Research Methods in Computer Science

(Serge Demeyer — University of Antwerp)

AnSyMo

Antwerp Systems and software Modelling
http://ansymo.ua.ac.be/

Universiteit Antwerpen

Helicopter View

(Ph.D.)
Research

How to *perform* research ?
(and get "empirical" results)

How to *write* research ?
(and get papers accepted)

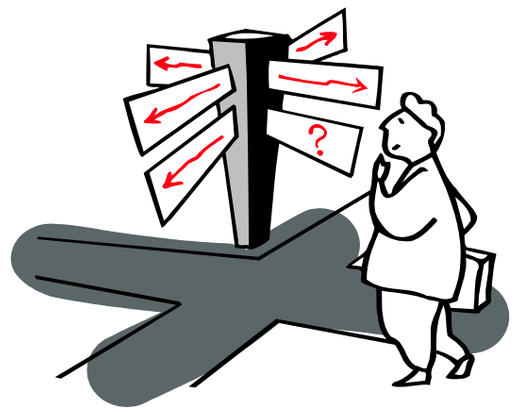
How many of you have a systematic way of writing an abstract ?

Personal Opinion Disclaimer

Sometimes I will give advice based on personal experience or representing a particular school of thought. These do not necessarily confirm with what your supervisor says !

Such opinions will be flagged with the Personal Opinion Disclaimer.

2. Reporting & Reviewing Research



Introduction
- The Publication Process
  + Publication Categories
  + Quality indicators

The Review Process
- Identify the Champion
- Implications for Authors
  + The 4-line abstract rule
  + The fish model
  + Natural emphasis of paragraphs
- Things to avoid
  + Method vs. Methodology

The Task of the referee
- Questions to answer ⇒ Review Template

Once Accepted …
- Tips and Tricks

Conclusion

Publications: Output Measure

"If I have seen a little further it is by standing on the shoulders of Giants."

(Isaac newton)

"Are We Polishing a Round Ball?"

(Michael Stonebraker; Panel abstract — Proceedings of the Ninth International Conference on Data Engineering)

Sceptic perspective:
- the quest for the "least publishable unit"
- "publish or perish"

"And since dissertations can be written about everything under the sun, the number of topics is infinite. Sheets of paper covered up with words pile up in archives sadder than cemeteries, because no one ever visits them, not even on All Souls' Day. Culture is perishing in overproduction, in an avalanche of words, in the madness of quantity. That's why one banned book in your former country means infinitely more than the billions of words spewed out by our universities."

(Milan Kundera, The Unbearable Lightness of Being; Part Three: Words Misunderstood — Sabina's Country)

Publication Categories

Journal Publications
- a1) citation index (ISI web of science)
- a2) international; peer reviewed
- a3) national; peer reviewed
- a4) other

source: guidelines for project reports
FWO (Research Fund Flanders)

Books
- b1) book
- b2) chapter
- a3) editor (incl. proceedings)

Comparing apples and oranges

International vs. National
- inherently regional research (law, politics, …)
- vulgarizing research
- scientists taking position in society debates

Publication Culture
- co-authorship (e.g. alphabetical sorting)
- citation behavior
- half-life time of ideas

Other
- c1) articles in proceedings
- c2) technical reports; extended abstracts; thesis
- c3) patents

Publication Categories — Computer Science

Journal Publications
- citation index (ISI web of science)
- international; peer reviewed

Conference Publications
- peer reviewed (acceptance ratio)

Books
- book
- editor (incl. proceedings)
- chapter

Artifacts
- tools
- patents

Other
- workshops
- technical reports; extended abstracts; thesis

Artifacts ???
- Computer Science and Telecommunications Board, C. 1994. Academic careers for experimental computer scientists and engineers. *Communications of the ACM* 37, 4 (Apr. 1994), 87-90.

The Pipeline Model

new, unpolished idea

good, valuable idea

archival reference

workshops → conferences → journals

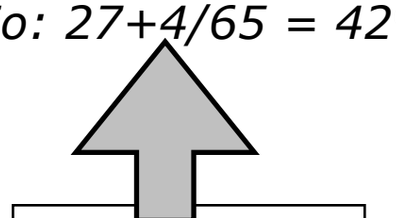
Typical for computer science.
Not in other scientific disciplines.

Quality Indicators

Proceedings: Acceptance Ratio

- Andy Zaidman, Bart Van Rompaey, Serge Demeyer, and Arie van Deursen. Mining software reposito- ries to study co-evolution of production and test code. In Proceedings ICST'08 (The 1st International Conference on Software Testing, Verification and Validation), pages 220–229. IEEE, 2008.
  [Acceptance ratio: 37/147 = 25%]

- Andy Zaidman, Bram Adams, Kris De Schutter, Serge Demeyer, Ghislain Hoffman, and Bernard De Ruyck. Regaining lost knowledge through dynamic analysis and aspect orientation - an industrial ex- perience report. In Proceedings CSMR'06 (the 10th Conference on Software Maintenance and Reengi- neering), pages 89–98. IEEE Computer Society, 2006.
  [Acceptance ratio: 27+4/65 = 42%]



Journal Publications: Impact factor

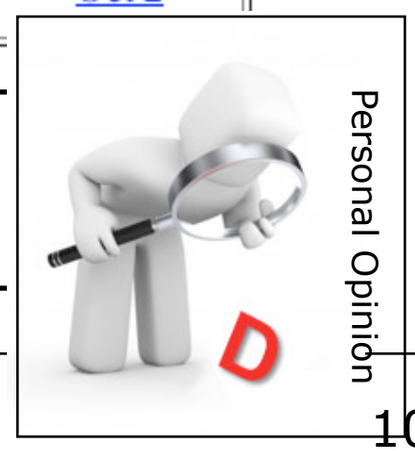
- Bart Van Rompaey, Bart Du Bois, Serge Demeyer, and Matthias Rieger. On the detection of test smells: A metrics-based approach for general fixture and eager test. Transactions on Software Engineering, 33(12):800–817, 2007.
  [SCI impact factor 1.967, ranked 7 / 79]

