Research Evaluation for Computer Science

An Informatics Europe report

Prepared by Bertrand Meyer

Draft, not for general circulation
Version 5.0, 7 February 2008

Executive summary

1. Computer science, also known as informatics, is an original discipline combining characteristics of science and engineering. Researcher evaluation must be adapted to the specifics of the discipline.

2. A distinctive feature of computer science publication is the importance of conferences, some of which are extremely selective, and books. Journal publication, while important for in-depth treatments of some topics, does not carry more prestige than top-quality conferences and books.

3. An important part of computer science research produces artefacts other than publications, in particular software systems. In measuring impact, these artefacts can be as important as publications.

4. Publication counts, weighted or not, must not be used as indicators of research value. They measure a form of productivity, but neither impact nor research quality.

5. Numerical impact measurements, such as citation counts, have their place but must never be used as the sole source of evaluation. Any use of these techniques must be subjected to the filter of human interpretation, in particular to avoid the many possible sources of errors. It must be complemented by peer review, and by attempts to measure impact of contributions other than publication.

6. Any evaluation criterion, especially if it yields a quantitative result, must be based on clear and published criteria.

7. Numerical indicators must not be used to compare research or researchers across different disciplines.

8. In assessing publications and citations, the ISI Web of Science is grossly inadequate for most fields of computer science and must not be used. Alternatives, imperfect but preferable, include Google Scholar and CiteSeer.

9. Evaluation criteria must themselves be subject to assessment and revision.

1. Research evaluation and its role

Research is a competitive endeavor. Researchers are accustomed to constant evaluation: any work submitted to a workshop, conference or journal — even, in some cases, an invited contribution — will be peer-reviewed; rejection is frequent, and even for a senior scientist remains a possibility for every new submission. Once a researcher's work has been accepted and published, it will be regularly assessed at all career stages against the work of other researchers. In addition to being evaluated, all researchers except for the most junior evaluate others: they act as referees, as participants in editorial board and program committees, as members of promotion and tenure committees, by responding to project evaluation requests from research-funding agencies, by writing letters of evaluation of colleagues as asked, often out of the blue, by various institutions.

The whole research management edifice relies on assessment of researchers by researchers. The criteria must be fair (at least as fair overall as can be expected of an activity circumscribed by human judgment); they must be clearly and publicly specified; and they must be globally accepted by the corresponding scientific community. This means in particular acceptance by the specific discipline involved: while other disciplines are often represented in evaluation processes, in particular for recruitment, it is not acceptable to impose criteria from one discipline on another, for example from an older, well-established science on a newer one that has developed its own distinctive principles.

In the case of computer science, a consensus has largely emerged in the US on both the peculiarities of the discipline and the properties it shares with others. This is in particular the result of the work of the Computing Research Association (CRA), which over the past three decades has represented the voice of academic computer science and established a fruitful relationship with other fields of research. An influential CRA report from 1999[1] defines, clearly and concisely, a set of "best practices" for the evaluation of computer scientists and engineers. The situation in Europe is less developed, as computer scientists have not so far made a concerted effort to explain the issues and principles to their colleagues from other fields.

It is one of the primary tasks of Informatics Europe, the association of computer science research and educational institutions in Europe, created in 2005[2], to make the requirements and specificities of computer science research and evaluation widely known.

The present Informatics report builds on the CRA's work; while it highlights only a few European specificities such as language diversity — for the simple reason that there are hardly any others to pinpoint, the criteria for research quality being the same anywhere in the world — it expands on some of the CRA report's points, and takes into consideration a number of developments that have happened since 1999.

---

[1] Computing Research Association: *Best Practices Memo — Evaluating Computer Scientists and Engineers for Promotion and Tenure*, prepared by a David Patterson, Lawrence Snyder and Jeffery Ullman; in *Computing Research News*, September 1999, available at www.cra.org/reports/tenure_review.html.

[2] www.informatics-europe.org

2. Computer science and its varieties

Computer science does not concern itself with computers but with *computing*: processing information using algorithmic techniques. The term *informatics*, popular in Europe, captures the essence of the discipline, emphasizing that it covers the whole span of information technology. This report assumes this broad sense of "computer science".

