

Software Effort Estimation: a Review of Taxonomies for use to-wards a Probabilistic Evidence Synthesis

Emilia Mendes

eme@ece.au.dk

Aarhus University

Marc

Aarhus University

Systematic Review

Keywords: Systematic Literature Review, Taxonomy, Knowledge classification, Taxonomies for Effort Estimation, Software Engineering, Software Effort Estimation

Posted Date: October 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11319252/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Software Effort Estimation: a Review of Taxonomies for use towards a Probabilistic Evidence Synthesis

Emilia Mendes, Marc Solé Estragués

eme@ece.au.dk, msole@ece.au.dk

Department of Electrical & Computer Engineering

Aarhus University, Denmark

Abstract

Context: Taxonomies have been proposed for a range of knowledge areas in Software Engineering (SE), and they are seeing as important because they can serve as valuable tools for advancing the SE field's maturation. They present many advantages, such as the means to handle/reduce ambiguity, foster communication between different stakeholder groups, management of complex domains and identification of gaps.

Objective: To present a synthesis of evidence on the state of the art on taxonomies for Software Effort Estimation (SEE); such synthesis is driven by four main research questions that aim to present and compare the taxonomies for SEE identified, and to which software application domains they relate to.

Method: A Systematic Literature Review (SLR) was employed to identify and synthesise the primary studies describing taxonomies for SEE. This SLR was mainly driven via the SEGRESS SLR guidelines; and after detailed screening over two large databases, six taxonomy papers for SEE were included and used to answer four research questions.

Main results (SLR): The taxonomies varied significantly in structure and did not cater for any emerging software application domains. Some taxonomies included categories such as effort predictors, estimation techniques and contextual factors, whereas others included only one category (e.g. size measures, or contextual factors, or estimation techniques). A semantic similarity comparison revealed that only 36 categories (44.4%) and 62 characteristics (17%) are shared across taxonomies at a 90% semantic similarity threshold, with 303 characteristics being unique to individual taxonomies. Five of the six taxonomies lacked external validation.

Conclusions: The main aspects identified through the SLR are the absence of taxonomies for emerging application categories (e.g. mobile apps, artificial intelligence); fragmentation across categories and characteristics, even between taxonomies targeting at the same software application domains; and lack of external validation. Future work in this area is needed to either extend existing taxonomies for SEE, or to propose a single taxonomy that properly caters for all the software application domains, presents no/controlled fragmentation and is validated in industry.

Keywords: Systematic Literature Review, Taxonomy, Knowledge classification, Taxonomies for Effort Estimation, Software Engineering, Software Effort Estimation

1 Introduction & Motivation

A cornerstone of software project management is Software Effort Estimation (SEE), the process by which effort is predicted and used as basis to forecast costs and to allocate resources effectively, thus supporting software projects to be delivered on time and within budget [1]. SEE is a very complex domain because the relationship between software effort drivers, and effort itself, is non-deterministic and exhibits an inherently uncertain nature [1],[2].

Over the past 50 years, a large body of SEE research has been published and there have been several attempts via secondary studies (Systematic literature reviews, Mapping Studies) and tertiary studies to provide either syntheses of evidence regarding the entire SEE area [3],[4],[5],[6],[7], or evidence syntheses focusing upon specific perspectives pertaining to SEE, such as machine learning techniques used for prediction and Agile software development (please see [10] for details).

Despite such attempts, their data syntheses were characterized by scientometrics and research trends, with great differences between studies' research questions (in the case of SLRs) and syntheses of evidence. Furthermore, these SLRs overlooked important methodological and wide-ranging contextual aspects fundamental to better understand under what conditions and how effort predictions can be improved [11].

The existing body of SEE research is large, with hundreds of published primary studies. To consider carrying out a systematic literature review covering the past 50 years, and to present numerous very large and detailed summary tables will neither genuinely support the knowledge representation of the area nor provide a detailed understanding about under which conditions & scenarios accurate estimations are obtained. In addition, summary tables are static displays of evidence synthesis that include vote counting at best; they do not embed any probabilistic reasoning that could inform decision-making via, for example, 'what-if' scenarios [99].

In the medical field Bayesian Network (BN) (e.g. [96],[97],[98]) have been employed for evidence synthesis of SLRs (e.g. [96],[91],[92],[93]), and using data from meta-analysis combined with simulation (e.g. [97],[98]). A BN is a probabilistic technique that supports the modelling of knowledge as a directed acyclic graph, where nodes represent the variables of interest, edges their causal relationships, and with a conditional probability table attached to each node [100]. Nodes, edges and probabilities can be learnt from data, via expert-knowledge, or using a hybrid-approach. BNs have been distinguished for providing the means to inform and support decision-making in Software Engineering (e.g. [100],[94],[101]), and to synthesise/encode knowledge and data, identify knowledge/data gaps, exploratory and "what-if" scenario analysis, and to improve system understanding (e.g. [100]). Despite their wide use in SE, to date they have not yet been employed in SE for evidence synthesis. As part of a larger research project, and setting the motivation for carrying out this SLR on taxonomies for SEE, we aim to build a BN model for SEE probabilistic evidence synthesis [90], where the model's nodes are informed by a SEE Taxonomy.