Acceptance Rates

Source http://people.engr.ncsu.edu/txie/seconferences.htm

Top General SE Conferences	ICSE	FSE/ESEC	ASE	OOPSLA	ECOOP	ISSTA	FASE
2009	50/405(12%)	32+7/217(15%)	38+33/222(17%)	25/144(17%)	25/117(21%)	25/93(27%)	30/124(24%)
2008	56/371(15%)	31/152(20%)	34+36/280(12%)	33/117(28%)	27/138(20%)	26+9/100(26%)	?(26%)
2007	49/334(15%)	43+20/251(17%)	37+40/312(12%)	33/156(21%)	25/160(16%)	22/10?(21%)	30/141(21%)
2006	36/395(9%)	25/125(20%)	22+12/121(18%)	26/157(17%)	21/160(13%)	22/84(26%)	27/166(17%)
2005	44/313(14%)	32/201(16%)	28+35/291(10%)	32/174(18%)	24/172(14%)	--	22/99 (22%)
2004	58/436(13%)	25/169(15%)	25+26/183(14%)	27/173(16%)	25/132(19%)	26+2/93(28%)	22/91(24%)
2003	42/324(13%)	33+9/168(20%)	22+20/170(13%)	26/147(18%)	18/88(20%)	--	20/89(22%)
2002	48/303(15%)	17/128(13%)	19+19/94(20%)	25/125(20%)	24/96(25%)	18+8/97(19%)	21/60(35%)
2001	47/268(18%)	29/137(21%)	32+28/164(20%)	27/145(18%)	18/108(17%)	--	22/74(30%)
2000	49/335(14%)	17/92(18%)	23+22/100(23%)	26/142(18%)	20/109(20%)	17+4/73(23%)	21/60(35%)
1999	50/269(19%)	29/141(21%)	25+25/123(20%)	30/152(20%)	20/183(11%)	--	13/?
1998	41/209(20%)	19%	24+20/150(16%)	?	24/124(19%)	16/47(34%)	18/59(31%)
1997	50/219(23%)	27/194(14%)	32+15/108(30%)	?	20/103(19%)	--	?
1996	52/213(24%)	?	?	16%	21/173(12%)	16+8/69(23%)	?
1995	28/155(18%)	29/150(19%)	?		18/90(20%)	--	?
Submission Deadline	Aug 29	March 16	May 4	March 19	Dec 17	Jan 30	Oct 2

- [100% - 50%[: not selective
- [50% - 30%[: reasonably selective
- [30% - 15%[: selective
- [15% - 0%[: too selective !?



Personal Opinion

Impact Factor — Citation Index

ISI Web of Knowledge℠

Journal Citation Reports®

WELCOME | HELP | MARKED LIST

2008 JCR Science Edition

Journal Summary List

Journal Title Changes

Journals from: subject categories COMPUTER SCIENCE, SOFTWARE ENGINEERING | VIEW CATEGORY SUMMARY LIST

Sorted by: Impact Factor | SORT AGAIN

Journals 1 - 20 (of 86) | [1 | 2 | 3 | 4 | 5] | Page 1 of 5

MARK ALL | UPDATE MARKED LIST

Ranking is based on your journal and sort selections.

Mark	Rank	Abbreviated Journal Title (linked to journal information)	ISSN	JCR Data						*Eigenfactor*™ Metrics	
				Total Cites	Impact Factor	5-Year Impact Factor	Immediacy Index	Articles	Cited Half-life	*Eigenfactor*™ Score	*Article Influence*™ Score
☑	1	ACM T SOFTW ENG METH	1049-331X	729	3.958	4.293	0.261	23	7.8	0.00165	1.284
☑	2	IEEE T SOFTWARE ENG	0098-5589	5449	3.569	4.241	0.423	52	>10.0	0.00695	0.956
☑	3	ACM T GRAPHIC	0730-0301	4083	3.383	4.997	0.150	107	4.7	0.02625	2.045
☑	4	J WEB SEMANT	1570-8268	438	3.023		0.414	29	3.8	0.00288	
☑	5	COMMUN ACM	0001-0782	12617	2.646	3.175	0.377	146	>10.0	0.01794	0.949
☑	6	IEEE MICRO	0272-1732	1478	2.565	2.848	0.278	36	6.4	0.00445	0.874
☑	7	ACM T MULTIM COMPUT	1551-6857	155	2.465		0.037	27	2.6	0.00110	
☑	8	IEEE T VIS COMPUT GR	1077-2626	2224	2.445	2.706	0.302	162	4.1	0.01075	0.956
☑	9	J ACM	0004-5411	5727	2.339	3.444	0.250	28	>10.0	0.00622	1.733
☑	10	MATH PROGRAM	0025-5610	4658	2.336	2.745	0.589	73	>10.0	0.01722	1.886
☑	11	IEEE INTERNET COMPUT	1089-7801	1568	2.309	3.245	0.436	55	5.4	0.00542	0.879
☑	12	IEEE T MULTIMEDIA	1520-9210	2010	2.288	2.932	0.160	144	3.9	0.00957	0.867
☑	13	IEEE MULTIMEDIA	1070-986X	708	2.258	2.189	0.069	29	6.0	0.00243	0.689
☑	14	ACM T MATH SOFTWARE	0098-3500	2111	2.197	3.361	0.526	38	>10.0	0.00581	1.820
☑	15	IEEE SOFTWARE	0740-7459	2371	2.099	2.732	0.388	67	7.6	0.00445	0.671
☑	16	COMPUTER	0018-9162	3133	2.093	2.591	0.357	84	6.9	0.01094	0.979
☑	16	IEEE T DEPEND SECURE	1545-5971	381	2.093	3.896	0.222	18	3.8	0.00228	1.072
☑	18	J DATABASE MANAGE	1063-8016	263	2.000		1.368	19	3.6	0.00076	
☑	19	IBM SYST J	0018-8670	1599	1.883	2.124	0.729	48	7.7	0.00243	0.456
☑	20	IEEE COMPUT GRAPH	0272-1716	1930	1.866	2.301	0.220	41	9.6	0.00377	0.813