Core computer science research divides itself into two broad categories: **Theory** and **Systems**. The division is not absolute, as much research work on either side involves elements from the other, but is convenient as a broad characterization:

- *Theory* research concerns itself with conceptual frameworks for understanding computations, algorithms, data structures and other aspects of computing. It can itself be divided into three rough subcategories: *algorithms, complexity and combinatorics* (mathematical models for understanding machines and computations); *semantics, specification and proofs* (mathematical models of programming and programming languages, in particular to ensure correct functioning); *computational science* (mathematical models for high-performance computations). All three variants make extensive use of mathematics, although the mathematics relevant for the first two cases is from domains not central to traditional scientific education: logic, formal languages, automata theory.
- *Systems* research is devoted to producing artefacts and assessing their properties. The artefacts may be programs, but also systems that involve software along with other elements, as in "embedded systems" (cell phones, trains, air traffic control…) which include both software and hardware, and in "management information systems" which include both software and organizational processes. The main subdivision here is between *building* systems — research prototypes, but also software that is stable enough to be actually used for production — and *measuring* the properties of existing systems and processes. The latter is known as *experimental computer science* and draws some of its techniques from the natural sciences and statistics (for "performance analysis", which studies for example the throughput of networks). These two variants are often intertwined, since researchers who build a system will also analyze their properties. Classified along its areas of application, Systems research includes such specialties as: software engineering, which studies the best ways of building and maintaining high-quality software systems, in particular large and complex ones; programming languages and their implementation (compilers, interpreters); human-computer interface and graphics; database research, which tackles the issue of managing rich repositories of information; networking and operating systems; security, which addresses the issue of maintaining integrity and privacy of information; and others.

As suggested by the term "core computer science", these two categories cover research whose value extends beyond any particular application area of computing. A full picture of CS research should include a third category, often closely related to computational science: **applications**, devoted to the specifics of computing for a particular discipline, for example computational chemistry or computational finance. This area is sometimes called "*Computational X*". Evaluating such research requires combining criteria specific to computer science and criteria from the application area — the *X*. For this reason the present report limits its scope to core computer science, leaving it to the reader to determine how to perform this combination for a particular *X*.

Theory research brings computer science close to mathematics and to *sciences* such as physics. Systems research shares many properties with *engineering* research in fields such as electrical and mechanical engineering. This duality is part of the attractiveness of computer science, but also makes research evaluation more delicate, at it requires the proper mix between criteria generally applied to pure science and those appropriate to engineering research.

Whether Theory- or Systems-oriented, computer science research exhibits a strong set of distinctive characteristics, which have evolved over the half-century of its existence. (While it is possible to find precursor work all the way back to the ancient Greeks, Arabs and Indians, not to forget mathematical pioneers such as Pascal, the field as such took off with the appearance of computers after World War II, and the first CS departments in universities were created in the nineteen-sixties.) Embodying the discipline's spirit are a number of seminal concepts and paradigms, such as the notion of algorithm, computability, invariants, the distinction between specification and implementation (information hiding), recursion and fixpoints, the issues of scaling up, the role of notation, translation between languages, the duality between function and data, the notion of algorithmic complexity, refinement, and a number of others. This common conceptual corpus is essential for computer scientists; experts from other disciplines sometimes do not realize it even exists, viewing computers as essentially a tool and computing as a supporting task rather than a distinct scientific domain. This view is all the less justified that many of the seminal developments that have overhauled the practice of science (from computational methods to advanced text processing to the Internet to massive search mechanisms on the Web) are the direct consequence of core computer science concepts. It is the responsibility of computer scientists to educate their colleagues from other fields about both the scientific basis that computer science shares with these fields and the specifics of computer science research.

This overview leads to this report's first recommendation:

> 1. Computer science, also known as informatics, is an original discipline combining characteristics of science and engineering. Researcher evaluation must be adapted to the specifics of the discipline.

3. The computer science research culture

Whether from the nature of the field or from circumstances, computer science research has developed some distinctive properties.

Particularly foreign to the culture of older scientific disciplines is the role of conferences. In many other fields the prestigious publication venues are journals, conferences being just the opportunity to present raw initial results. Not so in computer science research, especially its Systems side. Some of the most prestigious publications are highly selective conferences, for example POPL, PLDI, OOPSLA, ECOOP, ICSE (in the software engineering and programming areas), SIGGRAPH and Eurographics (in graphics and HCI), Usenix (operating systems) and others. These conferences are highly selective, with acceptance rates hovering between 10 and 20%, for example[3]:

- ICSE (International Conference on Software Engineering): 13%.
- OOPSLA (Object-Oriented Programming): 19%.
- POPL (Principles Of Programming Languages): 18%.

