UFMG

lab-soft
software engineering laboratory

Writing Good Software Engineering Research Papers

Minitutorial

Mary Shaw – Carnegie Mellon University

ICSE 2003

Empirical Software Engineering

Presenter: Cleiton Silva Tavares

October 21, 2019

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

Introduction

- Research papers are customary vehicles for reporting results to the research community
  - What he/she accomplished
  - How he/she accomplished it
  - Why the reader should care

- A good research paper should answer a number of question

Introduction

- What, precisely, was your contribution?
  - What question did you answer?
  - Why should the reader care?
  - What larger question does this address?

Introduction

- What is your new result?
  - What new knowledge have you contributed that the reader can use elsewhere?
  - What previous work do you build on? What do you provide a superior alternative to?
  - How is your result different from and better than this prior work?
  - What, precisely and in detail, is your new result?

Introduction

- Why should the reader believe your results?
  - Whats standard should be used to evaluate your claim?
  - What concrete evidence show that your result satisfies your claim?

Goal

- Examine how software engineerings answer the questions above, with emphasis on the design of the research project and the organization of the report (ICSE 2002)

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

What, precisely, was your contribution?

- Before reporting what you did, explain what problem you set out to solve of what question you set out to answer – and why this is important

What, precisely, was your contribution?

- What kinds of questions do software engineerings investigates?
  - Better Way to Develop Software
  - Better Way to Evaluate Software

What, precisely, was your contribution?

- Development
  - Creating and modifying the software
  - Design documents

- Evaluation
  - Predicting, determining properties of the software
  - Performance

What, precisely, was your contribution?

- Answers questions about:
  - Methods of development or analysis
  - Details of designing or evaluating a instance
  - Generalizations over whole classes of systems

What, precisely, was your contribution?

Table 1. Types of software engineering research questions	
Type of question	**Examples**
Method or means of development	How can we do/create/modify/evolve (or automate doing) X? What is a better way to do/create/modify/evolve X?
Method for analysis or evaluation	How can I evaluate the quality/correctness of X? How do I choose between X and Y?
Design, evaluation, or analysis of a particular instance	How good is Y? What is property X of artifact/method Y? What is a (better) design, implementation, maintenance, or adaptation for application X? How does X compare to Y? What is the current state of X / practice of Y?
Generalization or characterization	Given X, what will Y (necessarily) be? What, exactly, do we mean by X? What are its important characteristics? What is a good formal/empirical model for X? What are the varieties of X, how are they related?
Feasibility study or exploration	Does X even exist, and if so what is it like? Is it possible to accomplish X at all?

Which of these are most common?

Table 2. Types of research questions represented in ICSE 2002 submissions and acceptances			
Type of question	**Submitted**	**Accepted**	**Ratio Acc/Sub**
Method or means of development	142(48%)	18 (42%)	(13%)
Method for analysis or evaluation	95 (32%)	19 (44%)	(20%)
Design, evaluation, or analysis of a particular instance	43 (14%)	5 (12%)	(12%)
Generalization or characterization	18 (6%)	1 (2%)	(6%)
Feasibility study or exploration	0 (0%)	0 (0 %)	(0%)
TOTAL	298(100.0%)	43 (100.0%)	(14%)

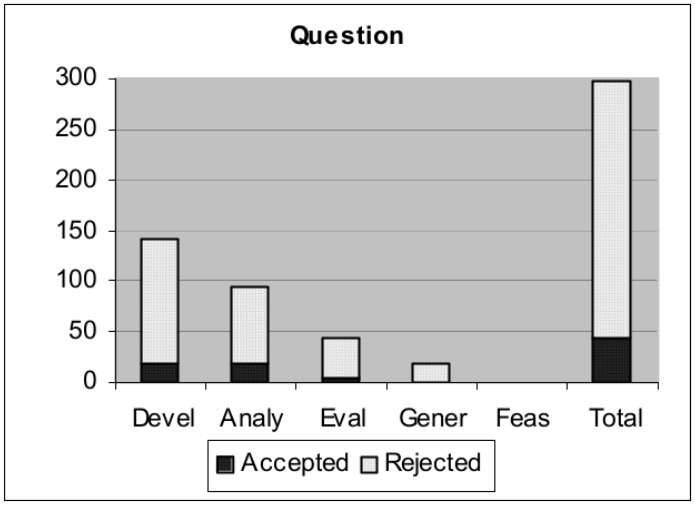


Figure 1. Counts of acceptances and rejections by type of research question

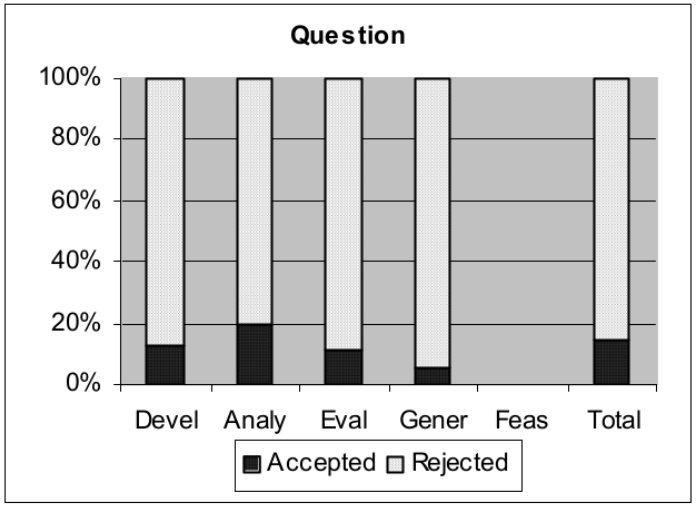


Figure 2. Distribution of acceptances and rejections by type of research question

What do program committees look for?

- A clear statement of the specific problem you solved
- An explanation of how the answer will help solve a problem

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

What is you new result?

- Explain precisely what you have contributed to the store of software engineering knowledge and how this is useful beyond your own project

What kind of results do software engineers produce?

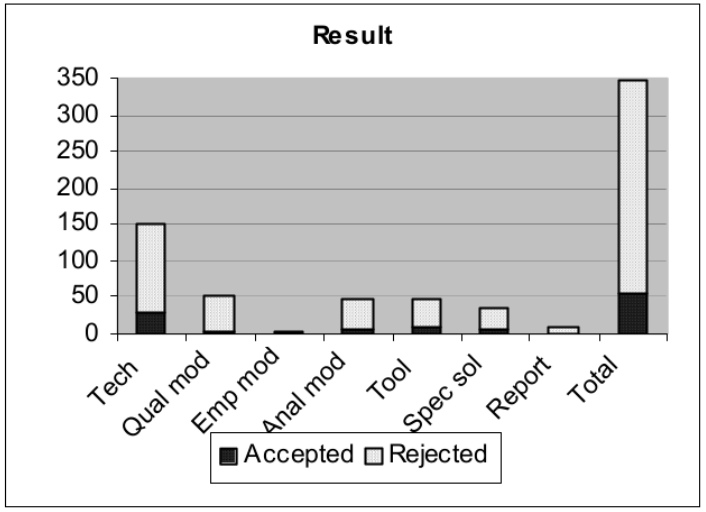
- Procedure or techniques for development or analysis
  - Models that generalize: specific examples, tools, solution, results