Therefore, we needed to first identify the existing taxonomies for SEE so to assess whether any of them could be readily used in the project as is, or whether another taxonomy would need to be proposed and validated. Note that the decision as to whether we will use a single pair SEE taxonomy & BN model for SEE, or diverse pairs catering to different SEE contexts & software application domains is outside the context of this paper.

Figure 1 presents an overview on how our research will use a taxonomy for effort estimation to inform the building of a Bayesian network model for probabilistic synthesis of evidence in software effort estimation. Once a Taxonomy has been proposed and validated, then an LLM & prompt can be used in a multi-label text classification task, to identify the number of studies that have used each of the taxonomy's characteristics (leave nodes). With such information and additional information on accuracy results, we can build the node probability tables for parent nodes and to also use LLMs to impute probabilities for some of the child nodes. The detailing of this project and some of the formal experiments being currently conducted using LLMs are provided in [102],[103].

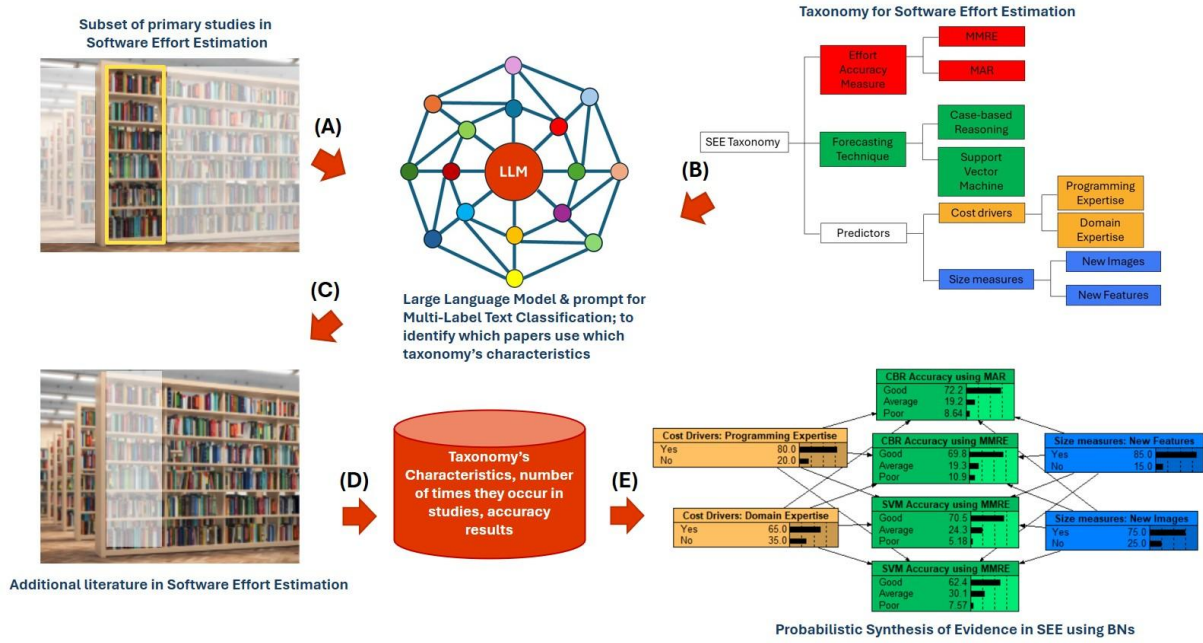


Figure 1 – Overview of the research on probabilistic synthesis of evidence for SEE that motivated the SLR detailed in this paper.

Ralph [14] and Usman et al. [12] argued for the importance of taxonomies in SE because they can serve as valuable tools for advancing the SE field's maturation. Their motivations for the need to create taxonomies in SE are as follows [12],[14]:

- Structured frameworks organise information into categories, so helping practitioners and researchers manage complex domains.
- Reduced ambiguity improves clarity, making it easier to researchers and practitioners to understand concepts and their connections.
- Classification systems establish common terminology, facilitating effective communication and reducing misunderstandings.
- Well-defined classifications simplify knowledge dissemination, spreading best practices and innovations.
- Classifying knowledge helps researchers identify gaps and guide the development of new theories and methods.
- Structured analysis of problems through classification supports informed and systematic decision-making in both research and practice.
- Classified knowledge enables easier selection of appropriate tools and methods, enhancing problem-solving efficiency in industry.