MARK ALL | UPDATE MARKED LIST

Journals 1 - 20 (of 86) | [1 | 2 | 3 | 4 | 5] | Page 1 of 5

The h-index

Represent both

- scientific productivity
- scientific impact

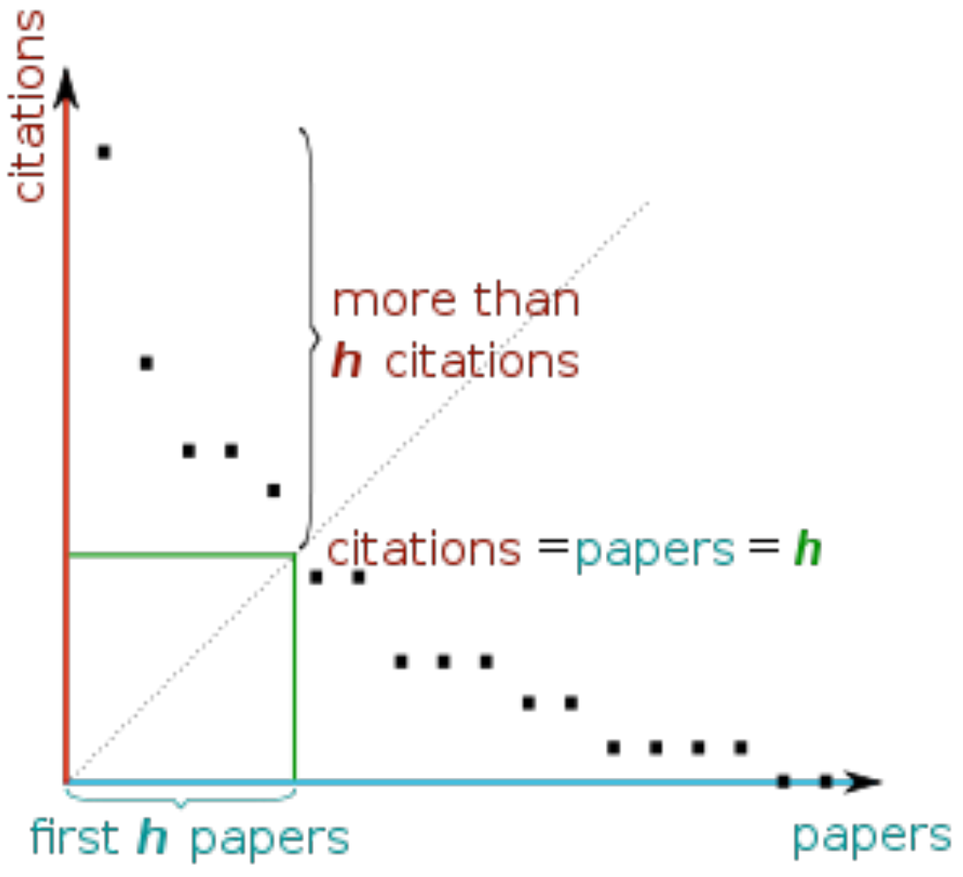
⇒ **in a single number (measurement)**

A scientist has index h if

- h of [his/her] Np papers have *at least* h citations each, and
- the other (Np − h) papers have *at most* h citations each.

Sources to calculate

- ISI web of knowledge http://isiknowledge.com/
- UAD - Search http://quadsearch.csd.auth.gr/



Quality Indicators — Beware

- impact factor of journal ≠ impact factor of article
  - + Seglen PO (1997). "Why the impact factor of journals should not be used for evaluating research". BMJ 314 (7079): 498–502.
  - + Joint Committee on Quantitative Assessment of Research (June 12, 2008). "Citation Statistics". International Mathematical Union.

- #citations ≠ impact
  - + Carlo Ghezzi; Reflections on 40+ years of software engineering research and beyond an insider's view (ICSE 2009, keynote)

- "The widespread practice of counting publications without reading and judging them is fundamentally flawed."
  - + Parnas, D. L. 2007. Stop the numbers game. Commun. ACM 50, 11 (Nov. 2007)

- "If used unwisely, as is increasingly the case, they discourage people (young ones in particular) right from the outset from daring to think, from exploring new paths […]"
  - + Math. Struct. in Comp. Science Editorial Board; Math. Struct. in Comp. Science (2009), vol. 19, pp. 1–4.

The Reviewer

- volunteer
  + don’t waste his/her time

- curious
  + catch his/her interest

- constructive
  + supervises other Ph.D.