Archival journals in the corresponding fields have their role, but often as a way to publish more in-depth versions of papers already presented at conferences, with details that could not be included because of the more stringent page limitations. While many researchers take the time to publish such longer versions, there are also examples of excellent researchers with mostly conference papers. This can cause problems in multi-disciplinary research evaluations, where colleagues from other fields consider journal publication as the basic yardstick of recognition. Our second recommendation helps avoid this common misunderstanding:

A related issue is the value of *books*. In many disciplines books are not considered important scientific contributions. In computer science (as in some fields in the humanities), books can be the primary form of impact. Almost any computer scientist, when asked about the most influential publication ever, will cite Knuth's *The Art of Computer Programming*, a compendium of books. In the evolution of software engineering, seminal concepts (such as "Design Patterns") became known through books before other modes of publication.

In increasing order of prestige, the typical process for publishing a new idea involves some of the following steps. The author may first publish an internal *report* of his or her institution, to establish precedence. Next he might submit a paper to a *workshop* to test the waters; workshops typically have the advantage of a fast review process, and are often affixed to a conference on a more general topic, allowing joint attendance. The next step is a *conference* submission; here the reputation of the conference, well understood by members of the discipline, will determine the prestige of the publication. In many cases the process will stop here, but the author may produce a longer *journal* version. Or he may decide to write a *book* giving a full exposition of the ideas.

Any evaluation process for computer scientists must be compatible with the discipline's publication culture:

---

[3] Averaged in each case over last three conferences, 2005-2007; for full details and many more conferences see ase.csc.ncsu.edu/semap/

2. A distinctive feature of computer science publication is the importance of conferences, some of which are extremely selective, and of books. Journal publication, while important for in-depth treatments of some topics, does not carry more prestige than top-quality conferences and books.

Even if correctly assessed, however, publications are not necessarily the sole form of scientific contribution (as they may be in some other disciplines). For researchers whose work involves a System component, the best way to demonstrate value is often to produce a program or other artefact that attracts the attention of their peers and of the rest of the world. This may be a more irrefutable demonstration of impact than a dozen papers. As an example, the Google success story is, at its heart, based on a fixpoint algorithm (building on one of the seminal concepts listed above): "Page Rank", which determines the popularity of a Web page based on the number of links to it, globally computed through an iterative algorithm. Before Google was commercial it was a research success story. One of the outcomes of the research was a paper on Page Rank[4]; another was the Google site and software. The paper described an ingenious algorithm, one of many that get published all the time; but the site had — beyond its future commercial value — a *research* value that the paper could not convey: the demonstration, for millions of users, of the scalability (another one of the above concepts) of the approach. Had Messrs. Brin and Page continued as researchers and come up for evaluation, the software would have been just as important as the paper.

It should be noted that assessing the value of such contributions can be delicate here, since one must also judge scientific merit: the mere report that a program has been downloaded a million times over the Internet is not by itself a proof of its conceptual contribution. Conference and journal publication, with its well-defined peer review process, provides more easily decodable and hence more reassuring evaluation grids. In assessing Systems work, however, it is improper to focus on publications alone and ignore contributions in kind:

3. An important part of computer science research produces artefacts other than publications, in particular software systems. In measuring impact, these artefacts can be as important as publications.

Another component of researcher evaluation is the often difficult issue of determining an individual's contribution to a collective work. Various disciplines have different practices; mathematical articles often have a small number of authors, whereas work in experimental disciplines commonly includes many participants who all want to share some of the credit. The practice in computer science is quite different. Not surprisingly, the number of coauthors per article tends to be lower for Theory-oriented work and higher for Systems-oriented work, but in either case they remain quite low as compared to the standards of the natural sciences, and higher than in mathematics. Compare:

---

[4] Sergey Brin and Lawrence Page: *The anatomy of a large-scale hypertextual Web search engine*, in *Computer Networks and ISDN Systems*, 33: 107-17, 1998. Also at infolab.stanford.edu/pub/papers/google.pdf.

- *Nature* over a year[5]: the maximum number of coauthors per article was 22 and the average 7.3.
- *American Mathematical Monthly*[6]: maximum 6, average 2.
- For both OOSPLA and POPL in 2007 (representative of high-quality software conferences): maximum 7, average 2.7.

In disciplines where numerous coauthors are the norm, researchers have developed elaborate author ordering conventions to suggest the extent of individual contributions. No such culture exists in computer science, although a growing number of researchers adopt the practice, in their publication lists, of underlining their names for articles to which they were the primary contributor. Such claims are hard to check. Apart from easily recognizable standard cases, such as a paper on a PhD student's thesis topic cosigned with the student's supervisor, it is not always easy to adjudicate contribution; there is a certain contradiction between the standard encouragement to embrace a collaborative research style and the possible dilution of one's contribution at the time of individual assessment.

