Bridging the Gap

From Research to Practical Advice

Claire Le Goues, Carnegie Mellon University

Ciera Jaspan, Google

Ipek Ozkaya and Mary Shaw, Carnegie Mellon University

Kathryn T. Stolee, North Carolina State University

// Software developers need actionable guidance, but researchers rarely integrate diverse types of evidence in a way that indicates the recommendations' strength. A levels-of-evidence framework might allow researchers and practitioners to translate research results to a pragmatically useful form. //

SOFTWARE ENGINEERING (SE) seeks cost-effective solutions to practical problems by applying well-codified, ideally scientifically validated, knowledge. Scientific validation takes many forms, such as experiments, statistics, formal proofs, or some combination thereof; these provide differing degrees of confidence in the knowledge.

Emily, a software engineer at a mid-sized company, recently oversaw a software release with a critical user-facing bug. In the postmortem, someone asks Emily to evaluate static analysis for preventing this type of bug in the future. Web searches about static analysis reveal dozens of companies selling static-analysis tools. Their claims about identifying a wide variety of bugs are clearly marketing material. They boast of different technologies, at a range of prices. It's difficult to compare them, let alone decide whether they should be used at all.

Unfortunately, when engineers seek answers to their practical problems, "perfect" scientific knowledge is not always available. If it's not, engineers readily accept "good-enough" evidence: case studies, small-scale experiments, blog posts, or even advice from acknowledged experts.

Emily turns to an unbiased source: the research literature. Unfortunately, searching yields thousands of papers, each evaluating a different technique in a different way. Emily is certain the answer exists, but she's not enough of an expert to find it.

What happens if even codified knowledge is not available? Or if the results are unclear, contradictory, or fragmented and distributed in many places?

Exasperated, Emily consults a colleague who used static analysis previously at another company. He remembers that it never found enough important issues to justify the cost and recommends to just write more integration tests. In the end, Emily gives up on static analysis, unable to translate decades of research into actionable knowledge. She sighs wistfully,

 0740-7459/18/$33.00 © 2018 IEEE

RESEARCHERS AND PRACTITIONERS VOLLEY ABOUT MAKING RESEARCH USEFUL

Dear Practitioners:

The research community is actually discovering things you might find useful. Please help us organize this knowledge so that it's actually useful to you. Understand that this isn't absolute truth, but rather the best we can do at the moment. You must be thoughtful about using this knowledge, but it's a lot better than guessing.

Sincerely,
The Researchers

Dear Researchers,

We have a lot of questions, and we suspect you have answers. Unfortunately, the answers are scattered among thousands of papers, and we can't tell fact from fiction. Worse, there are entire topics that no one is studying because they aren't "scientific enough." We have fallen back on getting insights from *Hacker News*, *Stevey's Drunken Blog Rants*, and Jeff, who just transferred from Accounting. We're pretty sure Jeff doesn't know anything, but he's the loudest person in our stand-up, and we don't have any evidence to dispute him. We'll take whatever evidence you have.

Sincerely,
The Practitioners

imagining a dialogue between researchers and practitioners (see the sidebar).

This pattern is common: engineers often rely on their experience and a priori beliefs[1] or turn to coworkers for advice. This is better than guessing or giving up. But what if incompletely validated research outcomes could be distilled into reliable sources, intermediate between validated results and folk wisdom?

To impact practice, SE research results must lead to pragmatic, actionable advice. This involves synthesizing recommendations from results with different assumptions and levels of rigor, assigning appropriate levels of confidence to the recommendations. Here, we examine how these tradeoffs between rigor and pragmatism have been handled in medicine, where risk is often acceptable in the face of urgency. We propose an approach to describing SE research results with varied quality of evidence and synthesizing those results into codified knowledge for practitioners. This approach can both improve practice and increase the pace of research, especially in exploratory topics.

Software Engineering Research Expectations over Time

When the 1968 NATO Conference introduced "software engineering" to our vocabulary,[2] research often focused on designing and building programs. There were guidelines for writing programs; the concept of reasoning mathematically about a program had just been introduced. The emphasis was on demonstrated capability—what we might now call feasibility—rather than rigorous validation.