- influential
  + wants to support “valuable” papers

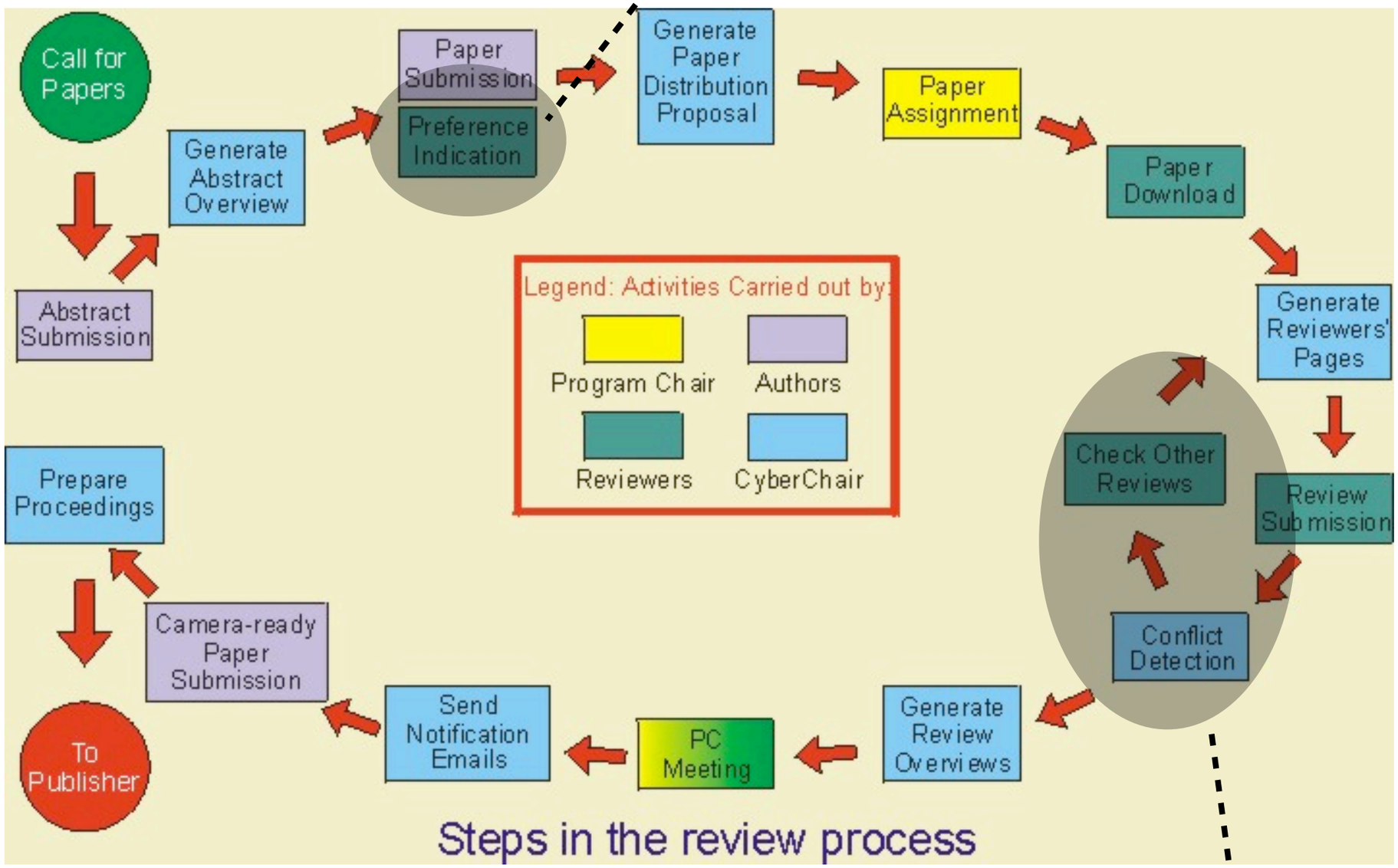
- anonymous
  + avoid tampering

… unfortunately …

- busy
  + read’s on train, bus, air-plane, …

Review Process Steps

Bidding for Abstracts
abstracts + key-words
= "first date" with your reviewer



source: CyberChair (http://www.CyberChair.org)

Identify the Champion
your reviewer needs arguments to support your paper

Providing Keywords

	▪ Automated reasoning techniques
	▪ Component-based systems
	▪ Computer-supported cooperative work
	▪ Configuration management
	▪ Domain modelling and meta-modelling
	▪ Empirical software engineering
	▪ Human-computer interaction
	▪ Knowledge acquisition and management
	▪ Maintenance and evolution
	▪ Model-based software development
	▪ Model-driven engineering and model transformation
	▪ Modeling language semantics
	▪ Open systems development
	▪ Product line architectures
	▪ Program understanding
	▪ Program synthesis
	▪ Program transformation
	▪ Re-engineering
	▪ Requirements engineering
	▪ Specification languages
	▪ Software architecture and design
	▪ Software visualization
	▪ Testing, verification, and validation
	▪ Tutoring, help, and documentation systems

As many as possible ?
vs. As few as possible ?

Writing Abstracts

Descriptive Abstract

- outlines the topics covered in a piece of writing
  + reader can decide whether to read entire document
- ≈ table of contents in paragraph form.

Informative Abstract

- provides detail about the substance of a piece of writing
  + readers remember key findings
  + reviewers find the claims
- ≈ claim and supporting evidence in paragraph form

≠ executive summary
(abstracts use *the same* level of technical language)

4-line abstract guideline

- source: Kent Beck “How to Get a Paper Accepted at OOPSLA”
  [http://lore.ua.ac.be/Teaching/ThesisMaster/BeckAbstract.html]

- 1) states the problem
  + WHO is suffering the problem ?
  + Connect with your target audience
- 2) why the problem is a problem
  + WHY is it a problem ?
  + Cost / Art rather than a science / …
- 3) startling sentence
  + WHAT is the claimed solution ?
  + the one thing to say that will catch interest
    … and that you will actually demonstrate in the paper
    ➡ must be falsifiable
- 4) the implication of my startling sentence
  + WHERE can we use this solution ?
  + implications for society, community, other researchers, …

Identify The Champion (1/2)

- source: Oscar Nierstrasz, “Identify the Champion,” in Pattern Languages of Program Design 4

- Make Champions Explicit
  + A: Good paper. I will champion it at the PC meeting.
  + B: OK paper, but I will not champion it.
  + C: Weak paper, though I will not fight strongly against it.
  + D: Serious problems. I will argue to reject this paper.

    ➡ “The most important thing for a reviewer to decide is whether he or she thinks that the paper is worth defending at the PC meeting, not whether it is a great paper or not.”

- Make Experts Explicit
  + X: I am an expert in the subject area of this paper.
  + Y: I am knowledgeable in the area, though not an expert.
  + Z: My evaluation is that of an informed outsider.

    ➡ detect inexpert champion — expert fence-sitter

These scores are *not* revealed to the authors

Identify The Champion (2/2)

- Identify the Conflicts (classify according to extreme reviews)
  + AA, AB: All reviews are positive, at least one champion.
  + AC: Likely accept; at least one champion, and no strong detractor.
  + AD: This is a serious conflict, and will certainly lead to debate.
  + BC: Borderline papers, no strong advocate nor a detractor.
  + BD: Likely to be rejected.
  + CC, CD, DD: Almost certain rejects.

- inexpert champion
  + If all champions are Y (or Z)
  + If all reviews are Y or Z
    ➡ solicit extra review

- expert fence-sitters
  + Experts tend to be more critical
    ➡ B or even C ratings by X may turn out to be champions
    (remember: PC members want to influence the research)

Example: Easychair

- Clear accept at top
- Clear reject at the bottom (not shown)
- middle area: to discuss