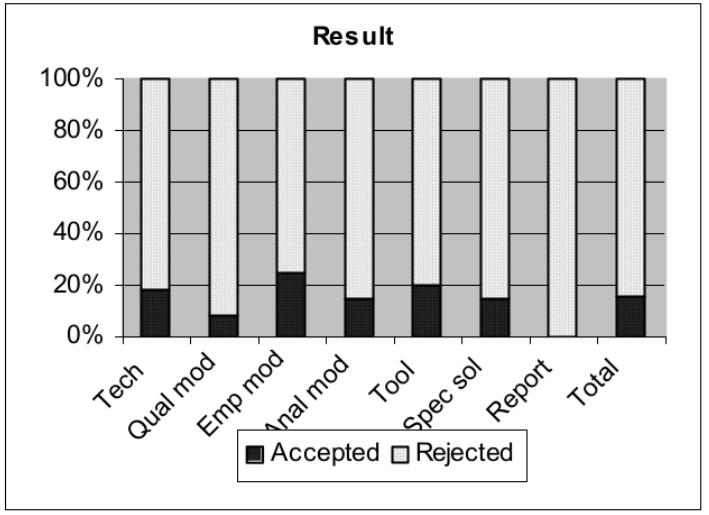
What kind of results do software engineers produce?

Table 3. Types of software engineering research results	
Type of result	**Examples**
Procedure or technique	New or better way to do some task, such as design, implementation, maintenance, measurement, evaluation, selection from alternatives; includes techniques for implementation, representation, management, and analysis; a technique should be operational—not advice or guidelines, but a procedure
Qualitative or descriptive model	Structure or taxonomy for a problem area; architectural style, framework, or design pattern; non-formal domain analysis, well-grounded checklists, well-argued informal generalizations, guidance for integrating other results, well-organized interesting observations
Empirical model	Empirical predictive model based on observed data
Analytic model	Structural model that permits formal analysis or automatic manipulation
Tool or notation	Implemented tool that embodies a technique; formal language to support a technique or model (should have a calculus, semantics, or other basis for computing or doing inference)
Specific solution, prototype, answer, or judgment	Solution to application problem that shows application of SE principles – may be design, prototype, or full implementation; careful analysis of a system or its development, result of a specific analysis, evaluation, or comparison
Report	Interesting observations, rules of thumb, but not sufficiently general or systematic to rise to the level of a descriptive model.

Which of these are most common?

Table 4. Types of research results represented in ICSE 2002 submissions and acceptances			
Type of result	**Submitted**	**Accepted**	**Ratio Acc/Sub**
Procedure or technique	152(44%)	28 (51%)	18%
Qualitative or descriptive model	50 (14%)	4 (7%)	8%
Empirical model	4 (1%)	1 (2%)	25%
Analytic model	48 (14%)	7 (13%)	15%
Tool or notation	49 (14%)	10 (18%)	20%
Specific solution, prototype, answer, or judgment	34 (10%)	5 (9%)	15%
Report	11 (3%)	0 (0%)	0%
TOTAL	348(100.0%)	55 (100.0%)	16%





What do program committees look for?

- Interesting, novel, exciting results that significantly enhace our ability to develop and maintain software
  - To know the quality
  - To recognize general principles
  - To analyze properties

What do program committees look for?

- What, precisely, do you claim to contribute?
- What's new here?
- What has been done before? How is your work different or better?
- What, precisely, is the result?

What do program committees look for?

- What, precisely, do you claim to contribute?
  - If your result ought to work on large systems, explain why you believe it scales
  - If you claim your result is “distributed”, it probably should not have a single central controller or server

What do program committees look for?

- What's new here?
  - Be clear about your claim...

Awful	▼	• I completely and generally solved … (unless you actually did!)
Bad	▼	• I worked on galumphing. (or studied, investigated, sought, explored)
Poor	▼	• I worked on improving galumphing. (or contributed to, participated in, helped with)
Good	▲	• I showed the feasibility of composing blitzing with flitzing. • I significantly improved the accuracy of the standard detector. (or proved, demonstrated, created, established, found, developed)
Better	▲	• I automated the production of flitz tables from specifications. • With a novel application of the blivet transform, I achieved a 10% increase in speed and a 15% improvement in coverage over the standard method.

What do program committees look for?