4. Bibliometry

Under increasing pressure from political authorities and the public to show results, university administrations worldwide are drawn to the use of quantifiable results — sometimes known as *bibliometry* — including basic and derived measures.

Basic measures include:

- Raw publication counts.
- Publication counts weighted by publication value (determined through some official ranking of the prestige of each kind of publication, for example specific journals and conferences).
- Citation counts, measuring not output but impact, estimated from the number of other works citing a given publication.

Derived measures such as the h-index (basic, normalized) and g-index are computed from formulae involving citation counts, as detailed below.

The very idea of using such indicators for researcher evaluation has triggered some negative reactions, such as an article[7] by David Parnas, a famous computer scientist, and a collective letter[8] of 93 Swiss computer science professors.

[5] Issues from 6 December 2007 to 29 January 2008. From http://www.nature.com/nature/archive/.
[6] All 2007 issues, articles only (excludes notes and correspondence). From www.maa.org/pubs/monthly.html.
[7] David Parnas: *Stop the Numbers Game — Counting papers slows the rate of scientific progress*, in *Comm. of the ACM*, vol. 50, no. 11, November 2007, pages 19-21, available at http://tinyurl.com/2z652a. Parnas mostly discusses counting publications, but deals briefly with citation counts.
[8] Letter to CRUS regarding Bibliometric Evaluation, at http://www.sarit.ch/crusletter/.

In spite of these attitudes it is unrealistic to discard the idea of numeric criteria entirely; even if one ignores the political context (the need for universities to show tangible results, and the attraction of numbers), it is not clear that the suggested alternatives are always better. It is more productive to examine the objections, devise appropriate safeguards to avoid misuse or misinterpretation, and define a proper place for numeric criteria as part of a battery of evaluation techniques. This is the approach we recommend.

The first observation is that what is worth measuring is not volume but impact[9]. Publication counts, raw or weighted, are not relevant to assessment except as an indicator of overall activity and energy. Giving them any other role encourages publication inflation (a serious problem in research, the publication glut making it harder to spot interesting contributions), "write-only" journals with authors and no readers, conference sessions attended only by the speakers, and Stakhanovist research profiles based on quantity rather than quality.

> 4. Publication counts, weighted or not, must not be used as indicators of research value. They measure a form of productivity, but neither impact nor research quality.

Some publication counts take into consideration the value of the publication venue, according to some ranking of publications. This variant has the advantage of accounting for the peer recognition that follows from acceptance into a prestigious venue. The problem, however, is in assessing this prestige: defining the ranking. Most of the issues of individual researcher assessment apply to publication assessment as well. Many research agencies have predetermined rankings, which cause disputes and frustration and are particularly delicate to maintain in a fast-evolving field such as computer science. Rankings are often given a mantel of respectability through *publication impact factors*, which for being automatically computed are not necessarily more believable. Even the editor of *Nature* — one of the publications most favored by impact factors — has come out forcefully[10] against the concept. The only way to obtain meaningful publication rankings would be a process involving experts of a discipline (not a country!) worldwide. In the absence of such community consensus, weighing publication counts by journal rankings does not make them substantially better as an assessment criterion.

Citation counts are more relevant since they do assess impact. They are made possible by citation databases such as: the ISI Web of Science, which (as detailed below) is inadequate for computer science; CiteSeer, which attracted considerable attention when it was launched but seems no longer to be actively maintained; Google Scholar, probably the most usable resource but based on proprietary criteria.

Citation counts too are subject to serious criticism. Probably the prime reason for the negative reactions cited above is the inconsiderate attempt to use the ISI Web of Science

---

[9] Beyond impact, the factor of real interest is research *quality*. A measure of quality, however, can only result from human judgment. Impact is one of the inputs to that judgment, and is more amenable to approximation through measurement.

[10] Philip Campbell: *Escape from the impact factor*, in Ethics in Science and Environment Politics, Vol. 8, January 2008, available at http://www.int-res.com/articles/esep2008/8/e008pp1.pdf

on computer scientists, as discussed in the next section. Even with more reasonable databases, however, concerns remain:

- Focus. What should be evaluated is research quality. Publication quality is just one aspect of quality; impact is just one aspect of publication quality; citations are just one aspect of impact.
- Identity. Some researchers' names are frequently misspelled in the record, leading to citations being lost. Conversely, contributions from different authors are lumped together, especially since it is common practice to retain only the first letter of first names, so that "J Smith", "J Schmidt" or "J Dupont" will have high counts. First and last names of Chinese or Hungarian authors are often switched. All databases— shockingly —have trouble dealing with diacritical marks (accents, umlauts), so that Professor Fröhr's publications may be allocated between authors Fröhr, Froehr, Frohr, and (writing jointly) Mr. Fr and Mr. Hr.
- Other errors. An INRIA report on bibliometry[11] cites nine different renderings of the affiliation of just four INRIA authors[12]. Another example is ETH Zurich, whose members come up under many different affiliations, some not making the difference with EPFL in Lausanne (as both are sometimes identified as "Swiss Federal Institute of Technology"). This requires particular caution in using the citation databases to compare institutions — as when ETH and EPFL compete for funds — even though some highly publicized surveys such as the "Shanghai ranking" do so.
- Language. A frequent complaint about existing databases is their bias towards documents in English. Even in Europe this may be less of an issue in computer science than in other fields such as mathematics and the humanities.
- Distortions. Surveys tend to be widely quoted by article introductions, to refer the reader to an overview of the general topic. Rather than the article that introduced a breakthrough concept, in a way that after the fact may be seen as incomplete and perhaps hard to read, followers may cite a posterior work that improved the presentation. The article that introduced NP-completeness, a fundamental concept of computer science, is cited far less than a later more pedagogical presentation[13].
- Misinterpretation. Citation does not always imply positive recognition; scientists may cite a paper to criticize it or point out an error. A famous paper[14] describing a protocol contained an error, and has been cited by many publications on program verification to show that specific verification tools are able to detect the error.
- Time effects. Citation counts are most meaningful for older contributions (and, as a consequence, established researchers). Newer work and newer authors may not

[11] A-M. Kermarrec, E. Faou, J-P. Merlet (editor), P. Robert and L. Segoufin, for the INRIA Validation Committee: *Analysis document – What do bibliometric indicators measure?*, 12 September 2007. See also an interview of Jean-Pierre Merlet (in French) at www.inria.fr/actualites/inedit/inedit59_actu.fr.html.

[12] Including INRIA, INRIA ROCQUENCOURT, INFIA ROCQUENCOURT, INST NATL RECH INFORMAT & AUTOMAT ROCQUEN COURT, NAT RES INST COMP SCI & CONTROL.

[13] Dror G. Feitelson and Uri Yovel mention this example and many others from the CiteSeer database in *Predictive Ranking of Computer Scientists Using CiteSeer data*, in *Journal of documentation*, 60(1), 2004, pages 44–61 available at www.cs.huji.ac.il/~feit/papers/CitePred04JDoc.pdf.

[14] Roger Needham and Michael Schroeder: *Using encryption for authentication in large networks of computers*, in *Communications of the ACM*, 21(12), December 1978.

have had the opportunity to get cited yet[15]. Reinforcing this concern is the observation that the value of some important work takes time to be recognized.

- Size effects. Citation counts are absolute numbers, but impact within a given scientific community is relative to the size of that community. A seminal publication in a specialized area will get cited less than an incremental contribution in a fashionable field with high publication activity.
- Networking effects. Even without any consciously unethical behavior, groups of authors with kindred interests tend to form Mutual Citation Societies.
- Political bias. Some authors hope (through a practice that is ethically questionable) to maximize their chances of submission acceptance by lavishly citing works of program committee members.

The last two observations lead to another common criticism of citation counts: the possibly perverse effects on research. According to this criticism, researchers who are evaluated through a specific quantitative criterion will adapt their activity to maximize its value, in this case citations, at the possible expense of research quality.

A number of publications analyze in further detail the flaws of available citation databases; even a quick look at Friedemann Mattern's work[16] suffices to temper any temptation to trust automatically collected measurements blindly.

The overall lesson is that the quality of an evaluation process based on data cannot be better than the quality of the data. Unfortunately the problems have not been alleviated since the citation sources came into being, and no organization appears to be working on a solution. One notable exception is the DBLP site at the University of Trier, maintained by Michael Ley[17], which unlike the other sources cited makes it easy to contact a human to correct an error; but this site lists publications, not citations.

It is important to be aware of the risks associated with numerical indicators. No alternative, however, is perfect. Many computer scientists cite peer review as their favorite evaluation method; but peer review is not without its own issues:

- The results are highly dependent on the choice of evaluators and their availability (the most competent evaluators are often the busiest).
- If peer review were to become the sole evaluation mechanism, researchers would spend most of their time evaluating others rather than doing research.
- Most fundamentally, peer review has the limitations of any process based on human judgment. In particular, since one can in most cases solicit only a small number of reviews, the result is highly dependent on the choice of reviewers.