#	title	scores	avg	decision
109	Stochastic Simulation of Graph Transformation Systems	**3**(3),**2**(3),**3**(3)	2.7	ACCEPT
41	Performance modeling and analysis of context-aware mobile software systems	**2**(2),**2**(3),**2**(2)	2.0	ACCEPT
14	Proving Consistency and Completeness of Model Classes Using Theory Interpretation	**2**(3),**3**(3),**0**(3)	1.7	accept?
34	An Automata-Theoretic Approach to Hardware/Software Co-verification	**2**(2),**1**(2),**2**(2)	1.7	ACCEPT
67	Automatic Cross Validation of Multiple Specifications: A Case Study	**2**(3),**1**(2),**2**(2)	1.7	ACCEPT
82	Operation-based, Fine-grained Version Control Model for Tree-based Representation	**1**(4),**2**(1),**2**(3)	1.7	ACCEPT
117	Reusing Model Transformations while Preserving Properties	**1**(2),**2**(3),**2**(3)	1.7	accept?
73	Memory Leaks Detection in Java by Bi-Abductive Inference	**2**(1),**1**(2)	1.5	
94	Dynamic Resource Scheduling in Disruption-Prone Software Development Environments	**1**(1),**1**(2),**2**(1)	1.3	ACCEPT
124	A Lightweight and Portable Approach to Making Concurrent Failures Reproducible	**2**(3),**0**(2),**2**(2)	1.3	ACCEPT
25	Shape Refinement through Explicit Heap Analysis	**1**(4),**1**(2),**1**(2)	1.0	ACCEPT
106	Evaluating Ordering Heuristics for Dynamic Partial-Order Reduction Techniques	**0**(2),**2**(2),**1**(3)	1.0	ACCEPT
44	A Process to Identify "Guilty" Antipatterns in Software Performance Analysis	**0**(3),**1**(2),**1**(2)	0.7	accept?
46	Incremental Service Composition based on Partial Matching of Visual Contracts	**1**(2),**1**(2),**0**(1)	0.7	accept?
57	Real-Time Model Transformations in MOMENT2	**1**(2),**0**(4),**1**(3)	0.7	accept?
83	Formal Analysis and Verification of Self-Healing Systems	**1**(3),**1**(4),**0**(2)	0.7	accept?
93	Specification-based Slicing and Slice Graphs	**-1**(3),**1**(3),**2**(3)	0.7	reject?
96	A Framework for Defining the Semantics of Big-Step Modelling Languages	**3**(3),**1**(2),**-2**(3)	0.7	
108	Efficient Runtime Assertion Checking of Assignable Clauses with Datagroups	**2**(4),**1**(2),**-1**(3)	0.7	accept?
115	A Method for Analyzing Code Homology in Genealogy of Evolving Software	**2**(2),**0**(4),**0**(4)	0.7	accept?
23	Model Migration with Epsilon Flock	**2**(4),**-1**(3),**0**(3)	0.3	accept?
38	Synthesis-Based Loose Programming in the JABC	**1**(2),**1**(2),**-1**(3)	0.3	reject?
63	A Modular Model Composition Technique	**-2**(4),**2**(3),**1**(3)	0.3	
66	Reasoning about Function Objects	**-2**(3),**2**(2),**1**(2)	0.3	
78	A Formalisation of Constraint-Aware Model Transformations	**1**(2),**-2**(4),**2**(3)	0.3	
81	Efficient State Space Exploration: Interleaving Stateless and State-based Model Checking	**-1**(1),**1**(2),**1**(2)	0.3	
87	Analyzing the Impact of Changes in Multi-threaded Programs	**2**(3),**0**(3),**-1**(2)	0.3	
99	An Algebra of Hierarchical Graphs	**1**(2),**-1**(4),**1**(3)	0.3	
132	Generating Models of Communication Protocols using Regular Inference with Abstraction	**-1**(4),**1**(2),**1**(3)	0.3	
5	A Verifiable Modeling Approach to Configurable Role-Based Access Control	**-3**(2),**2**(2),**1**(3)	0.0	
55	A Formal Approach to Modelling Time Properties of Service-Oriented Systems	**1**(3),**-1**(2)	0.0	
74	Flexible Automated Synthesis of Models	**1**(2),**1**(4),**-2**(3)	0.0	
75	A Framework for Hybrid Automata Learning	**1**(2),**-1**(1),**0**(4)	0.0	reject?
86	From Memoryless Orchestrators Based on Global Priorities to Distributed Protocols	**-1**(3),**2**(2),**-1**(2)	0.0	reject?
105	Designing a Representation of Software Variation	**1**(2),**-1**(2),**0**(3)	0.0	
35	Interactive Completion of Scenario Specifications	**2**(1),**-1**(3),**-2**(4)	-0.3	reject?
40	Learning Workflow Petri Nets	**1**(3),**0**(1),**-2**(4)	-0.3	reject?
56	Reverse Engineering of GUI Models for Testing	**1**(4),**-1**(2),**-1**(2)	-0.3	reject?
61	Supporting Reuse Mechanisms for Developments in Event-B: Composition	**-1**(4),**0**(3),**0**(1)	-0.3	reject?
90	On Precise Modeling of Regular Replacement	**-2**(1),**1**(1),**0**(2)	-0.3	reject?
102	Automatic Requirements Extraction from Test Cases	**1**(2),**-2**(3),**0**(4)	-0.3	REJECT
133	Version Management of Business Process Models with Dynamic Computation of Position Parameters	**-1**(3),**2**(2),**-2**(4)	-0.3	
1	Minimal Antagonistic Sets of Software Components	**2**(1),**-2**(1),**-2**(2)	-0.7	
16	Specifying time-sensitive systems with TLA+	**0**(3),**-2**(2),**0**(3)	-0.7	
18	Non-local Choice and Implied Scenarios	**1**(2),**-1**(1),**-2**(3)	-0.7	reject?
21	Refinement Patterns for the Top-Down Development of Statecharts	**-1**(3),**0**(3),**-1**(3)	-0.7	reject?
39	Process Synthesis in Practice	**-2**(3),**-2**(3),**2**(2)	-0.7	
47	A Tool for Fractal Components Reconfiguration Based on Events	**-1**(1),**-2**(4),**1**(2)	-0.7	reject?
77	An Architecture for Certifying the Results of Invariant-Based Verification Tools	**-1**(2),**0**(2),**-1**(2)	-0.7	reject?
88	Automated Inference of Risk Association Rules via Mining Risk Analysis Documents	**0**(2),**-2**(2),**0**(2)	-0.7	reject?
91	From Lenses to Tiles: Model Synchronization via Double Categories	**-2**(4),**-2**(2),**2**(4)	-0.7	
101	Scenarios-based Testing of Systems with distributed Ports	**0**(3),**-2**(3),**0**(2)	-0.7	reject?
103	Model Synthesis for Parametric Analysis of Real-Time System Designs	**-1**(3),**-2**(1),**1**(3)	-0.7	reject?
119	Formal Analysis of Scenario Aggregation	**-1**(2),**-2**(3),**1**(2)	-0.7	reject?
122	Synthesis of Greedy Algorithms From Dominance Relations	**-2**(1),**1**(2),**-1**(2)	-0.7	
125	Automatic generation of model checking script based on environment modeling	**-2**(3),**0**(2),**0**(2)	-0.7	reject?
129	System testing of service oriented systems with Telling TestStories	**-1**(3),**0**(4),**-1**(3)	-0.7	reject?
2	Compensation and forward recovery in long-running transactions	**0**(2),**-1**(1),**-2**(4)	-1.0	reject?
27	Evaluating UML Sequence Models Using the SPIN Model Checker	**-3**(4),**-1**(3),**1**(2)	-1.0	reject?
43	Hierarchical Algorithms for Preserving Privacy of Composite Services Compensation	**0**(2),**-2**(4),**-1**(2)	-1.0	reject?

Make it Easy for your Champion

- Select appropriate keywords
  + Why are you in the scope of the conference/journal/… ?

- Test the abstract
  + Start early with the abstract
  + Ask for early (external) feedback

- Visible claims
  + Abstract + intro + conclusion have have visible claim(s)
  + Ask early feedback to summarize what reviewers think the claim is

- Clear validation
  + Champion is then able to defend it against detractors

- Write to the Program Committee
  + Target a PC member
  + Have a clear picture of your champion

The Fish Model

(1) The problem
- who has the problem ?
- why is it a problem ?
- what is the (sketch of) the solution ?

(5) The conclusion
- quick summary of solution + empirical evidence
- implications of the solution
- future work for the community (long term)

(3) The solution
- detailed description (sufficient for replication)
- empirical evidence

(2) The problem context
- why is it a *difficult* problem ? (few bibliographical details)
- which aspect of the problem do you tackle ?
- how can you show that you solved the problem ? (criteria / units of analysis / …)

(4) The problem context revisited
[a.k.a. "Related Work"]
- I only solved one aspect of problem
- others have worked on it (many bibliographical details)
- future work (short term)

⇒ together we made progress

Role of "Related Work"

Related Work

Problem Statement (beginning of paper)

Other researchers define the research agenda

⇒ high entry barrier

(for experts only)

Problem Context (end of paper)

Other researchers do complimentary work

⇒ crisp problem statement

(difficult to write)

Target Audience

Target Audience

Experts in sub-domain (in-crowd)

= preaching to the quire

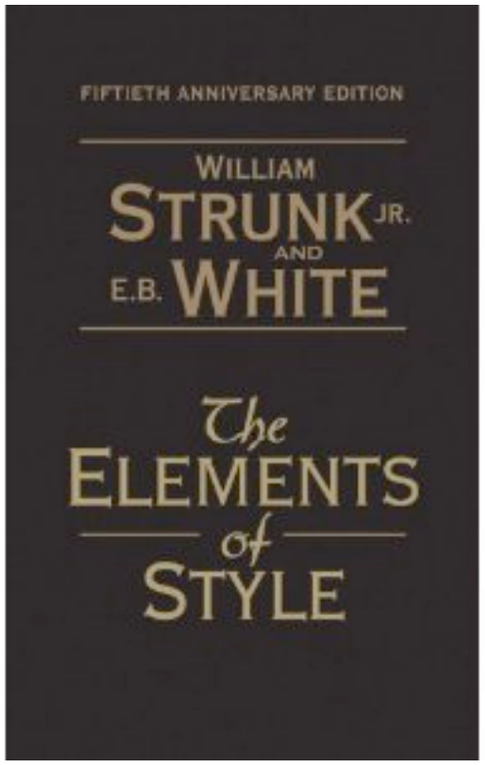
Broader Audience (informed outsider)

= arguing the problem and inviting others to contribute

- Conferences: ICSE, ESEC/FSE
- Journals: TSE, TOSEM
- magazines: IEEE Software, IEEE Computer, Communications of the ACM

Books on writing

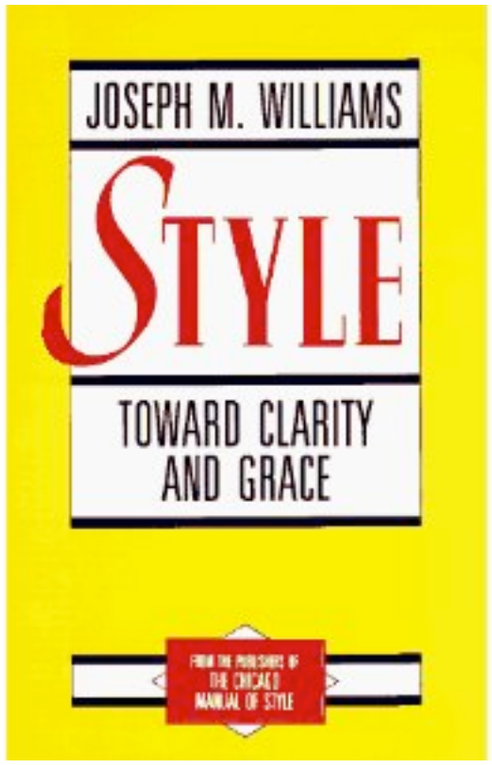
- *The Elements of Style*
  William Strunk Jr., E. B. White



- 18 simple guidelines
  + elementary rules of usage
  + elementary rules of composition

- You have to know the rules before you can break them

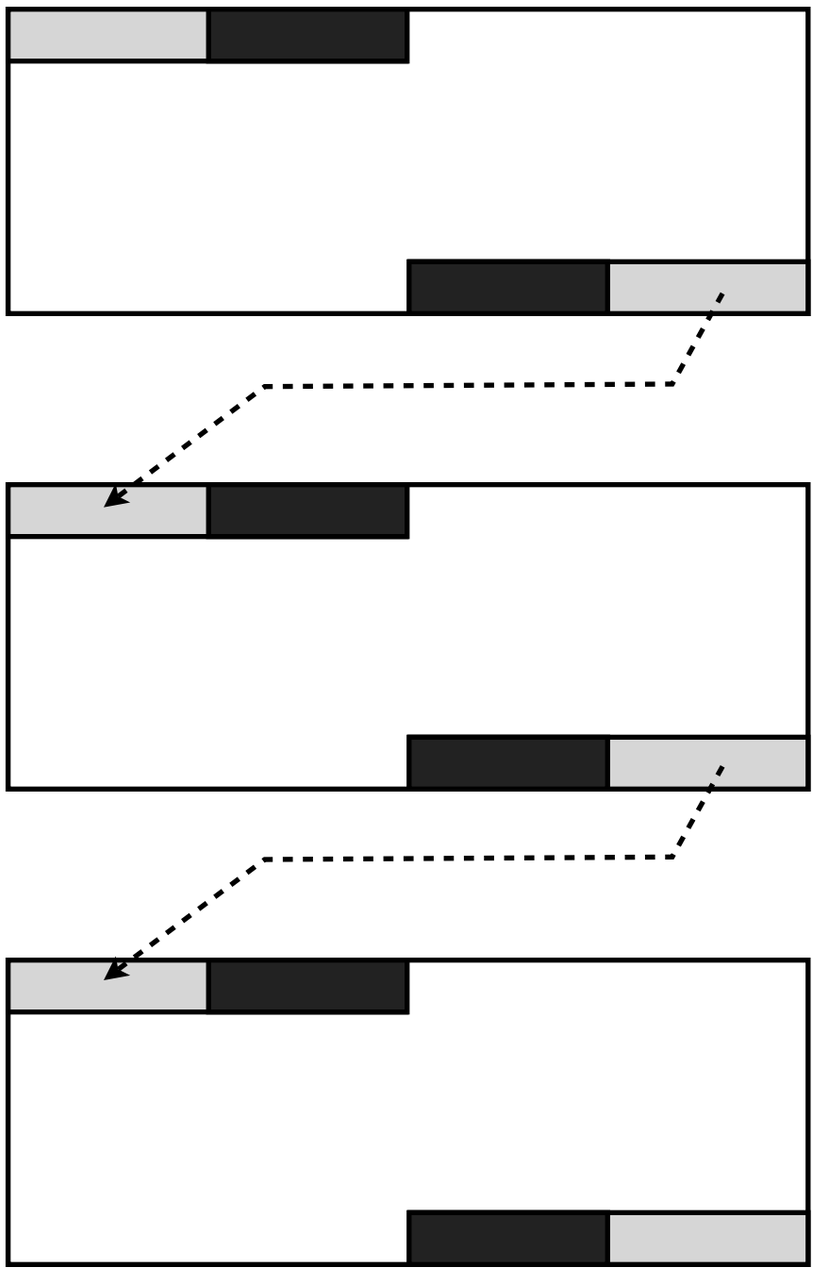
- *Style: Toward Clarity and Grace*
  Joseph M. Williams, Gregory G. Colomb



- guidelines
  + *refactoring* rules

- Give a man a fish and you feed him for a day. Teach a man to fish and you feed him for a lifetime.

Skimming texts — Emphasis



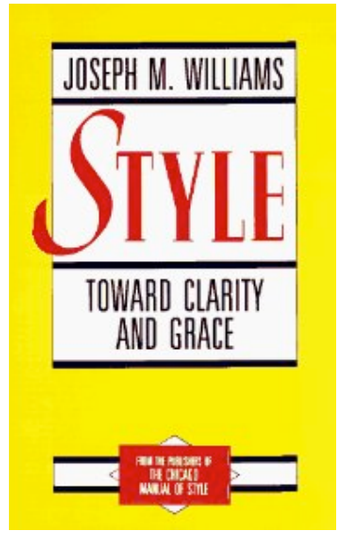
"natural" emphasis of paragraphs
- 1rst 1/2 of last sentence (most)
- 2nd 1/2 of first sentence

On section/chapter level
- say what you gonna say
- say it
- say what you have said

Source: Joseph M. Williams, "Style: Toward Clarity and Grace" The University of Chicago Press 1990

How to structure your writing

The last thing one discovers in writing a book is what to put first [Blaise Pascal]



all of us … must understand three things about complex writing:
- it may precisely reflect complex ideas
- it may gratuitously complicate complex ideas
- it may gratuitously complicate simple ideas

© Joseph M. Williams, "Style: Toward Clarity and Grace"

FIXED	Issue	Discussion
VARIABLE	Point	—

FIXED	Topic	Stress
VARIABLE	Old/Familiar	New/Unfamiliar

FIXED	Subject	Verb	Complement
VARIABLE	Characters	Action	—

Things to Avoid

- report order ≠ investigate order
  + arguments should appear in order that bests support the claim

- unsubstantiated claims, hopes, assumptions
  + XXX will make it easy/fast/better/integrate with other tools ...
    - do you actually demonstrate these claims in your paper ?
  + We believe ..., We hope ...
    - My favorite reviewing sentence:
      "We are doing science, not religion …"
  + XXX is valuable ..., XXX can help ...,
    XXX is an excellent ...
    - My favorite reviewing sentence:
      "Are these opinions? Hypotheses?
      Proven facts? Please add references."