Taxonomies have been proposed for a range of knowledge areas in Software Engineering (SE) (see [18]), such as: Software Construction, Software Design, Software Requirements, Software Maintenance, Software Testing, Software Quality, SE Models and methods, SE process, Software Management, Software Economics; Software Professional practice [12], and Software Effort Estimation (e.g. [27],[28],[29]).

Examples of additional fields, outside of Software Engineering (SE), for which taxonomies also exist include astronomy [23],[24], medicine [25], and quantum computing [26]. However, the best-known example is the biological tree of life [15]. This taxonomy is regularly updated as new species are discovered [16],[17] and such practice should also be embraced with regard to any other taxonomies that are not considered deprecated.

With regard to SEE, despite the clear benefits that taxonomies can bring (e.g. [12],[14]) and the importance of SEE for Software Engineering (e.g. [30]), to our knowledge no fully refereed secondary study on Taxonomies for SEE has been published to date. We were only able to identify a technical report that misleadingly argued to provide evidence on taxonomies for SEE [31]. Further details are given in the Related work and Discussion sections.

Herein the phenomenon of interest, as well as population, and outcome are all properly specified; therefore, we can carry out an aggregative review [32] using as method a Systematic Literature Review. An SLR method includes an objective and systematic selection criteria, which helps to mitigate authors' biases when detailing findings. Furthermore, we also wanted to employ research questions precisely formulated, detailed quality assessment of taxonomies, and clear inclusion & exclusion criteria. These were all fundamental to support the SLR's reproducibility, and to provide scientific evidence. Given these criteria, other existing methods such as narrative reviews [33], critical interpretive synthesis [34], and scoping reviews [35] are not applicable as alternative choices.

Therefore, the main goal and contribution of this paper is to detail a Systematic Literature Review (SLR) that identifies and synthesizes evidence on existing taxonomies for SEE, focusing specifically on:

- (1) The categories & characteristics of these taxonomies;
- (2) Their target software application domain(s);
- (3) (if applicable) the detailing of taxonomies that are evolutions of previously published ones;
- (4) The similarities and differences across the identified taxonomies in terms of structure and content; and
- (5) Their quality, in terms of a given set of attributes proposed in [40].

The remainder of the paper is organized as follows: Section 2 introduces some background knowledge, followed by Section 3 where related work is described; next, the SLR's methodology is detailed in Section 4. Results are presented in Section 5, followed by a Discussion in Section 6. Finally, conclusions and directions for future work are given in Section 7.

2 Background

2.1 Software Effort Estimation: Main Concepts

Every delivered solution within the context of software/software intensive systems (e.g. desktop application, Website, mobile application, services) materializes within the scope of a project and is managed via a development process (e.g. traditional, iterative, agile) [36]. Software effort estimates (e.g., person/hours) are a core input for project and contract management, project budgeting, staffing, scheduling, and trade-off analyses (e.g. [30]). Thus, estimation accuracy is a must.

A Software Effort Estimation (SEE) process (see Figure 1) aims to forecast the effort needed to accomplish a given task, within the context of a software/software intensive development project; such estimation can be carried out solely by humans (expert-based) or in combination with the use of statistical and/or Artificial Intelligence techniques. In general, the actual effort spent on similar finished projects, plus size measure(s) characterizing what is to be developed, and additional factors that are seen as influential upon effort (cost drivers) are combined to obtain an effort estimate for a new project. An example of a size measure and a cost driver can be, respectively, a Website's estimated number of new Web pages and developers' programming experience [37]. As effort is the main component of project costs, so the terms cost and effort are often used interchangeably [30].

Please note that recently there have been several studies investigating the use of Large Language Models to support different SEE tasks such as the automatic measurement of size (e.g. function points [37]), Agile Story Points (e.g. [38]), and to directly estimate effort (e.g. [39]). However, such proposals to automate parts of, or the entire estimation process, do not contradict the main steps that characterise a SEE process, as outlined in Figure 2.

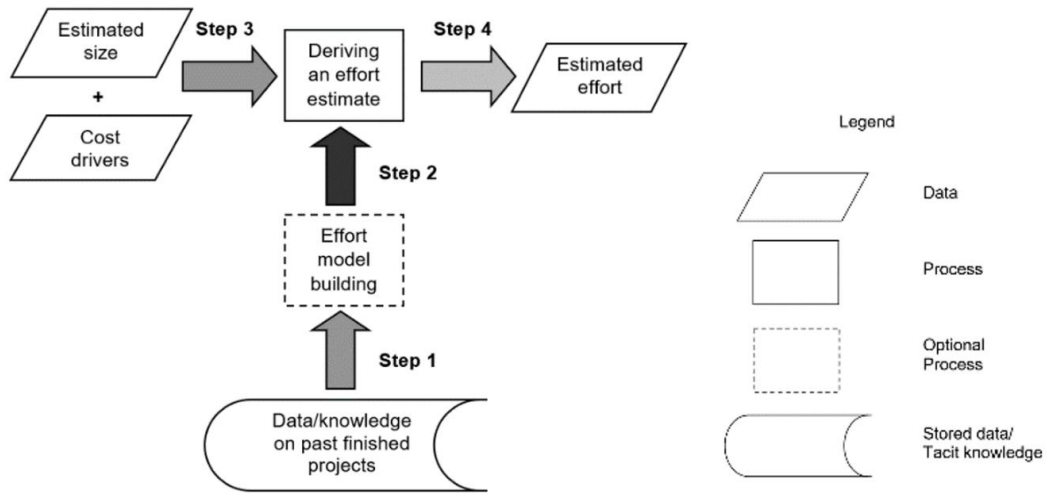


Figure 2: Steps used to obtain an effort estimate [100]

2.2 Taxonomy Terminology

The taxonomy terminology used herein has been presented in Unterkalmsteiner & Abdeen [40], and was adopted in order to be aligned with the framework for taxonomy quality assessment we are employing - also proposed in [40]. Note that this framework is detailed in sub-section 2.3, next.

In general, a Taxonomy is structured using a tree structure, containing nodes and relationships between nodes – edges. These edges represent a parent-child relationship between nodes. Every tree has three types of nodes, as follows:

- Root node: a tree has a single root node, from which all the other nodes originate.
- Leaf nodes: nodes from which no other nodes originate. Herein these nodes are called *characteristics* [40].
- Intermediate nodes: nodes positioned between root and a leaf node. Herein these nodes are called *categories* [40].

Every node, except for the topmost node, always has a single parent node. The topmost node has no parent node. The number of edges between a node n and a tree's root node is n 's depth. Each *category* node can be considered itself as a tree, which is also called a *dimension* of a taxonomy. Therefore, a taxonomy can have several *dimensions*; one representing the entire tree structure, and then others for each of the *categories* part of that taxonomy.

2.3 Framework for Taxonomy Quality Evaluation

The Framework for Taxonomy quality assessment used within the context of this SLR was proposed by Unterkalmsteiner & Abdeen [40]. This study was one of two studies we identified that proposed concrete metrics for data analysis. However, it was the only one putting forward several objective metrics that could be measured without relying on user-feedback/user studies. The other study was by Kaplan et al. [41], who also suggested metrics for taxonomies' quality assessment; however, these metrics are subjective and, in order to be measured, depend upon input from domain experts, which is often not straightforward to obtain and may also lead to biased results.

Unterkalmsteiner & Abdeen [40] developed a comprehensive framework for taxonomy quality evaluation by synthesizing quality attributes from 324 publications across software engineering and information systems research. From an initial set of 43 different quality attributes identified in the literature,

they consolidated overlapping concepts into seven distinct, non-overlapping quality attributes with corresponding measurements (see Table 1). These seven quality attributes were organised according to the measurement framework introduced by Fenton and Pfleeger [42]. Such framework is used to drive the identification of the entities that are the focus of one's measurement goal, their corresponding attributes and how each attribute is to be measured.

Entities can represent processes, products or resources [42]. Processes are collections of activities usually associated with some timescale. An example of a process would be 'the process used to build a Taxonomy for SEE'. Products are any artefacts, deliverables or documents that result from a process activity; thus, an example of a product could be a 'Taxonomy for SEE'. Finally, resources are entities required by a process activity; an example of a resource could be 'a domain expert', who is contributing towards the building of a taxonomy for SEE.

Each attribute can be either internal, or external [42]. An internal attribute of a product, process or resource is that which can be measured by examining the product, process or resource on its own, separate from its behaviour. For example, if we were interested in measuring the product 'Taxonomy for SEE', and its attribute 'depth', then this attribute could be measured by solely examining the Taxonomy's tree structure. As for an external attribute of a product, process or resource, the working of the product, process or resource is what is important, rather than the details pertaining to the entity itself. For example, if we were interested in measuring the product 'Taxonomy for SEE', and its attribute 'ease of use', then this attribute could only be measured by means of some use case involving users, who would then assess how easy the product 'Taxonomy for SEE' is for use in their context(s).

Each of the seven quality attributes, proposed in [40], to assess the quality of a Taxonomy will be detailed next.

Comprehensiveness: Measures to what extent a taxonomy does fully classify the domain knowledge for which it has been created. This attribute measures the quality of the process employed to create the taxonomy and employs two metrics: i) Design process quality, which measures the quality of the process that was used to create the taxonomy; and ii) Count of unclassified objects, which measures the number of objects that are part of a knowledge domain which are not represented in the proposed Taxonomy. The authors in [40] do not explain what is meant for 'objects'. However, within the context of this work, we assume that objects represent knowledge, facts, concepts, and terminology pertaining to a knowledge domain.

Robustness: Measures how well the taxonomy differentiates between related and unrelated characteristics. The method analyses two key properties: *cohesiveness* (related characteristics grouped together) and *orthogonality* (unrelated characteristics in separate groups). The approach identifies groups of characteristics within the taxonomy and calculates the semantic similarity within each group using word embeddings (Metric 'Semantic proximity and distance'). For each group, the minimum pairwise similarity establishes a cohesiveness threshold. An "intruder" is detected when a characteristic from outside the group shows higher similarity to any group member, below this minimum threshold. The robustness score ranges from 0 to 1, where 1 indicates perfect separation with no intruders detected. Note that herein we are using the term 'intruder' because this is the term employed in [40].

$$R(T) = \frac{\sum_{i=1}^{n_{groups}} \left(1 - \frac{n_{ic}}{n_{gc} * (n_{ac} - n_{gc})} \right)}{n_{groups}} \quad (1)$$

The Robustness of a Taxonomy - R(T), is calculated using Equation (1), where:

- n_{groups} is the number of groups of characteristics within the taxonomy (Metric 'Count of Taxonomy constructs')
- n_{ic} is the number of intruder characteristics (Metric 'Count of misclassified objects').
- n_{gc} is the number of group characteristics.
- n_{ac} is the number of all characteristics in the taxonomy.

Note that this attribute also aligns with Ralph's [14] requirement that taxonomies should enable clear distinctions between instances.

Conciseness: This attribute measures to what extent a taxonomy has classified objects using the minimum number of dimensions, categories, and characteristics needed. In other words, it measures a taxonomy's conciseness by evaluating the structural efficiency of the classification hierarchy. Equation 2, below, calculates the sum of inverse depths for all categories and characteristics, where deeper elements (higher depth values) contribute less to the total sum. This approach penalizes overly deep taxonomies, as they require more navigation steps and increase users' cognitive load. Conciseness' score ranges from 0 to 1, where higher values indicate more efficient taxonomy structures that balance breadth and depth effectively. This metric reflects the principle that effective taxonomies should minimize the number of steps required to locate specific characteristics while maintaining logical organization.

$$C(T) = \frac{1}{1 + \ln \left(\sum_{i=1}^{n_{cat}} \left(\frac{1}{d(Cat_i)} \right) + \sum_{j=1}^{n_{char}} \left(\frac{1}{d(Char_j)} \right) - 1 \right)} \quad (2)$$

The Conciseness of a Taxonomy – C(T) is calculated using Equation 2, where:

- $\ln$ is a logarithmic transformation, which ensures that taxonomies with moderate complexity receive higher conciseness scores, while extremely shallow or deep structures are penalized.
- n_{cat} is the number of categories (Metric 'Human memory heuristic', 'Amount and depth of constructs').
- $d(Cat_i)$ is the depth of category i (Metric 'Amount and depth of constructs').
- n_{char} is the number of characteristics (Metric 'Human memory heuristic').
- $d(Char_j)$ is the depth of category j (Metric 'Amount and depth of constructs').

Extensibility: This attribute characterises the ability to modify the taxonomy's structure by adding, changing, or removing dimensions, categories, or characteristics. The two metrics used are: i) 'Documented extension points', which is measured by checking whether the documentation provided with a taxonomy includes information on how the taxonomy can be extended; and ii) 'Rate of change', which measures the frequency with which a taxonomy is changed.

Explanatory: This attribute is characterised by to what extent users can locate objects in a taxonomy based on this taxonomy's existing characteristics (Metric 'Location support'); or to deduce an object's characteristics based on its location in the taxonomy's structure (Metric 'Orientation efficiency and effectiveness').

Mutual exclusiveness: This attribute is used to assess to what extent each object in the knowledge domain embedded in a taxonomy appears in only one category per dimension, providing unique identification. It is measured using two metrics: i) 'Classification constraint', which relates to whether the accompanying documentation provides explicit instructions on how to handle a violation of mutual exclusiveness; and ii) 'Ambiguous objects', which relates to whether an object can be classified in different categories.

Reliability: This attribute represents the extent to which consistent classification results are obtained across different uses. It is measured using two metrics: i) 'Inter-rater reliability', which measures the consistency with which two or more raters classify the same object (are the classification rules understood in the same way by different raters?; Does a single object lead to the same classification, given two or more raters?); and ii) 'Intra-rater reliability', which measures the consistency with which a single rater classifies the same or similar objects at different times.

Table 1 presents the attributes and measurement options proposed by Unterkalmsteiner & Abdeen [40] to measure the quality of taxonomies. This information is a subset of what is shown in [40] as Table 4.

Finally, the seven quality attributes abovementioned address different aspects of taxonomy utility and exhibit important interrelationships, as follows:

- Trade-offs: Unterkalmsteiner & Abdeen [40] identify key trade-offs between attributes, particularly between robustness and conciseness: increasing the number of dimensions and characteristics (robustness) typically reduces conciseness, though this trade-off is not absolute.
- Complementary relationships: Some attributes work together to support a taxonomy's effectiveness. For instance, explanatory power supports reliability by helping users make consistent classification decisions, while comprehensiveness and mutual exclusiveness both address content adequacy but from different perspectives - completeness versus uniqueness.
- Use-case dependent prioritization: Unterkalmsteiner & Abdeen [40] emphasize that the attributes' importance may vary according to a taxonomy's context of use. Regarding automated classification systems, robustness and mutual exclusiveness may be prioritized, while human-centred applications might emphasize explanatory power and conciseness.

Entity: Product	Quality Attributes		Metrics used to measure the quality attributes	
	Name	Nature	Metric	Scale
Taxonomy for SEE	Comprehensiveness	External	Design process quality	Nominal
			Count of unclassified objects	Absolute
	Robustness	External	Count of taxonomy constructs	Absolute
			Semantic proximity and distance	Ratio
			Count of misclassified objects	Absolute
	Conciseness	External	Human memory heuristic	Absolute
			Amount and depth of constructs	Ratio
			Count of unused constructs	Absolute
	Extensibility	Internal	Documented extension points	Nominal
			Rate of change	Ratio
	Explanatory	External	Location support	Nominal
			Orientation efficiency and effectiveness	Ratio
	Mutual exclusiveness	External	Classification constraint	Nominal
			Ambiguous objects	Ratio
	Reliability	External	Inter-rater reliability	Ratio
			Intra-rater reliability	Ratio

Table 1: Attributes and measurement options proposed by Unterkalmsteiner & Abdeen [40] to measure the quality of taxonomies. This information is a subset of what is showing in [40] as Table 4.

2.4 Validation of the Primary Studies

To identify the validation employed by the primary studies included in this SLR, we employed the classification framework proposed by Wieringa et al. [43]. This framework - initially developed for use in Requirements Engineering research, distinguishes between research activities (which generate new knowledge) and design activities (which create new artifacts or techniques).

Wieringa et al. [43] identify six types of research papers according to the type of evaluation they include:

- **Evaluation Research:** Investigates problems in practice or implementations of techniques already deployed in real-world settings.
- **Solution Proposal:** Proposes a solution to a problem without a full-blown validation. Typically includes proof-of-concept demonstrations.
- **Validation Research:** Investigates the characteristics of proposed solutions that have not yet been validated in practice, and in general employs methods such as experiments, simulations, prototyping, or mathematical analyses/proof of properties.
- **Philosophical Papers:** Presents a new conceptual framework describing a new way of looking at a problem.
- **Opinion Papers:** Expresses the author's opinion about what they see as good or not so good regarding a particular problem.
- **Personal Experience Papers:** Describes the author's personal experience regarding a problem.

The use of this classification enabled us to identify the type of validation employed by the primary studies in this SLR.

2.5 Software Application Domains

Taxonomies in SEE can cover different software application domains and we use the classification proposed by Pressman & Maxim [44] to identify the domains that correspond to those targeted by the primary studies included herein.

Pressman & Maxim [44] propose seven broad categories:

System software: Refers to software solutions that support other programs, such as compilers, operating systems, and networking software.

Application software: Involves standalone software solutions designed to meet specific business requirements and handle business or technical data.

Engineering/scientific software: Covers data science and computational software solutions spanning various fields from astronomy to automotive stress analysis.

Embedded software: Refers to software solutions integrated into products or systems to enable control features and functions.

Product-line software: Involves reusable software solutions and components built to deliver specific capabilities for multiple customers.

Web/mobile applications: Includes network-focused software solutions such as browser-based apps, cloud computing, and mobile device software.

Artificial intelligence software: Applies heuristics to address complex problems in fields such as robotics, machine learning, and pattern recognition.

We note that the SWEBOK (Software Engineering Body of Knowledge) was not adopted as a reference framework for evaluating SEE taxonomy coverage. The SWEBOK chapter on Software Engineering Economics addresses estimation at a conceptual level, providing only a single high-level "Estimation" entry that lists possible ways to obtain an estimate, without detailing specific effort predictors (such as size measures and cost factors) at the characteristic level required for meaningful taxonomy evaluation. For example, the largest taxonomy in our SLR contains 164 characteristics across 12 categories, representing specific effort predictors, a level of granularity that SWEBOK does not provide for the SEE domain. Furthermore, the SWEBOK estimation entry omits the use of machine learning and AI methods for effort estimation. Pressman & Maxim's classification [44], with its seven software application domain categories, provides a more appropriate and directly relevant framework for assessing how well existing SEE taxonomies cover the diversity of modern software development contexts.

2.6 Methodology for comparing Taxonomy elements

Taxonomies can be constructed and deconstructed from word embeddings (e.g. [45],[46],[47]). The use of text embeddings for comparing taxonomy categories and characteristics is grounded in the distributional semantic hypothesis, which establishes that words appearing in similar linguistic contexts tend to

have similar meanings [49]. Modern embedding models operationalize this principle at scale, and have been empirically validated for qualitative research tasks involving terminology comparison [50]. In our analysis, embeddings are used specifically for a term-matching task, that is identifying lexically and semantically similar terms across taxonomies rather than for deep definitional analysis. This approach follows Unterkalmsteiner & Abdeen [40], who explicitly recommend the use of "natural language processing techniques to measure Robustness via semantic proximity." We acknowledge that embeddings cannot capture all definitional nuances; however, when combined with conservative similarity thresholds and transparent reporting of matched pairs, they provide a systematic and reproducible method for initial identification of potentially equivalent terms, while human expert judgment remains essential for any actual taxonomy merging decisions.

To support full reproducibility, the following technical specifications were employed in our semantic similarity analysis. The embedding model selected was jina-embeddings-v3 [54]. It produces 1024-dimensional embeddings with task-specific LoRA adapters, when using the task parameter 'text-matching' and default inference configuration. This model was selected based on published benchmark performance, where it achieved 65.56 average score on the MTEB benchmark, outperforming text-embedding-3-large (64.59) and Cohere-embed-v3 (64.47) [54]. In relation to the preprocessing that was done, we retained full characteristic phrases as input (e.g., "Team Experience Level"), with no stemming or lemmatization applied, and using UTF-8 encoding. The similarity metric employed was the cosine similarity between embedding vectors, where a threshold of 90% cosine similarity (Table 27) was applied. This was deliberately a conservative threshold choice that was chosen because it identifies only high-confidence near-identical terms, minimizing false positives.

3 Related work regarding taxonomies in SEE

We only identified a technical report by Vera et al. [31] when searching for secondary studies detailing syntheses of evidence on taxonomies for SEE.

This report details a survey of primary studies obtained from searches using several databases – IEEE Xplore, ACM, ScienceDirect, and Springer, and within the timeframe 2000 to 2017. They state that 17 taxonomies for SEE were identified, as will be detailed in Section 6; however, none of the included primary studies proposed any taxonomy for SEE. The studies included by Vera et al. were solely primary studies in effort estimation (e.g. studies comparing machine learning techniques) that did not propose any taxonomy for SEE. Vera et al. themselves decided to create post-hoc hierarchical structures based upon each of the primary studies, and then to argue that the primary studies they included proposed taxonomies for SEE. These made-up structures focus primarily on categorizing estimation methods rather than analysing the taxonomies themselves as knowledge organization structures.

Furthermore, this technical report also revealed additional significant methodological mistakes regarding adherence to systematic literature review guidelines, study selection procedures, and evidence synthesis approaches that compromise even further the reliability of its findings. These methodological issues are discussed in detail in Section 6 alongside our discussion.

We chose to present Vera et al. [31] work in the related work section for the sake of transparency, and also in support to our previous statement that this SLR is the first to aggregate the evidence pertaining to taxonomies in SEE, which is a novel research contribution not only to research in SEE but also to practice.

4 Systematic Literature Review

This section details the SLR that we carried out. It was executed using primarily the SEGRESS guidelines in SE [55] (Table 22 in Appendix A), as well as the PRISMA checklist [56]; however, it also considered the SLR guidelines by Kitchenham & Charters [57].

An SLR is defined as a method that enables the evaluation and interpretation of a significant sample of available research relevant to a proposed question, and a specific subject matter or interest; it generally comprises the following steps [57]:

1. Identification of the need for carrying out a systematic review.
2. Formulation of focused research question.

3. A comprehensive, exhaustive search for primary studies.
4. Quality assessment of included studies.
5. Identification of the data needed to answer the research question.
6. Data extraction.
7. Summary and synthesis of study results.
8. Interpretation of the results.
9. Report-writing.

The SLR protocol is provided in Appendix C and the replication package in our Github repository¹

4.1 The Motivation for a systematic literature review

As previously stated, the aim of the SLR detailed herein is to identify the existing taxonomies for SEE so to assess whether any of them could be readily used as input to build a BN model for SEE probabilistic evidence synthesis [90], or whether another taxonomy would need to be proposed and validated. Furthermore, although the first author has (co-)proposed Taxonomies for SEE, the most recent of these taxonomies was published in 2017, and therefore it was important to identify whether additional Taxonomies for SEE have been published afterwards.

As previously stated, given that the phenomenon of interest, as well as population, and outcome are all properly specified, we can carry out an aggregative review [32] using as method a Systematic Literature Review. An SLR method includes an objective and systematic selection criteria, which helps to mitigate authors' biases when detailing findings. Furthermore, we also wanted to employ research questions precisely formulated, detailed quality assessment of taxonomies, and clear inclusion & exclusion criteria. These were all fundamental to support the SLR's reproducibility, and to provide scientific evidence.

In addition, an SLR on taxonomies for SEE can provide the means for an in-depth analysis relating to the identified primary studies, how these taxonomies compare to one another, their quality and type of validation. Furthermore, such an SLR also provides the means to identify, for example, specific types of software application domains that may lack representation, as well as to put forward suggestions for future work in taxonomies for SEE.

4.2 Research questions

Table 2 below presents the SLR's research questions and their corresponding motivation.

Research Question	Main Motivation
RQ1: What are the categories and characteristics used in the existing taxonomies for <i>SEE</i> ?	A concise examination of the categories and characteristics in the existing taxonomies is of utter importance. With this information we can assess to what degree of specificity do these taxonomies capture the complexities of software effort estimation, and how representative they are regarding the landscape of applications being developed worldwide.
RQ2: What are the software application domains targeted by the taxonomies identified in this SLR?	Such understanding will provide the means to know the extent to which all the different types of applications are represented in the existing taxonomies for SEE; it also supports the identification of potential limitations and applications for which a taxonomy needs to be elaborated/extended. We use the software application domains proposed by Pressman & Maxim [44], which was introduced in sub-section 2.4.
RQ3: (if applicable) Which of the taxonomies identified in this SLR are an evolution of previously published ones?	Here the goal is to determine whether taxonomies are evolving from earlier versions, and, if applicable, to also identify the specific changes being made that represent

Which aspects were changed?	such evolution. This is a way of understanding advancements in this field and whether there are patterns relating to such changes.
RQ4: What are the similarities and differences across the identified taxonomies for SEE in terms of their structures and content?	Within the context of this question, structural aspects represent dimensions, categories and depth; whereas content aspects represent shared versus unique categories and characteristics. The understanding of similarities and differences between taxonomies will help understand whether these taxonomies are characterising the SEE phenomenon in a similar or diverse way.

Table 2: SLR's Research Questions

4.3 Developing a search strategy

Following the definition of the research questions, a concise search strategy was developed, comprising the following steps:

1. Conduct a comprehensive search using a carefully constructed search string across selected, reputable databases.
2. Screen titles and abstracts to identify potential studies, applying predefined inclusion and exclusion criteria.
3. Assess the inclusion of studies that appear relevant based on the initial screening.
4. Resolve disagreements among reviewers through discussion. If consensus cannot be reached, involve a third reviewer for arbitration.
5. Whenever studies' relevance remains unclear, review the full text to make a final determination.

4.4 Study selection

4.4.1 Inclusion & Exclusion criteria

The study selection was carried out by applying an inclusion and exclusion criteria to each of the studies that were retrieved via one or more databases searches. Please note that both inclusion and exclusion must be explicitly stated, and detailed as best as possible, so to enable the possible replication of this study and the clear specification on how to identify primary studies that are/are not relevant to be selected for this SLR. A primary study herein is a study that proposes/uses a taxonomy for SEE. This means that studies that solely investigate the SEE phenomenon, e.g., via comparing the accuracy of effort estimation techniques, without explicitly proposing a taxonomy for SEE, are not to be included in this SLR. Also note that we did not constrain the search to any specific time range because we wanted to include taxonomies for SEE proposed at any point in time.

- ✓ **Inclusion criteria:** (IC1) Studies that propose and/or use a taxonomy for software effort estimation (AND) (IC2) Studies published at peer-reviewed journals and conference papers (IC3) Studies written in English.
- ✗ **Exclusion criteria:** (EC1) