Validation (Analysis, statistically valid):

Establish evaluation criteria through interviews

Apply tailored COBRA, least squares, and company's informal model

Compare results in several ways, including t-tests

A Generalization Paper



S. Sim, E. Easterbrook, R. Holt. *Using Benchmarking to Advance Research: A Challenge to Software Engineering*. Proc ICSE 2003 p.74.

Question (Generalization): What are benchmarks, in general, and how could using them improve software engineering research?

Result (Qualitative model):

Examine three successful benchmarks

Formulate descriptive theory

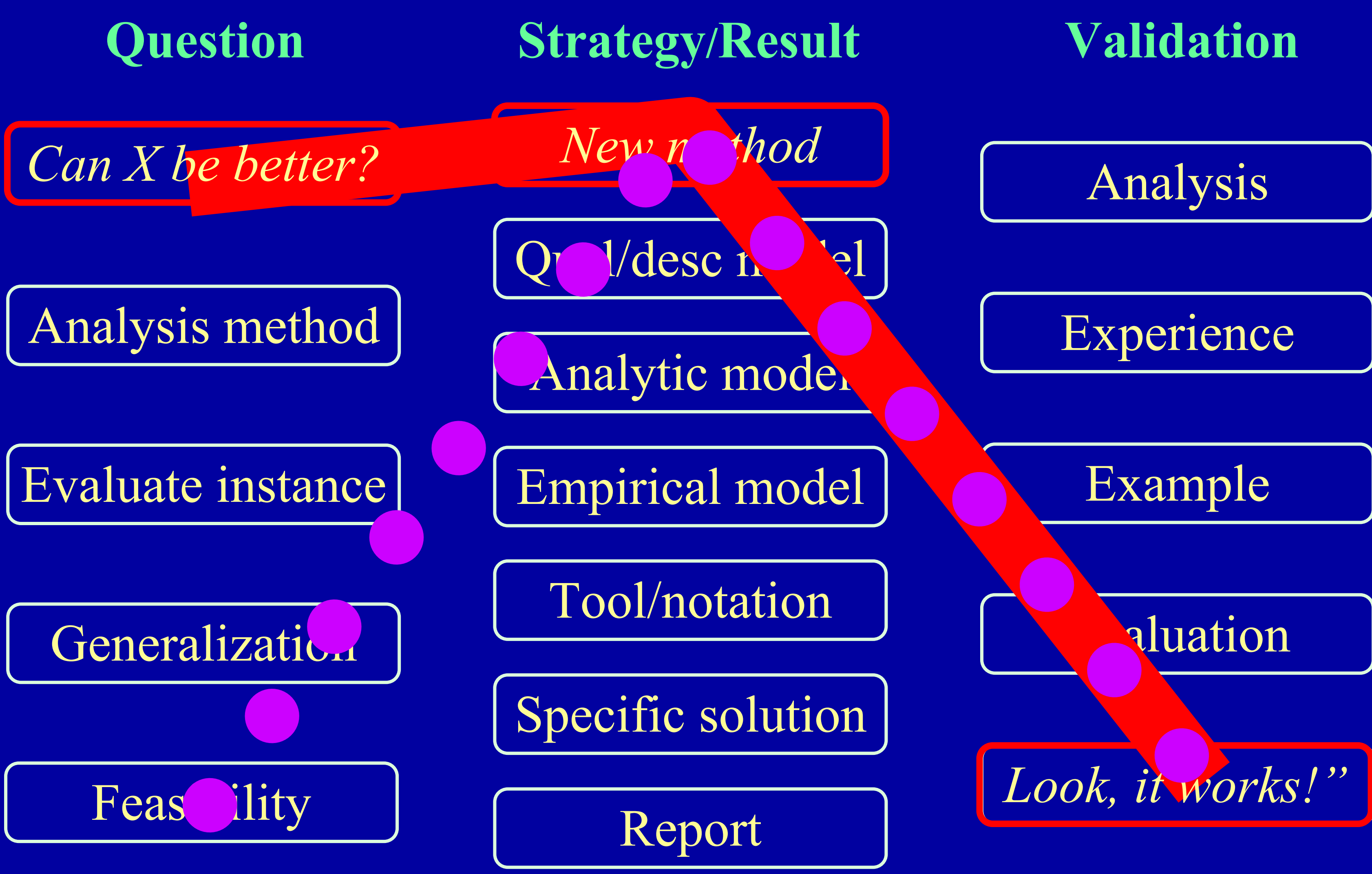
Describe how theory should inform practice

