Evidence-Based Software Engineering: Secondary studies and Systematic Reviews

Expert Reviews

The ‘review paper’ is a well-established model that is used in many domains, with such papers often being published in a journal that only publishes reviews, and the papers being written by experts in the field who survey the current state of the literature on a topic and draw appropriate conclusions. In computing we have *ACM Computing Reviews* and in the IS area, *MIS Quarterly*. However, because these are ‘expert reviews’ it is quite possible that, for the same topic, two different experts might select different papers and draw separate conclusions!

The *Evidence-Based* paradigm originated in clinical medicine, spurred on by the epidemiologist Archie Cochrane’s concern about the quality and variability of research evidence being used to inform practice/teaching. *Evidence-Based Medicine* (EBM) employs *secondary studies* to find, judge and synthesise the outcomes of all relevant empirical studies in order to draw conclusions about particular treatments[1] in a repeatable and unbiased way (unlike an expert review).

Secondary studies

While a primary study is one that makes measurements of some effect (such as the use of some SE practice), a secondary study aggregates the outcomes of many primary studies (which may have different forms) in an *objective* and *unbiased* manner using either qualitative or quantitative forms of analysis. A secondary study is motivated by a protocol that identifies such elements as the research question, how it can be categorised in keywords, and how and where the search for source material will be conducted.

Secondary studies take different forms, according to their purpose and the type of research question being addressed. Two major ones used in SE are the *mapping study* used to survey the literature on a topic and categorise it; and the *systematic review*[2], which seeks to aggregate the findings of primary studies. Systematic reviews may address qualitative or qualitative questions, each requiring a particular form of aggregation process (synthesis).

Secondary studies are used in many domains, and while Clinical Medicine is a major user (helped by the use of *Randomised Controlled Trials* (RCTs) that permit the use of statistical meta-analysis for aggregating the findings of primary studies) other domains that are employing this approach include Education (where such reviews often inform policy); as well as other branches of healthcare such as Nursing & Midwifery; and various branches of the Social Sciences.

What’s available?

A short study by Budgen & Brereton (2022) looked at the profile of 131 secondary studies using a sample of those published in three years, spread across an 11 year period. The majority of secondary studies in SE were mapping studies, many of these examining research trends. Systematic reviews made up the bulk of the remaining studies, and these in turn were mainly qualitative. Unlike clinical medicine, where studies are mainly quantitative ones looking at the effects of interventions, studies in SE are likely to be producing qualitative findings that (say) identify those factors that assist with adopting particular practices, or barriers to doing so.

Secondary studies in Software Engineering: examples

A study conducted by Dybå and Dingsøyr (2008) looked at what was known about agile methods. They found a distinct lack of trustworthy primary studies, especially where management methods were concerned. Most studies were about XP, only one at that time addressed Scrum. Their figures are interesting:

- their search process returned 1996 papers (after removing duplicates)

[1] The *Cochrane Collaboration* (www.cochrane.org) was founded in 1993 to support this work.

[2] Earlier SE documents used the term ‘Systematic Literature Review’, while later ones have mostly adopted the more term ‘Systematic Review’ as used in other disciplines.

- exclusion based on title reduced this to 821
- after reading the abstracts, they had 270 left
- after reading the full papers, they ended up with 36 for the study

A second example is the study of Pair Programming by Hannay et al. (2009). As the primary studies were largely similar experiments they were able to conduct a meta-analysis of 18 studies, and found *limited* evidence for the following characteristics of pair programming:

- its use can lead to higher quality code (small significance)
- it can be faster than conventional (solo) programming (small significance)
- it can be less productive (moderate significance)

Evidence-Based Practice

As interpreted for use in Software Engineering the basic process is:

1. Convert need for information into an answerable questions
2. Find the best evidence with which to answer the question(s)
3. Critically appraise the evidence for its validity (closeness to the truth), its impact (size of the effect), and its applicability (usefulness)
4. Integrate the critical appraisal with SE expertise and with stakeholders' values
5. Evaluate the effectiveness and efficiency in steps 1-4 and seek ways to improve them

The first three of these steps are essentially the process of *systematic review*, while the fourth step is the process of *knowledge translation*, involving matching the outcomes of a review with practitioner expertise in order to provide guidelines for practice.

Conducting a Systematic Review

Phase 1: Plan Review consists of three steps:

- specify research question—a non-trivial task as used to construct search strings and may go through several stages of refinement, possibly helped by a wide scope mapping study
- develop review protocol—a key document needed to guide conduct of the review
- validate review protocol—among other things, doing so may need some form of 'dry run'

Phase 2: Conduct Review involves five activities:

- identify relevant research—consists of executing the search strategy defined in the protocol in terms of search strings, sources to search, bounding dates, etc., possibly leading to a large number of source documents
- select primary studies—sift through the candidate papers' titles, abstracts and (if necessary) bodies
- assess study quality—mainly concerned with assessing how a given primary study was conducted, and how well-structured the data reporting is
- extract required data—usually involves 'data extraction forms' to help structure this process, and ideally uses two analysts to ensure consistency
- synthesise the data—depends on the form, may be qualitative, or quantitative using statistical forms

Phase 3: Document Review is essential the writing and validating of the resulting report.

Experiences with conducting Systematic Reviews in Software Engineering

No standard abstracting services (unlike Medicine). So finding primary studies is mainly performed using search engines such as Scopus, Web of Science, ACM, etc. which search different sets of sources (mainly digital libraries), use different interfaces and do not always produce consistent outcomes. Even collectively they are unlikely to find all papers, and experience suggests that around 5-10% more papers (estimated) can still be found through snowballing.

Also, reporting standards are poor. Abstracts for primary studies tend to be written badly and often omit important information. Also, the papers are apt to report only that part of the data that is relevant to their own research question, with no sense of adding to the overall set of knowledge about a topic.