This is visible in a sampling of major results of the period. For example, Carnegie Mellon University identified a set of canonical papers published between 1968 and 2002.[3] Several are formal analyses or empirical studies, and a few are case studies. However, the majority are carefully reasoned essays that propose new approaches based on the authors' experience and insight.

The field has historically built on results with varying degrees of certainty. Indeed, Fred Brooks proposed a "certainty-shell" structure

for knowledge, to balance "the tension between narrow truths proved convincingly by statistically sound experiments, and broad 'truths,' generally applicable, but supported only by possible unrepresentative observations."[4]

Brooks' structure recognizes three nested classes of results: scientifically validated findings, observations, and rules of thumb—with different evaluation criteria for each. By properly identifying each result, we can take advantage of incomplete or partial knowledge. For example, Butler Lampson's "Hints for Computer System Design" is an excellent set of well-thought-out rules of thumb.[5]

Expectations for rigor in SE research have evolved since the NATO conference. Around the turn of the 21st century, the SE research community became concerned about the lack of quantitative, experimental research. A preponderance of papers accepted for the 2002 International Conference on Software Engineering (ICSE 02) defended their results with examples, followed by papers that supported results with formal or controlled experimental techniques and reports on experience in practice.[6] Beginning in 2004, the *evidence-based software engineering* (EBSE) community began calling for synthesizing research results through *systematic literature reviews* (SLRs).

The field has matured in its awareness of the variety of research methods available. In 2014 and 2015, ICSE asked authors to classify submissions as analytical, empirical, methodological, perspective, or technological, providing criteria for each category. Compared to ICSE 02, ICSE 16 had substantially more empirical reports and much more rigorous validation.[7] The strength of validation was the most important factor affecting acceptance, and there were clear alignments between the types of result and the validation techniques.

This evolution is consistent with the way ideas typically evolve in our field—building from key insights and early exploration to products, over several decades. Different research methods are appropriate at different stages of this evolution, as increasing confidence in the work justifies larger-scale, controlled evaluation. However, well-controlled experiments almost inevitably narrow the scope of their results and their immediate practical relevance.

Additionally, certain types of research (like design), and most early-stage research, remain more narrative; setting inappropriate evaluation expectations can deter progress. Even results falling short of current standards are "better than nothing" for practitioners. To help reconcile this tension between the research community's standards of rigor and the pragmatic needs of practitioners, we observe how *evidence-based medicine* (EBM) connects research results to the needs of clinical medical practice, and vice versa.

Evidence-Based Medicine

EBM is the conscientious, explicit, and judicious use of the current best evidence from medical clinicians to make timely decisions about the care of individual patients. EBM arranges medical evidence into a hierarchy (see Figure 1): case reports and series are toward the bottom, progressing to individual randomized clinical trials and then meta-analysis and systematic reviews.[8] In this, EBM emphasizes the synthesis of (possibly weaker) individual results into stronger conclusions. Significantly, EBM then supports practical decision making that traces the level of evidence and confidence in a decision to its source by assigning strengths to a recommendation based on the level of the evidence that supports it:

> *What are we to do when the irresistible force of the need to offer clinical advice meets with the immovable object of flawed evidence? All we can do is our best: give the advice, but alert the advisees to the flaws in the evidence on which it is based.*[9]

Table 1 gives the rules for assigning recommendation grades.[9]

Today, many diagnoses combine the context of the particular patient case with levels of uncertainty from the analysis model to determine diagnosis certainty and confidence in the recommended treatment. The decision can map to combined levels of certainty, including less certain results from different analysis methods. On the basis of EBM principles, physicians can now consult best medical-practice guidance via a smartphone from a patient's bedside.

EBM favors neither the research nor the practice. It instead represents a systematic approach to clinical problem solving that allows the integration of the best available research evidence with clinical expertise and patient values. EBM can teach SE the value of researchers and practitioners collaborating. Researchers make in-development techniques available for testing, and practitioners combine such evidence using their best judgment.

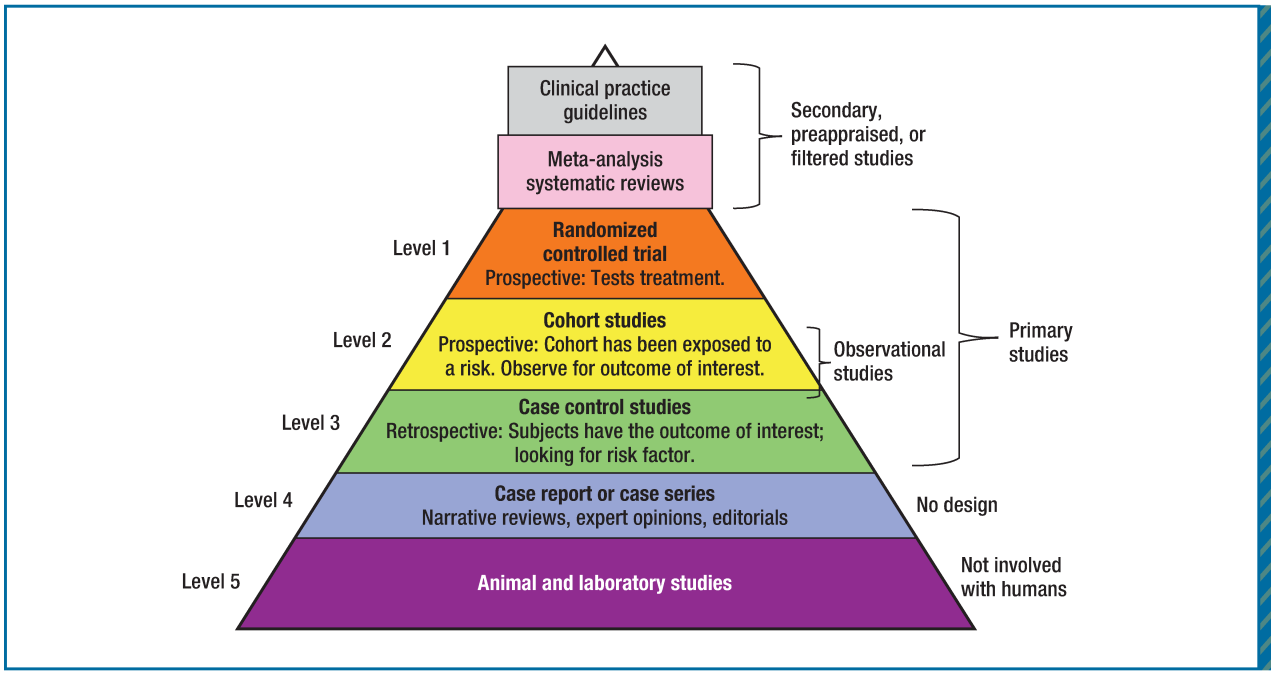

FIGURE 1. Levels of evidence for evidence-based medicine.[10]

Mapping Evidence to Recommendations

Can you use lower-quality evidence without filtering it through a systematic review? For SE, we say, "Yes, with caution." As Brooks observed, "Any data are better than none"[4]—provided the quality of that data is accurately described.

Table 2 provides a preliminary hierarchy mapping evidence to the level of rigor for SE research, drawing on earlier proposals for classifying research on the basis of the character of its evidence and validation.[4,6,11–14] As in EBM, this hierarchy acknowledges that evidence quality arises not only from the choice of research method but also from the design and rigor with which the research is conducted (e.g., the appropriateness of the subject and the sample size).

We identify eight levels. The highest levels of primary studies provide systematic, high-confidence evidence, including formal proofs or derivations (level 1) and quantitative results from empirical studies with good statistical control (level 2). Primary studies that are principally observational include results from sound qualitative methods (level 3), well-designed surveys (level 4), studies from multiple projects (level 5), and objective reviews on specific implementations or projects (level 6). This hierarchy also recognizes evidence that arises organically, without prior study design, such as anecdotes and rules of thumb as well as position papers and expert opinion (level 7). Systematic reviews with recommendations for practice (level 0) synthesize results from the other levels.

This classification accommodates distinctions made by prior researchers, such as these:

- Frederick Brooks. Findings, observations, and rules of thumb map to levels 1 and 2, 4 through 6, and 7, respectively.[4]

Table 1. Grades of recommendation.*[9]

Grade	Description
A	Consistent level 1 studies
B	Consistent level 2 or 3 studies or extrapolations from level 1 studies
C	Level 4 studies or extrapolations from level 2 or 3 studies
D	Level 5 evidence or troublingly inconsistent or inconclusive studies of any level

** The level numbers correlate with the levels in Figure 1.*

Table 2. The hierarchy of evidence for software engineering research.