Validation (Experience):

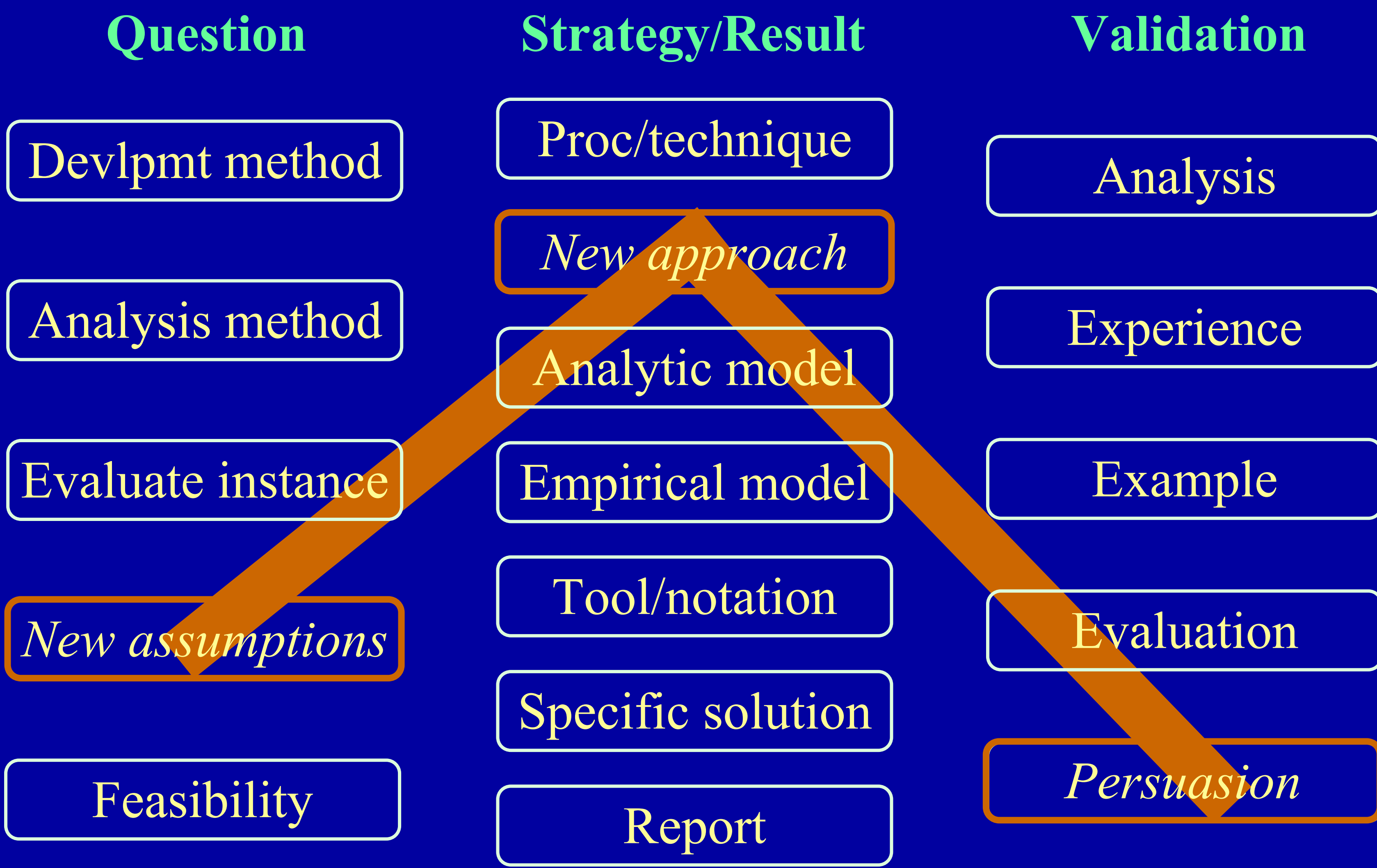
Apply theory to interpret two reverse engineering benchmarks

Identify three areas that are ripe for benchmarking

A Common, but Bad Plan
An Uncommon, but Good, Plan



Sometimes a breakthrough
(but sometimes nonsense)



ICSE 2002 and 03 Paper Types

	Devel Meth	Anal-Meth	Inst-ance	Gener-aliz'n	Feas-ibility
Proc, Tech	❷❷❷❷❷ ❷❷❷%% %%%%	❷❷❷❷❷ ❷❷❷❷❷ ❷❷❷❷% %%%%%% %%%%%%			
Qual Model	❷❷%%%		%	%%	
Emp Model				%%%%%	
Anal Model	❷❷%%%	❷❷❷%% %%%%%%		%	
Nota-tion	❷%	❷			
Spec Soln			❷❷❷❷ ❷%%		
Report			%		

	Anal-ysis	Exper-ience	Exam-ple	Eval-uation	Persu-asoion
Proc, Tech	❷❷❷❷ ❷❷❷% %%%%%	❷❷❷❷ ❷%%%	❷❷❷❷ ❷❷❷❷ ❷%%% %%%%	❷%%	
Qual Model		❷❷%%%%	%%	%	
Emp Model	%%			%%	
Anal Model	%%	❷❷❷	❷❷%% %%%% %	%%	
Nota-tion	❷	❷%			
Spec Soln	❷❷❷%%		❷❷		
Report			%		

Newman's "Enhanced Model"

EM: Enhanced model

Existing model-type models are deficient in dealing with properties of solution strategy. An enhanced model-type is described, capable of providing more accurate analyses / predictions of properties in solution strategy designs. The model has been tested by comparing analyses / predictions with empirically measured values of properties.

Key: EM provides new or better way of looking at problems

Question

Generalization / characterization: What, exactly do we mean by X? What is a good formal/empirical model of X?

