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
The Coming-of-Age of Software Architecture Research

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

Basic Research
Recognize problem, Invent ideas

Concept Formation
Refine ideas, publish solutions

Development & Extension
Try it out, clarify, refine

Internal Exploration
Stabilize, port, use for real problems

External Exploration
Broaden user group, extend

Popularization
Propagate through community

Usable capability
Production quality, commercial support

Seminal paper or system
Outsiders use it

Key Idea

The Coming-of-Age of Software Architecture Research

Software Technology Maturation Points

<table>
<thead>
<tr>
<th>Major Technology Areas</th>
<th>Technology Concepts</th>
<th>Methodology Technology</th>
<th>Consolidated Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Engineering</td>
<td>Structured Programming</td>
<td>SCR Methodology</td>
<td>Smalltalk 80</td>
</tr>
<tr>
<td>Compiler Construction</td>
<td>Abstract Data Types</td>
<td>DOD-STD-SDS</td>
<td>Cost Models</td>
</tr>
<tr>
<td>Knowledge-Based Systems</td>
<td>Metrics</td>
<td>AFR 800-14</td>
<td>Unix</td>
</tr>
<tr>
<td>Verification</td>
<td></td>
<td></td>
<td>SREM</td>
</tr>
</tbody>
</table>

Maturation Times

<table>
<thead>
<tr>
<th>Technology</th>
<th>Maturation Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFR 800-14</td>
<td>6 years</td>
</tr>
<tr>
<td>unix</td>
<td>3 yrs</td>
</tr>
<tr>
<td>Structured Programming</td>
<td>4 yrs</td>
</tr>
<tr>
<td>Verification</td>
<td></td>
</tr>
<tr>
<td>Abstract Data Types</td>
<td></td>
</tr>
<tr>
<td>SREM</td>
<td></td>
</tr>
<tr>
<td>Compiler Construction</td>
<td>3 yrs</td>
</tr>
<tr>
<td>SCR Methodology</td>
<td></td>
</tr>
<tr>
<td>Smalltalk 80</td>
<td>6 yrs</td>
</tr>
<tr>
<td>Cost Models</td>
<td></td>
</tr>
<tr>
<td>Software Engineering</td>
<td>5 yrs</td>
</tr>
<tr>
<td>Metrics</td>
<td></td>
</tr>
<tr>
<td>Knowledge-Based Systems</td>
<td>8 yrs</td>
</tr>
<tr>
<td>DOD-STD-SDS</td>
<td># of years</td>
</tr>
</tbody>
</table>
### Phase Times and Publications

<table>
<thead>
<tr>
<th>Phase</th>
<th>Basic Resch</th>
<th>Concept Form</th>
<th>Dev + Ext</th>
<th>Internal Exp</th>
<th>External Exp</th>
<th>Popularize</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 years</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>12 years</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>14 years</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16 years</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>18 years</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

**Typical publication venues**
- Research workshops
- Conferences
- Archival journals
- Reviews
- Development workshops
- Popular journals
- Trade 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
The Coming-of-Age of Software Architecture Research

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?


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

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

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.

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 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
The Coming-of-Age of Software Architecture Research

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
The Coming-of-Age of Software Architecture Research

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

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

ICSE 2002 submissions

<table>
<thead>
<tr>
<th>Type of question</th>
<th>Submitted</th>
<th>Accepted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method or means of development</td>
<td>142 (48%)</td>
<td>18 (42%)</td>
</tr>
<tr>
<td>Method for analysis or evaluation</td>
<td>9 (32%)</td>
<td>19 (44%)</td>
</tr>
<tr>
<td>Design, evaluation, or analysis of a particular instance</td>
<td>43 (14%)</td>
<td>5 (12%)</td>
</tr>
<tr>
<td>Generalization or characterization</td>
<td>18 (6%)</td>
<td>1 (2%)</td>
</tr>
<tr>
<td>Feasibility study or exploration</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>298 (100%)</td>
<td>43 (100%)</td>
</tr>
</tbody>
</table>