- What's has been done before?
  - Explain the relation to other work clearly

Awful	▼	The galumphing problem has attracted much attention [3,8,10,18,26,32,37]
Bad	▼	Smith [36] and Jones [27] worked on galumphing.
Poor	▼	Smith [36] addressed galumphing by blitzing, whereas Jones [27] took a flitzing approach.
Good	▲	Smith's blitzing approach to galumphing [36] achieved 60% coverage [39]. Jones [27] achieved 80% by flitzing, but only for pointer-free cases [16].
Better	▲	Smith's blitzing approach to galumphing [36] achieved 60% coverage [39]. Jones [27] achieved 80% by flitzing, but only for pointer-free cases [16]. We modified the blitzing approach to use the kernel representation of flitzing and achieved 90% coverage while relaxing the restriction so that only cyclic data structures are prohibited.

What do program committees look for?

- What, precisely, is the result?
  - Explain what your result is and how it works
  - If you introduce a new model, be clear about its power
  - If you introduce a new metric, define it precisely

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

Why should the reader believe your results?

- Show evidence that your result is valid – that actually helps to solve the problem you set out to solve

What kinds of validation do software engineerings do?

Table 5. Types of software engineering research validation

Type of validation	Examples
Analysis	I have analyzed my result and find it satisfactory through rigorous analysis, e.g. … For a formal model … rigorous derivation and proof For an empirical model … data on use in controlled situation For a controlled experiment … carefully designed experiment with statistically significant results
Evaluation	Given the stated criteria, my result... For a descriptive model … adequately describes phenomena of interest … For a qualitative model … accounts for the phenomena of interest… For an empirical model … is able to predict … because …, or … generates results that fit actual data … Includes feasibility studies, pilot projects
Experience	My result has been used on real examples by someone other than me, and the evidence of its correctness/usefulness/effectiveness is … For a qualitative model … narrative For an empirical model or tool … data, usually statistical, on practice For a notation or technique … comparison of systems in actual use
Example	Here's an example of how it works on For a technique or procedure …a "slice of life" example based on a real system … For a technique or procedure …a system that I have been developing … For a technique or procedure … a toy example, perhaps motivated by reality The "slice of life" example is most likely to be convincing, especially if accompanied by an explanation of why the simplified example retains the essence of the problem being solved. Toy or textbook examples often fail to provide persuasive validation, (except for standard examples used as model problems by the field).
Persuasion	I thought hard about this, and I believe passionately that ... For a technique … if you do it the following way, then … For a system … a system constructed like this would … For a model … this example shows how my idea works Validation purely by persuasion is rarely sufficient for a research paper. Note, though, that if the original question was about feasibility, a working system, even without analysis, can suffice
Blatant assertion	No serious attempt to evaluate result. This is highly unlikely to be acceptable

Which of these are most common?

Table 6. Types of research validation represented in ICSE 2002 submissions and acceptances			
Type of validation	**Submitted**	**Accepted**	**Ratio Acc/Sub**
Analysis	48 (16%)	11 (26%)	23%
Evaluation	21 (7%)	1 (2%)	5%
Experience	34 (11%)	8 (19%)	24%
Example	82 (27%)	16 (37%)	20%
Some example, can't tell whether it's toy or actual use	6 (2%)	1 (2%)	17%
Persuasion	25 (8%)	0 (0.0%)	0%
No mention of validation in abstract	84 (28%)	6 (14%)	7%
TOTAL	300(100.0%)	43 (100.0%)	14%

