A Beginner's Guide to Evidence-Based Software Engineering and Systematic Reviews

1. Evidence-Based Software Engineering (EBSE)

Evidence-Based Software Engineering is the application of evidence-based principles, originally developed in medicine, to software engineering. Its goal is to support decisions about methods, tools, and practices using systematically collected and analyzed empirical evidence. Rather than relying only on intuition or isolated experiences, EBSE emphasizes integrating the best available research with professional judgment.

2. Purpose of EBSE

EBSE serves multiple purposes:

- **Summarizing knowledge**: Provides structured overviews of existing studies on a given topic.
- **Supporting decisions**: Helps practitioners and researchers choose appropriate technologies or methods.
- **Identifying gaps**: Reveals areas where evidence is missing or inconclusive, guiding future research.
- **Reducing risks**: Prevents the adoption of tools or practices that are ineffective or harmful.

3. EBSE in Research and Practice

In research, EBSE is used to frame new studies in the context of what is already known. For example, before investigating a new testing technique, researchers analyze prior studies systematically to identify which methods have been tested, in what context, and with what results. In practice, EBSE helps professionals adopt tools and processes that are empirically supported, minimizing risks of failure.

4. Why it Matters for Master's and PhD Students

For graduate students:

- **Undergraduate level**: A systematic review or mapping study exercise can train students to critically read research papers, compare results, and understand how

knowledge accumulates in software engineering. It strengthens analytical thinking and provides skills useful for capstone projects or research-oriented courses.

- **Master's level**: An SLR or mapping study can form the foundation of a thesis by providing a structured survey of existing work and helping identify relevant research questions.
- **PhD level**: SLRs and mapping studies can reveal deeper research gaps, justify novelty, and position contributions within the broader literature. They are also often publishable as stand-alone papers.

5. Types of Secondary Studies

- **Systematic Literature Review (SLR):** A structured, comprehensive synthesis of all relevant studies addressing a specific research question. It emphasizes rigor, replicability, and transparency.
- **Systematic Mapping Study (Scoping Review):** Focuses on categorizing and classifying existing research to give a broad overview of trends, volume, and distribution of studies. It emphasizes breadth over depth.
- **Rapid Review:** A streamlined version of an SLR. It sacrifices some comprehensiveness for speed, often used when timely evidence is needed for decision-making.

6. Process: From Protocol to Execution

A systematic review requires a **protocol**, a pre-defined plan that ensures transparency and replicability. The process can be structured into three phases: Planning, Conducting, and Reporting.

A. Planning the Review (Protocol Development)

1. **Define Research Questions (RQs):** Clear, focused, and answerable questions that guide the review.
2. **Provide Definitions:** Define specialized terms to avoid ambiguity.
3. **Craft Search String:** To search effectively, build structured search strings that can be applied in different databases. This involves three parts: (a) **Clarify Definitions**: define important terms in your research questions so their meaning is unambiguous; (b) **Identify Keywords**: list the main keywords and collect synonyms, related terms, and alternative spellings; (c) **Build Search Strings**: use Boolean operators (AND, OR, NOT) to combine the keywords into structured queries, adapting syntax for each database. For example: ("software testing" OR "unit testing" OR "integration testing") AND ("machine learning" OR "AI").
4. **Define Inclusion/Exclusion Criteria:** State clearly what types of studies will be considered and what will be excluded for the research (e.g., years, languages, topics).
5. **Plan for Quality Assessment:** Decide how to assess the rigor and reliability of each study identified.
6. **Design Data Extraction Forms:** Specify what data to extract (e.g., context, methods, results).

7. **Plan for Data Synthesis:** Decide whether synthesis will be qualitative (themes, concepts) or quantitative (meta-analysis).

B. Conducting the Review

1. **Run Searches:** Apply queries to each database and document results.
2. **Screen Studies:** Filter based on inclusion/exclusion criteria, first by title/abstract, then by full text.
3. **Assess Quality:** Apply checklists or coding schemes to evaluate study reliability.
4. **Extract Data:** Use predefined forms to collect consistent data across studies.
5. **Analyze Data:** Synthesize findings into coherent results (thematic summaries, frameworks, statistical synthesis).

C. Reporting the Review

1. **Transparency:** Provide a clear chain of evidence from search to synthesis.
2. **Flow:** Show how studies were included/excluded.
3. **Synthesis Presentation:** Use tables, figures, and conceptual frameworks.
4. **Conclusions:** Present evidence-based answers to RQs, recommendations for practice, and gaps for future research.

Key Takeaways for Students

- EBSE is about **using evidence systematically** in software engineering.
- SLRs, mapping studies, and rapid reviews are **secondary research methods** to synthesize prior work.
- Developing a **protocol** is essential for credibility, replicability, and transparency.
- The process is structured but adaptable: clarity, rigor, and transparency are the central principles.

Suggested Readings

1. Paul Ralph and Sebastian Baltes. 2022. Paving the Way for Mature Secondary Research: The Seven Types of Literature Review. *Proceedings of The ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022) Ideas, Visions and Reflections Track*, Singapore: ACM, Nov. 14–18.
2. Moher D, Liberati A, Tetzlaff J, Altman DG, The PRISMA Group (2009). *P*referred *R*eporting *I*tems for *S*ystematic Reviews and *M*eta-*A*nalyses: The PRISMA Statement. PLoS Med 6, 7: e1000097. doi:10.1371/journal.pmed1000097
3. Michael Borenstein and Larry V. Hedges and Julian P.T. Higgins and Hannah R. Rothstein. 2009. *Introduction to Meta-Analysis.* John Wiley & Sons Ltd.
4. Daniela S. Cruzes and Tore Dybå. 2010. Synthesizing evidence in software engineering research. In Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM ‘10).

Association for Computing Machinery, New York, NY, USA, Article 1, 1–10. DOI:10.1145/1852786.1852788

5. Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering.
6. Matthew B. Miles and A. Michael Huberman and Jonny Saldana. 2014. Qualitative Data Analysis: A Methods Sourcebook. Sage Publications Inc.
7. Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic mapping studies in software engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE).* (Jun. 2008), 1–10.