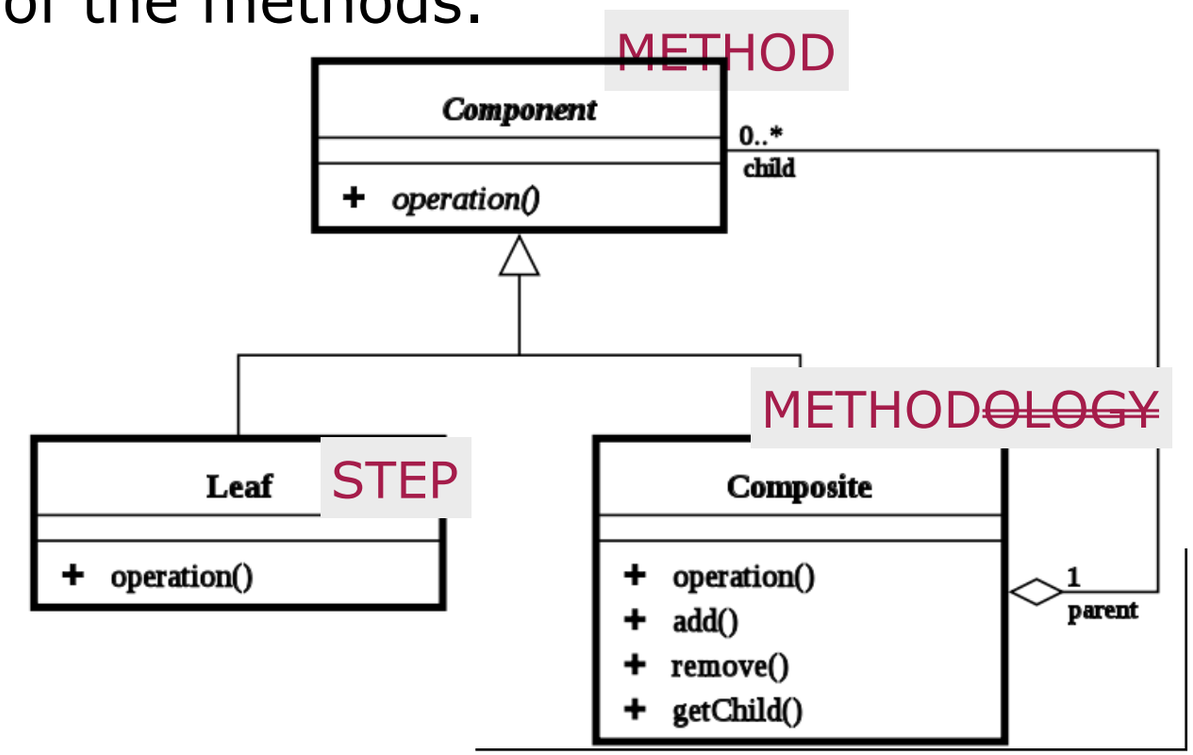
- tackling a non-problem, a problem which you cannot solve
  + A software engineering example
    - papers citing “Software Crisis”

Things to Avoid: Methodology

- "In this paper we propose a methodology for XXX"
  - + My favorite reviewing sentence:
    - Do not use the word "Methodology" for something simple like a technique, algorithm or even method; this is inflation of words

- the postfix -OLOGY
  - biology = the study of the living organisms
  - psychology = is the study of the human mind
  - cosmology = is the study of the cosmos
  - ➡ methodology = the study of the methods.

- method = a series of steps or acts taken to achieve a goal
  - + substeps of method remain a method
  - + cfr. Composite design pattern



The Task of a referee (1/2)

- source: Alan Jay Smith, "The Task of the Referee," Computer, vol. 23, no. 4, pp. 65-71, Apr. 1990

Decide

- Makes sufficient contribution ?
  + depends on the standards of the journal/conference/workshop/…

Questions to answer

- What is the purpose of this paper ?
- Is the paper appropriate? (for computer science / software engineering / reengineering / …)
- Is the goal significant ?
- Is the method of approach valid ?
- Is the actual execution of research correct ?
- Are the correct conclusions drawn from the results ?
- Is the presentation satisfactory ?
- What did you learn ?

The Task of a referee (2/2)

Categories

- (1) Major results; very significant (fewer than 1% of all papers).
- (2) Good, solid, interesting work; a definite contribution (≤ 10 %)
- (3) Minor, but positive, contribution to knowledge (perhaps 10-30 %).

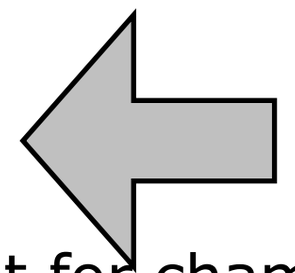
- (4) Elegant and technically correct but useless. This category includes sophisticated analyses of *flying pigs*.
- (5) Neither elegant nor useful, but not actually wrong.

- (6) Wrong and misleading.
- (7) So badly written that technical evaluation is impossible.

Reviewing Template

Review

- Strong accept / weak accept / weak reject / strong reject
  ➡ Including a solid motivation for your recommendation
- Template
  + summary (neutral)
  + strong points (bullet points)
  + points to improve (bullet points)
  + details
  + PC-only comments



Important for champion/detractor

Time estimation

1 paper = ± 4 hours

- 1,5 hour reading + annotating
  + read on paper
    ➡ submission for review incl. page numbers & white-space
- 1 hour writing review
- 1 hour discussion + adapting reviews
  + over mailing lists etc.
- 0,5 hour overhead
  + print papers (write numbers on them !!!)
    ➡ “first contact” with the papers

+ managing conference reviewing system
+ distribute among co-reviewers
+ …

Ph.d. students as Co-reviewer

- 2nd opinion (reduces time spent for “reading” and “writing review”)
- Ph.d. students experience “the other side of the fence”
- Mentioned in the proceedings (CV)

Once Accepted ...

... at the Conference

- prepare an elevator-pitch
  + based around “startling sentence” from your abstract

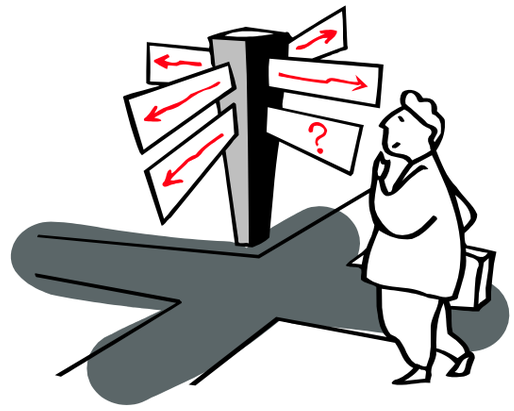
- approach gurus
  + they like it, it's good for their ego

- "explain your Ph.d. topic to at least 3 persons each day"
  + = advice from ICSM 2009 Ph.d. symposium

- submit to Ph.d. symposium
  + receive valuable feedback
  + network with future peers

- participate in workshops
  + test how the community reacts to research questions
  + the gurus struggle too !

Conclusion



Introduction
- The Publication Process
  + Publication Categories
  + Quality indicators

The Review Process
- Identify the Champion
- Implications for Authors
  + The 4-line abstract rule
  + The fish model
  + Natural emphasis of paragraphs
- Things to avoid
  + Method vs. Methodology

The Task of the referee
- Questions to answer ⇒ Review Template

Once Accepted …
- Tips and Tricks

Conclusion