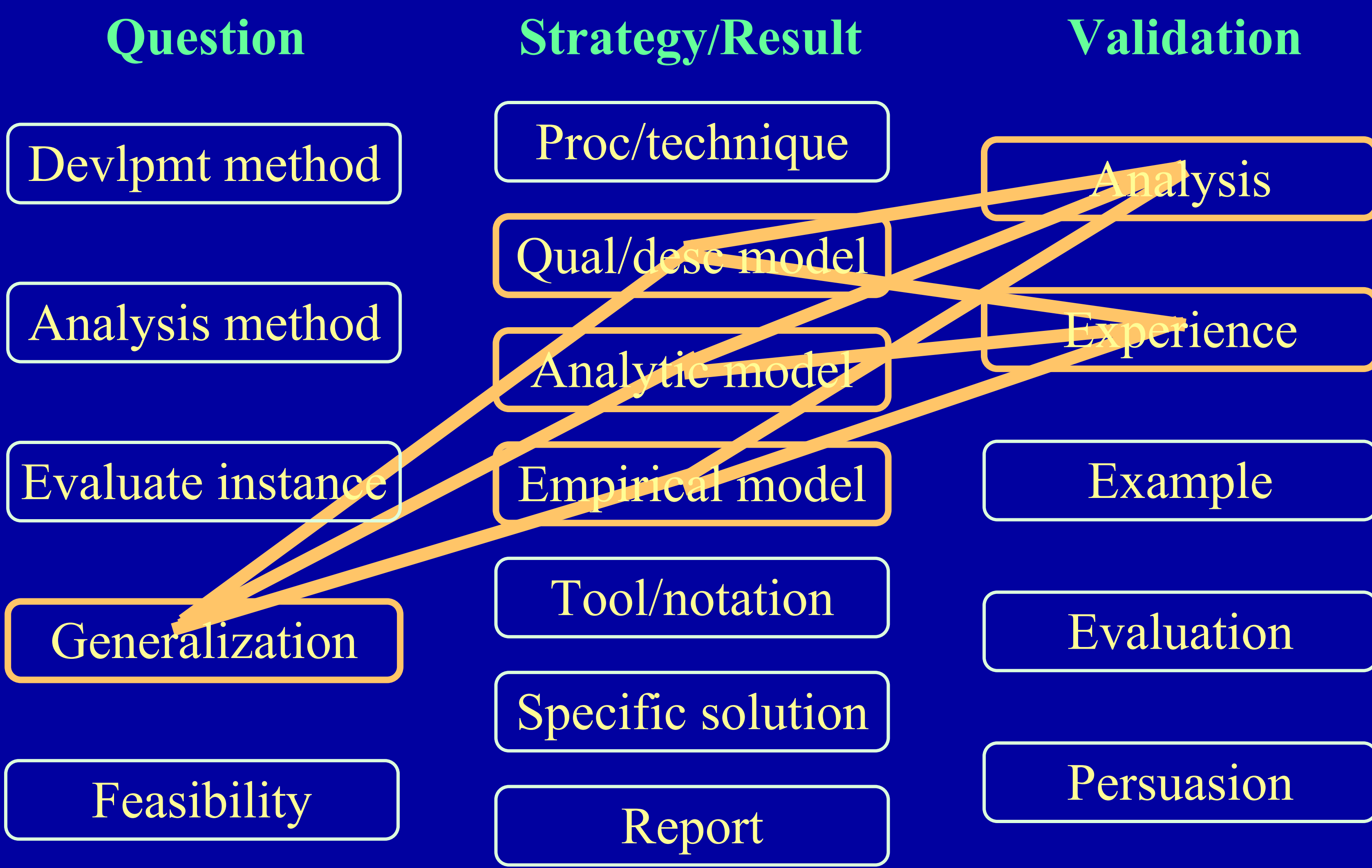
Result

Models, preferably analytic or empirical, but *precise* descriptive or qualitative are acceptable

Validation

Empirical analysis, controlled experiment; perhaps experience

Newman’s “Enhanced Model”



Pro Forma Research Strategies

Locating the *pro forma* abstracts in research strategy space

	Devel Meth	Anal- Meth	Inst- ance	Gener- aliz'n	Feas- ibility
Proc, Tech	**ET**	**ET**			
Qual Model				**EM**	
Emp Model				**EM**	
Anal Model				**EM**	
Nota- tion					
Spec Soln			**ES** **RS**		**RS**
Report			**ES, RS** **XH**	**XH**	**RS**

	Anal- ysis	Exper- ience	Exam- ple	Eval- uation	Persu- asoion
Proc, Tech			**ET** **ET**		
Qual Model	**EM**	**EM**			
Emp Model	**EM**	**EM**			
Anal Model	**EM**	**EM**			
Nota- tion					
Spec Soln		**ES**		**RS** **RS**	**RS** **RS**
Report		**ES**		**RS RS** **XHXH**	**RS RS** **XHXH**

Putting the Words on Paper

- A research paper is a purposeful, designed artifact
  - > Just like a software system
- Apply software design techniques to paper design
  - > Start with the requirement: read the call for papers
  - > Select an architecture: plan the sections, what they say
  - > Plan a schedule: allow time for review, revision
  - > Check consistency: type-check text like code
- See writing guidance at
  - > www.cs.cmu.edu/~shaw > Education > WordWright

Good Research in Software Engineering

Examine the kinds of research questions software engineers ask and the ways they study those questions

- Research questions are of different kinds

  Kinds of interesting questions change as ideas mature

- Research strategies also vary

  They should be selected to match the research questions

- Ideas mature over time

  They grow from qualitative and empirical understanding to precise and quantitative models

- Good papers are steps toward good results

  Each paper provides some evidence, but overall validation arises from accumulated evidence

Final word – about this report

- In Brooks' sense, a rule of thumb or generalization
- Not a technical result (a finding) …
  - > No attempt to show anyone else can apply the model
  - > No principled analysis
  - > Limited data
    - » one full set of abstracts and observation of PC
    - » one set accepted papers as published
  - > Use of abstracts as proxies for full papers is suspect
    - » Though accepted 2003 papers suggest they're not bad
  - > Little discussion of related work