Type of study		Level	Evidence
Secondary or filtered studies		0	*Systematic reviews* with recommendations for practice; meta-analyses
Primary studies	Systematic evidence	1	*Formal or analytic results* with rigorous derivation and proof
		2	*Quantitative empirical studies* with careful experimental design and good statistical control
	Observational evidence	3	*Observational results* supported by sound qualitative methods, including well-designed case studies
		4	*Surveys* with good sampling and good design; field studies; data mining
		5	*Experience from multiple projects*, with analysis and cross-project comparison; a tool, a prototype, a notation, a dataset, or another artifact (that has been certified as usable by others)
		6	*Experience from a single project*: an objective review of a specific project; lessons learned; a solution to a specific problem, tested and validated in the context of that problem; an in-depth experience report; a notation, a dataset, or an unvalidated artifact
No design		7	*Anecdotes on practice*; a rule of thumb; an evaluation with small or toy examples; a novel idea backed by strong argumentation; a position paper or an op-ed based principally on expert opinion

- Hakan Erdogmus. Systematic evidence, anecdotal evidence, and feasibility checks map to levels 1 and 2, 3, and 4 and 5, respectively.[11]
- Forrest Shull and his colleagues. Empirical methods map to levels 2 through 6.[14]
- Chris Scaffidi and Mary Shaw. Low-ceremony evidence maps to levels 6 and 7.[15]

EBM has previously inspired SE research practice. Barbara Kitchenham and her colleagues introduced EBSE, producing recommendations for practitioners and researchers.[12] EBSE recognizes that impact in both medicine and SE arises from the collection of multiple sources surrounding an idea, ideally synthesized through secondary studies. Key outcomes of this work are a set of recommendations for conducting SLRs and a call for increased empirical and controlled experiments in SE research.

Most SLRs in SE to date have addressed research questions rather than actionable advice. David Budgen and his colleagues found that only 37 out of 178 SLRs published between 2010 and 2015 provided recommendations or conclusions of relevance to education or practice.[16] A follow-up study found that the SLRs with recommendations on practice were derived predominantly from primary studies conducted in industry. While the focus on what has been done is helpful for researchers, the lack of focus on what has been learned is unhelpful for practitioners.

In one paper that provided actionable insights, the authors synthesized nine papers from the software-testing literature.[17] They gave rules of thumb for practice, such as "when you need higher assurance, ... a data-flow technique ... can be more effective than random testing."[17] What was missing was labeling the strength of the recommendations. The authors instead provided caveats about the initial studies, such as small evaluation programs (level 7) and the use of seeded faults (level 6). SLRs in SE have also not systematically addressed recommendation strength, which requires assessing whether the evidence is consistent and field-tested.

We advocate adopting the additional step of EBM, expecting SLRs to make explicit recommendations on practice, clearly labeled with strength of recommendation reflecting the level of rigor of the underlying evidence. Whereas EBSE emphasizes the quality of execution of individual studies as a basis for assigning confidence to SLR outcomes, we emphasize the level of evidence of the individual studies. Doing this recognizes the role of lower-confidence evidence in informing practice, both in medicine (where animal studies can be

informative) and in SE (where engineers value information from experience reports and overviews as well as empirical results).[13] We recognize the need for a set of rules similar to Table 1 for assigning strength to the recommendations in secondary studies, such as SLRs.

The strength-of-recommendation taxonomy from EBM offers a starting point. For example, in EBM, consistent evidence from the field has higher recommendation strength than expert opinions. Other factors that may influence decision making with respect to this hierarchy include the age of a result, application or evaluation context (e.g., using practitioners or students as subjects), and tooling availability. The research and practitioner communities should refine the hierarchy of evidence before designing a specific rule for strengths of recommendations in SE.

What Next?

We envision a system that allows researchers and practitioners to reliably synthesize research results into actionable, real-world guidance. Imagine an alternative universe for Emily:

> *Emily quickly identifies the latest reputable SLR on static analysis. The study provides specific recommendations for evaluating false-positive rates of commercial tools and integrating and customizing those tools. It also identifies a comparative case study (level 3) on a benchmark code base, showing which tools catch the errors of interest with few false positives. In addition, the study references several level 6 studies reporting experiences at other companies. By noon, Emily has selected a tool to try, modeled on the other studies.*

We are proposing not a new subfield of SE but rather a new way to label, organize, and synthesize results across SE to more concretely benefit practice. This vision is achievable, given sufficient community participation and cooperation. It requires the following:

- *Consensus on a formal framework for levels of evidence, together with a mapping between the evidence and the strength of the resulting recommendation.* SE researchers and practitioners should collaborate to refine the classification of research methods in Table 2. The refinement should establish guidelines for consistent application of those methods, supporting replication and meta-analysis. Rules for describing the level of confidence in a result should be formulated at all levels of this classification. They should reflect both the intrinsic power and quality of execution of each type of research, including the recommendation strength.
- *Explicit identification of methods and results*. This should allow the interested software engineer to easily identify where on the "pyramid" a contribution falls.
- *Incentives for and recognition of reviews that synthesize, interpret, or provide meta-analysis of bodies of prior work*. Such meta-analyses must have the goal of synthesizing actionable practical guidance. They should be labeled with the confidence and range of applicability.
- *Education of software engineers in how to use the framework*. SE students should be taught how to find appropriate studies and interpret their recommendations.

This sketch of a levels-of-evidence framework leaves open key questions for the SE community.

First, how can such a framework help researchers value research with different levels of evidence appropriately, and select research methods appropriate to their research questions? How should publication venues decide which types of research to include? How should they be labeled and differentiated?

Venues such as ICSE have previously required authors to label the type of their submissions, although this practice has not been consistent. We expect that top venues can and should continue to accept papers making use of "less rigorous" methodologies, assuming they are suitably identified. An important element of our argument is that such studies can importantly contribute to the body of knowledge, especially for emerging techniques. Opening the field to a wider variety of research results can increase the pace and novelty of the research we perform. It can also encourage the exploration of new directions even when controlled empirical data is difficult to collect.

In addition, how and when should meta-analyses be conducted and presented to software engineers? What incentives would persuade researchers and practitioners to perform these syntheses, and where should they be published and discussed? EBM relies on a central repository of SLRs, but this was established before Internet search largely replaced indexes and repositories as the preferred means of finding information. Going forward, it seems appropriate to publish SLRs in venues that match their subject matter and to make virtual collections as appropriate for comparison or comment.

ABOUT THE AUTHORS

CLAIRE LE GOUES is an assistant professor of computer science at Carnegie Mellon University's Institute for Software Research. Her research interests lie in automatically reasoning about and improving software quality in real-world, evolving systems. Le Goues received a PhD in computer science from the University of Virginia. Contact her at clegoues@cs.cmu.edu, @clegoues.

CIERA JASPAN is a senior software engineer at Google. She leads the Engineering Productivity Research team, which aims to identify inefficiencies in development tools and processes and improve the productivity of Google engineers and engineers who use Google products. Jaspan received her PhD in software engineering from Carnegie Mellon University. Contact her at ciera@google.com.

IPEK OZKAYA is a principal research scientist at Carnegie Mellon University's Software Engineering Institute. Her research includes the development and application of techniques for improving software architecture practices and practices to manage technical debt in large-scale, software-intensive systems. Ozkaya received a PhD in computational design from Carnegie Mellon University. Contact her at ozkaya@sei.cmu.edu, @ipekozkaya.

MARY SHAW is the Alan J. Perlis University Professor of Computer Science at Carnegie Mellon University's Institute for Software Research. Her research focuses on software engineering and software design, particularly software architecture and the design of systems used by real people. Shaw received a PhD in computer science from Carnegie Mellon University. She's a Life Fellow of ACM and IEEE. Contact her at mary.shaw@cs.cmu.edu.

KATHRYN T. STOLEE is an assistant professor in North Carolina State University's Department of Computer Science. Her research interests include program analysis, code search, and empirical software engineering. Stolee received a PhD in computer science from the University of Nebraska-Lincoln. Contact her at ktstolee@ncsu.edu.

As an engineering discipline, SE research should strive to impact practice. Favoring certain types of evidence over others will not suffice. Instead, we require a framework for aggregating the results of multiple pieces of work with different types of evidence into actionable practical feedback. In addition to encouraging technology transfer and true research impact, such a framework can simultaneously open our field to accepting a wider variety of research, including results that constitute the less rigorous (but still important!) codified knowledge that engineers use in practice.

What about this article? It's clearly opinion, and we tried to make it well-reasoned. It does draw heavily on practice, albeit from a different field. So this article is at level 7. That justifies further discussion and exploration. The next step should be community refinement of the hierarchy of evidence and the protocol for establishing the level of confidence.

Acknowledgments

We appreciate constructive comments from Fred Brooks and David Budgen. This work was supported by the Alan J. Perlis Chair of Computer Science at Carnegie Mellon University and by US National Science Foundation Structure and Hardware Foundation awards 1645136 and 1714699. Ipek Ozkaya's contribution is based on work funded and supported by the US Department of Defense under contract FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. DM18-0598

References

1. P. Devanbu, T. Zimmerman, and C. Bird, "Belief & Evidence in Empirical Software Engineering," *Proc. 38th Int'l Conf. Software Eng.* (ICSE 16), 2016, pp. 108–119.
2. P. Naur and B. Randell, eds., *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968*, NATO, 1968; http://homepages.cs.ncl.ac.uk/brian.randell/NATO/index.html.
3. M. Shaw et al., *Seminal Papers in Software Engineering: The Carnegie Mellon Canonical Collection*, tech. report CMU-ISR-15-107, Inst. for Software Research, Carnegie Mellon Univ., Sept. 2015.
4. F.P. Brooks, "Grasping Reality through Illusion—Interactive Graphics Serving Science," *Proc. 1988 SIGCHI Conf. Human Factors in Computing Systems* (CHI 88), 1988, pp. 1–11.
5. B.W. Lampson, "Hints for Computer System Design," *Proc. 9th ACM Symp. Operating Systems Principles* (SOSP 83), 1983, pp. 33–48.
6. M. Shaw, "Writing Good Software Engineering Research Papers," *Proc. 25th Int'l Conf. Software Eng.* (ICSE 03), 2003, pp. 726–736.
7. C. Theisen et al., "Software Engineering Research at the International Conference on Software Engineering in 2016," *SIGSOFT Software Eng. Notes*, vol. 42, no. 4, 2018, pp. 1–7.
8. Evidence-Based Medicine Working Group, "Evidence-Based Medicine: A New Approach to Teaching the Practice of Medicine," *J. Am. Medical Assoc.*, vol. 268, no. 17, 1992, pp. 2420–2425.
9. "Oxford Centre for Evidence-based Medicine—Levels of Evidence (March 2009)," Centre for Evidence-Based Medicine, Oxford Univ., Mar. 2009; https://www.cebm.net/2009/06/oxford-centre-evidence-based-medicine-levels-evidence-march-2009.
10. J. Forrest, *Evidence-Based Decision Making: Introduction and Formulating Good Clinical Questions*, Continuing Dental Education Course 311, dentalcare.com, 2014; https://www.dentalcare.com/en-us/professional-education/ce-courses/ce311.
11. H. Erdogmus, "How Important Is Evidence, Really?," *IEEE Software*, vol. 27, no. 3, 2010, pp. 2–5.
12. B.A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, Chapman & Hall, 2015.
13. M. Montesi and P. Lago, "Software Engineering Article Types: An Analysis of the Literature," *J. Systems and Software*, vol. 81, no. 10, 2008, pp. 1694–1714.
14. F. Shull, J. Singer, and D.I.K. Sjøberg, eds., *Guide to Advanced Empirical Software Engineering*, Springer, 2007.
15. C. Scaffidi and M. Shaw, "Toward a Calculus of Confidence," *Proc. 1st Int'l Workshop Economics of Software and Computation* (ESC 07), 2007, pp. 7–9.
16. D. Budgen et al., "Reporting Systematic Reviews: Some Lessons from a Tertiary Study," *Information and Software Technology*, vol. 95, 2018, pp. 62–74.
17. N. Juristo et al., "A Look at 25 Years of Data," *IEEE Software*, vol. 26, no. 1, 2009, pp. 15–17.