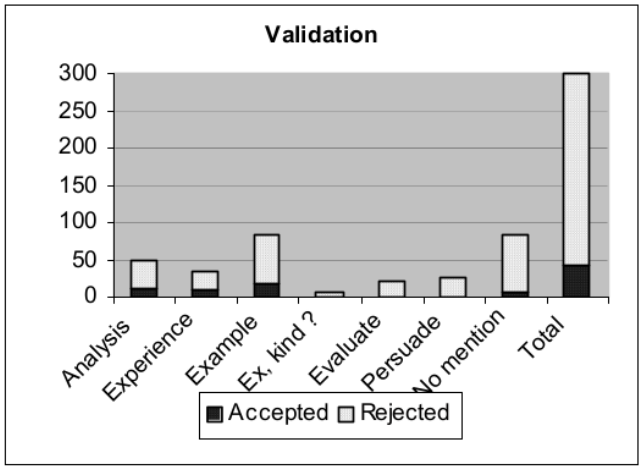


Figure 5. Counts of acceptances and rejections by type of validation

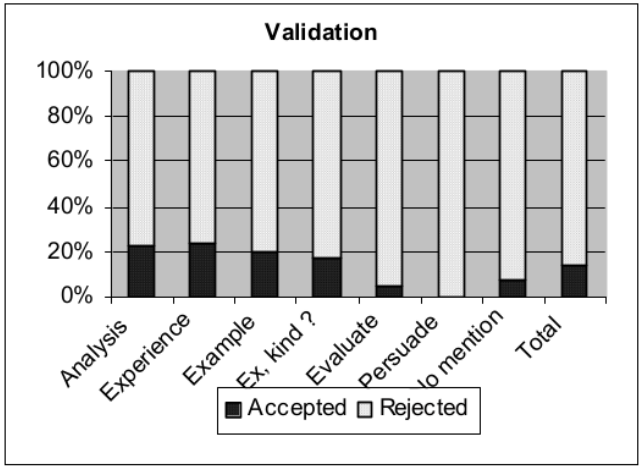


Figure 6. Distribution of acceptances and rejections by type of validation

What do program committees look for?

- Solid evidence to support your result
- It’s not enough that your idea works for you, there must also be evidence that the idea or the technique will help someone else as well

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

How do you combine the elements into a research strategy?

- It is clear that not all combinations of a research question, a result, and a validation strategy lead to good research
- Software Engineering has not developed good general guidance on this question

How do you combine the elements into a research strategy?

Table 7. Paradigms of ICSE2002 acceptances			
Question	**Result**	**Validation**	**#**
Devel method	Procedure	Analysis	2
Devel method	Procedure	Experience	3
Devel method	Procedure	Example	3
Devel method	Qual model	Experience	2
Devel method	Analytic model	Experience	2
Devel method	Notation or tool	Experience	1
Analysis method	Procedure	Analysis	5
Analysis method	Procedure	Evaluation	1
Analysis method	Procedure	Experience	2
Analysis method	Procedure	Example	6
Analysis method	Analytic model	Experience	1
Analysis method	Analytic model	Example	2
Analysis method	Tool	Analysis	1
Eval of instance	Specific analysis	Analysis	3
Eval of instance	Specific analysis	Example	2

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

Does the abstract matter?

- Many of the clearest abstracts had a common struture
  - Two or three sentences about the current state of the art, identifying a particular problem
  - One or two sentences about what this paper contributes to improving the situation
  - One or two sentences about the specific result of the paper and the main idea behind it
  - A sentence about how the result is demonstrated or defended

Agenda

1. Introduction
2. What, precisely, was your contribution?
3. What is your new result?
4. Why should the reader believe your result?
5. How do you combine the elements into a research strategy?
6. Does the abstract matter?
7. Questions you might ask about this report

Questions you might ask about this report

- Is this a sure-fire recipe?
  - No, it’s not a recipe
  - Not all software engineers share the same views of interesting and significant research
  - You can help by explaining your work clearly (this report should help you do that)

Questions you might ask about this report

- Is ICSE different from other conferences?
  - ICSE recognizes several distinct types of technical paper
  - “How to get a Paper Accepted at OOPSLA”
  - “How to Have Your Abstract Rejected” and “Advice to Authors of Extended Abstracts” - SIGSOFT
  - “How to Increase the Chances Your Paper is Accepted at ACM SIGCOMM”

Questions you might ask about this report

- What about this report itself?
  - There is no attempt to show that anyone else can apply the model
  - The model is not justified by any principled
  - Only one conference and one program committe is reflected here
  - There use of abstracts as proxies for full papers is suspect

Questions you might ask about this report

- What about this report itself?
  - This report does meet Brooks' standard for "rules of thumb"