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

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

## ICSE 2002 submissions

<table>
<thead>
<tr>
<th>Type of result</th>
<th>Submitted</th>
<th>Accepted</th>
<th>2003 Ratio Acc/Sub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure or technique</td>
<td>152 (44%)</td>
<td>28 (51%)</td>
<td>18%</td>
</tr>
<tr>
<td>Qualitative or descriptive model</td>
<td>50 (14%)</td>
<td>7 (13%)</td>
<td>15%</td>
</tr>
<tr>
<td>Empirical model</td>
<td>5 (1%)</td>
<td>1 (2%)</td>
<td>20%</td>
</tr>
<tr>
<td>Analytic model</td>
<td>49 (14%)</td>
<td>7 (13%)</td>
<td>15%</td>
</tr>
<tr>
<td>Tool or notation</td>
<td>3 (9%)</td>
<td>1 (2%)</td>
<td>25%</td>
</tr>
<tr>
<td>Specific solution, prototype, answer, or judgment</td>
<td>4 (11%)</td>
<td>5 (9%)</td>
<td>15%</td>
</tr>
<tr>
<td>Report</td>
<td>1 (3%)</td>
<td>0 (0%)</td>
<td>0%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>348 (100%)</td>
<td>55 (100%)</td>
<td>16%</td>
</tr>
</tbody>
</table>
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 ...
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Formal model: rigorous derivation and proof</td>
</tr>
<tr>
<td></td>
<td>Empirical model: data on use in controlled situation</td>
</tr>
<tr>
<td></td>
<td>Controlled experiment: carefully designed statistical experiment</td>
</tr>
<tr>
<td>Experience</td>
<td>My result has actually been used; the evidence is</td>
</tr>
<tr>
<td></td>
<td>Qualitative model: narrative</td>
</tr>
<tr>
<td></td>
<td>Empirical model, tool: data, usually statistical, on practice</td>
</tr>
<tr>
<td></td>
<td>Notation, technique: comparison of systems in actual use</td>
</tr>
<tr>
<td>Example</td>
<td>Here’s how my result works on a small example</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Given these criteria, my result ...</td>
</tr>
<tr>
<td></td>
<td>Descriptive model: adequately describes phenomena of interest</td>
</tr>
<tr>
<td></td>
<td>Empirical model: is able to predict … because …</td>
</tr>
<tr>
<td>Persuasion</td>
<td>I thought hard about this, and I believe...</td>
</tr>
<tr>
<td>Blatant assertion</td>
<td>No serious attempt to evaluate result</td>
</tr>
</tbody>
</table>
The Coming-of-Age of Software Architecture Research

ICSE 2002 submissions

<table>
<thead>
<tr>
<th>Type of validation</th>
<th>Submitted</th>
<th>Accepted</th>
<th>2003 Ratio Acc/Sub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>48 (16%)</td>
<td>11 (26%)</td>
<td>23%</td>
</tr>
<tr>
<td>Evaluation</td>
<td>21 (7%)</td>
<td>1 (2%)</td>
<td>5%</td>
</tr>
<tr>
<td>Experience</td>
<td>34 (11%)</td>
<td>8 (19%)</td>
<td>24%</td>
</tr>
<tr>
<td>Example</td>
<td>88 (27%)</td>
<td>16 (37%)</td>
<td>20%</td>
</tr>
<tr>
<td>Some examples, can't tell whether it's toy or actual use</td>
<td>6 (2%)</td>
<td>1 (2%)</td>
<td>17%</td>
</tr>
<tr>
<td>Persuasion</td>
<td>24 (8%)</td>
<td>0 (0%)</td>
<td>0%</td>
</tr>
<tr>
<td>No mention of validation in abstract</td>
<td>84 (28%)</td>
<td>6 (14%)</td>
<td>7%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>300 (100%)</td>
<td>43 (100%)</td>
<td>14%</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Question</th>
<th>Strategy/Result</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>Proc/technique</td>
<td>Analysis</td>
</tr>
<tr>
<td>Analysis</td>
<td>Qual/desc model</td>
<td>Experience</td>
</tr>
<tr>
<td>Evaluate instance</td>
<td>Analytic model</td>
<td>Example</td>
</tr>
<tr>
<td>Generalization</td>
<td>Empirical model</td>
<td>Evaluation</td>
</tr>
<tr>
<td>Feasibility</td>
<td>Specific solution</td>
<td>Persuasion</td>
</tr>
<tr>
<td></td>
<td>Report</td>
<td></td>
</tr>
</tbody>
</table>
The Coming-of-Age of Software Architecture Research

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

Real World Practical problem

Research Setting Idealized problem

Research product (technique, method, model, system, …)

Real World Solution to practical problem

Validation Task 1:
Does the product solve the idealized problem?

Validation Task 2:
Does the result help to solve the practical problem?

**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

**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


**Question** (Analysis 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

*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
The Coming-of-Age of Software Architecture Research

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

Sometimes a breakthrough
(but sometimes nonsense)
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”

<table>
<thead>
<tr>
<th>Question</th>
<th>Strategy/Result</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developmnet method</td>
<td>Proc/technique</td>
<td>Analysis</td>
</tr>
<tr>
<td>Analysis method</td>
<td>Qualitative model</td>
<td>Experience</td>
</tr>
<tr>
<td>Evaluate instance</td>
<td>Analytic model</td>
<td>Example</td>
</tr>
<tr>
<td>Generalization</td>
<td>Empirical model</td>
<td>Evaluation</td>
</tr>
<tr>
<td>Feasibility</td>
<td>Tool/notation</td>
<td>Persuasion</td>
</tr>
<tr>
<td></td>
<td>Specific solution</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Report</td>
<td></td>
</tr>
</tbody>
</table>

Pro Forma Research Strategies

Locating the pro forma abstracts in research strategy space

<table>
<thead>
<tr>
<th>Developmnet Meth</th>
<th>Analysis Meth</th>
<th>Instance</th>
<th>Gener-Feas-</th>
<th>Anal-</th>
<th>Experience</th>
<th>Example</th>
<th>Evaluation</th>
<th>Persua-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proc, Tech</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qual Model</td>
<td>Emp Model</td>
<td></td>
<td>EM</td>
<td>Emp</td>
<td>EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emp Model</td>
<td>Anal Model</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anal Model</td>
<td>Notation</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notation</td>
<td>Spec Soln</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spec Soln</td>
<td>Report</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td>EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Coming-of-Age of Software Architecture Research

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](http://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