The solution appears to be in a combination of peer review and objective indicators. These indicators, as discussed, should not be restricted to publications; they should be

---

[15] Models have been proposed to replace the actual citation count by a predictive value for an author. See Feitelson and Yovel, note 13.

[16] *Bibliometric Evaluation of Computer Science — Problems and Pitfalls* (slide presentation); available at www.vs.inf.ethz.ch/publ/slides/Mattern-Bibliometry-SARIT06.pdf.

[17] www.informatik.uni-trier.de/~ley/db/.

assessed for relevance and reliability; and they should always be used through human interpretation, subject to critical analysis.

> 5. Numerical impact measurements, such as citation counts, have their place but must never be used as the sole source of evaluation. Any use of these techniques must be subjected to the filter of human interpretation, in particular to avoid the many possible sources of errors. It must be complemented by peer review, and by attempts to measure impact of contributions other than publication.

Critical analysis and assessment of the indicators assumes that the method for collecting the data is transparent:

> 6. Any evaluation criterion, especially if it yields a quantitative result, must be based on clear and published criteria.

It should be noted, however, that in the present state of bibliometry databases this last requirement remains wishful thinking. The method by which Google Scholar and ISI select documents and citations are not themselves published, or subject to public debate. Decision-makers must consider this limitation when using quantitative data in their assessments of individual researchers.

One important caveat on the use of any such indicators is the wide difference between disciplines. Patterns of publications vary considerably across areas of science and engineering; the earlier comment that it would be inappropriate to judge one according to the rules of another particularly applies here[18]:

> 7. Numerical indicators must not be used to compare research or researchers across different disciplines.

5. The ISI case

One issue of great concern to computer scientists is the common tendency to use as reference for publications and citations the database of Thomson Scientific's ISI Web of Science. This resource was devised for the natural sciences; while the corresponding community seems to be satisfied with its applicability there, it is grossly inadequate for computer science.

The major inadequacy of ISI comes from its arbitrary classification of what is or is not worthy of being counted. The selection criteria are arbitrary and opaque. Most conferences are not listed; books are generally not listed; conversely, some references are included indiscriminately.

[18] This is one of the recommendations of the INRIA document by Merlet et al (note 11).

The results make any computer scientist cringe[19]. For Niklaus Wirth, a famous computer scientist honored by the Turing Award (the highest honor in computer science) and known in particular for his design of Pascal, the ISI database lists a number of minor papers that happen to have been in indexed publications, but not his 1970 "Pascal User's Manual and Report" (with Kathleen Jensen), published as a book and one of the best known references in all of computer science. Ask any computer scientist what is *the* most influential publication in the field, and most will cite Donald E. Knuth's *The Art of Computer Programming* book series, which has acquired legend status; that reference does not figure in the ISI database. (On Google Scholar it gets over 15,000 citations, an astounding number.) Of the many articles that Knuth — also a Turing Award winner — has published, the three most frequently cited according to Google Scholar, each with about 1000 citations, do not even appear in the ISI records.

Evidence of how ISI breaks down for computer science is "internal coverage": the percentage of citations that cite a publication in the same database. Whereas ISI's internal coverage exceeds 80% for physics or chemistry, it is only 38% for computer science[20].

An extreme example of the arbitrariness of ISI criteria is Springer's *Lecture Notes in Computer Science* (LNCS), which ISI classifies as a journal, so that any inclusion in an LNCS volume is treated as a journal publication. In reality, LNCS is for all practical purposes a publishing house (a subdivision of Springer), which with great foresight identified, three decades ago, a lucrative market in fast publication of workshop and conference proceedings, plus some PhD theses and other monographs. Of the more than 5000 volumes so far many are excellent, some are mediocre; within good volumes, one finds good and less good contributions. Lumping all LNCS publications into a single journal category is absurd, especially since many excellent conferences not published by LNCS are not listed[21]. For example:

- The International Conference on Software Engineering (ICSE), the top conference in a field that has its own special ISI category, is considered a premier publication venue by anyone in the field; it is not indexed by ISI.

- Any software engineering workshop published in LNCS, the kind of venue where an author would typically try out an idea *before* it is ready for submission to ICSE, is indexed by ISI[22].

[19] All ISI results obtained from Web of Science searches on 4 February 2008, through entire database (options: Timespan=All Years. Databases=SCI-EXPANDED, SSCI, A&HCI, IC, CCR-EXPANDED).
[20] From Moed H.F., Visser M.S. *Developing Bibliometric Indicators of Research Performance in Computer Science: an Exploratory Study*, in ISSI 2005 (Proc. 10th Intl. conf of Intl. Soc. for Scientometrics and Infometrics), Karolinska Univ. Press, Stockholm, 2005, pages 275-279. Cited by Merlet et al (note 11).
[21] The LNCS publishers themselves are careful not to misrepresent their offering. The official LNCS site presents the series as: "*a medium for the publication of new developments in computer science and information technology research and teaching — quickly, informally, and at a high level*" (www.springer.com/computer/lncs?SGWID=0-164-6-73659-0).
[22] An example among hundreds: *Proceedings of the 9th International Symposium on System Configuration Management*, LNCS 1675, 1999. Addresses configuration management, a subtopic of software engineering. An ICSE paper on this topic, arising from a revision of a contribution to this workshop, would not be listed in the ISI Web of Science.

As another example, ISI indexes *SIGPLAN Notices*, a publication of the Programming Languages group of the ACM (one of the two major professional societies in computer science). SIGPLAN Notices is actually an *unrefereed* publication, used in its ordinary issues to publish drafts, notes, letters; but it devotes special issues to the proceedings of some of the most prestigious conferences such as POPL and PLDI. Unlike those to ICSE, contributions to these conferences will appear in ISI, but treated in the same way as an informal reader's note in a regular issue.

The database has little understanding of what constitutes computer science. The 50 most cited references in computer science according to ISI[23] include such entries as "*Chemometrics in food science*" (#13), from a journal called *Chemometrics and Intelligent Laboratory Systems*", a topic and a publication entirely alien to computer science. This is not just an isolated example; most of the entries on the list, even those which are related to computer science (usually from specialized fields rather than the core of the discipline) are not recognizable to a computer scientist as milestone contributions. The cruelest comparison is with the list of most cited computer science works on the CiteSeer site, devoted to computer science[24]; while imperfect like any such selection, the CiteSeer lists many articles and books familiar to all computer scientists. It has *not a single entry in common*[25] with the ISI list[26].

Merlet et al.[27] note that the top-ranked ISI journal is 195th on CiteSeer, and the top CiteSeer journal is 26th for ISI. While some might be tempted to use this as a reason to dismiss rankings altogether, examination of the differences shows that they simply reflect how far off ISI is from the general understanding of computer scientists.

The ISI list of "highly cited researchers" reflects the database's ignorance of computer science. Wirth, Parnas and Knuth, all iconic names in the field, do not appear. Of the ten Turing Award winners between 2000 and 2006, only one is listed (Ronald Rivest, the R of the RSA cryptographic algorithm), but not, for example, Adi Shamir (the S of RSA), another revered figure of theoretical computer science[28].

---

[23] ISI query including all "*COMPUTER SCIENCE*" topics except one: *TS=(COMPUTER SCIENCE, INFORMATION SYSTEMS OR COMPUTER SCIENCE, HARDWARE & ARCHITECTURE OR COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS OR COMPUTER SCIENCE, THEORY & METHODS OR COMPUTER SCIENCE, SOFTWARE ENGINEERING OR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE) Databases=SCI-EXPANDED, SSCI, A&HCI, IC, CCR-EXPANDED*. The omitted category is "*COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS*", to avoid a bias against core computer science; predictably, including it yields even more outlandish results.

[24] citeseer.ist.psu.edu/articles.html, consulted February 2008; data from August 2006.

[25] *One* author (Scott Kirkpatrick) appears on both lists, for different articles.

[26] [Note applicable to this draft: Goodrum, McGain, Lawrence and Giles, in *Scholarly Publishing in the Internet Age: A Citation Analysis of Computer Science Literature*, in *Information Processing Management* 37, pages 661-675, available at wotan.liu.edu/dois/data/Articles/juljuljiqy:2001:v:37:i:6:p:661-675.html, found (in 2000) a small but non-empty common subset. I am investigating this discrepancy, in particular whether the ISI queries used in the preparation of this article are the right ones.]

[27] See note 11.

[28] Of the just announced three 2007 winners, two are in the ISI list.

Although one might indeed expect ISI to give better results for Theory work, closer to mathematics (a long-established discipline), the difference is marginal because of the fundamental deficiencies in the process.

Since ISI indexing is based on an opaque process with no room for assessment or appeal of decisions, the situation is unlikely to improve.

Alternatives such as CiteSeer and Google Scholar are subject to criticism as well. While CiteSeer attempts to eliminate self-references, Google Scholar does not; neither project publishes its precise inclusion criteria[29]. These deficiencies, however, are negligible when viewed against those of the ISI Web of Science:

> 8. In assessing publications and citations, the ISI Web of Science is grossly inadequate for most fields of computer science and must not be used. Alternatives, imperfect but preferable, include Google Scholar and CiteSeer.

Anyone in charge of research assessment should be aware that attempts to use ISI for computer science will cause massive opposition. Announcements of such plans have led some computer scientists to reject all measurement-based techniques. This is an overreaction; but decision-makers and scientists from other disciplines must not try to impose on computer scientists a scheme that is demonstrably inapplicable to their field.

Beyond the specific deficiencies of ISI, all systematic studies of citation databases show wide variations between the results they yield, in particular for computer science. This has led some authors[30] to suggest systematic reliance on *several* databases, a commendable practice that seems hard to impose in practice. Perhaps software will appear that performs this automatically. In the meantime it is again essential to remember the limitations of data quality in today's databases, and avoid any career-affecting decision based on metric indicators whose validity has not be checked thoroughly.

6. Assessment formulae

A recent phenomenon, not addressed in earlier studies such as the CRA report, is the growing reliance on numerical measures of an author's impact, derived from citation databases. The most commonly cited formula is the **h-index**, defined as the highest n such that C $(n) \geq n$, where C (n) is the number of citations of a publication by the author. The justification for this formula seems to be that it correlates well with other measures of success such as Nobel prizes, although to our knowledge the supporting studies did not involve computer science research. Variants of the h-index include:

---

[29] For a general comparision between Google Scholar, ISI and other sources (including Elsevier's Scopus) see Anne-Wil Harzing, *Google Scholar — A New Data Source for Citation Analysis*, available at www.harzing.com/resources.htm#/pop_gs.htm. Anne-Wil Harzing is the author of the "Publish or Perish" tool discussed in the next section. A detailed study: is Lokman I. Meho and Kiduk Yang, *A New Era in Citation and Bibliometric Analyses:Web of Science, Scopus, and Google Scholar*, to appear in *J. of Amer. Soc. for Information Science and Technology*, available at arxiv.org/ftp/cs/papers/0612/0612132.pdf.

[30] Lokman E. Meho: The Rise and Rise of Citation Analysis, Physics World, January 2007; available at eprints.rclis.org/archive/00008340/01/PhysicsWorld.pdf.

- The individual h-index, obtaining from the h-index by dividing it by the number of authors, with the goal of better assessing individual contributions.

- The g-index, for which the value is the highest n such that that the top n publications received (together) at least n^2 citations.

The g-index corrects a significant deficiency of the h-index: that it does not recognize extremely influential individual publications. If your second most cited publication has 100 citations, it does not make a difference to the h-index whether your top publication has 101 citations or (as in Knuth's case above) 15000. The G-index corrects this.

The "Publish or Perish" site[31] makes it possible to compute these indexes for any scientist in a few seconds.

It would be as counter-productive to reject these techniques as it would be to use them blindly to yield a single magic number defining a researcher's value. In computer science as in other fields, there is no substitute for a careful evaluation process involving many complementary sources of information. Peer review is one of these sources; properly qualified and interpreted numerical measures can be another valuable one.

7. Assessing the assessment

Negative reactions to assessment formulae often elicit in return the reproach that the complainants are sore losers or refuse to go with the times. This is generally unfair. Any scientist, as noted at the beginning of this report, is accustomed to evaluation as a constant fact of life. What causes irritation is reliance on inadequate assessment methods. Scientists are taught to use rigor in their own work: to submit any hypothesis to scrutiny, any result to duplication, any theorem to independent proof. They naturally assume that assessment processes affecting their own careers will be subjected to high standards too. Just as they do not expect, in a discussion with a PhD student, to impose a scientifically flawed view on the sole basis of seniority, so will they not let university management impose a flawed evaluation mechanism on the sole basis of authority. The principle of collective self-assessment, which has been instrumental in the development of modern science, must continue to apply even as technology brings about new tools of evaluation.

The first step is to ensure, as noted above, that evaluation criteria are public. But they should also be justified on rational grounds, and subject to constant reassessment. This is particularly true of computer science because of the fast evolution of the discipline.

> 9. Evaluation criteria must themselves be subject to assessment and revision.

Openness and adaptability are the price to pay to ensure a successful process, wholeheartedly endorsed by the computer science community.

[31] www.harzing.com/resources.htm#/pop.htm.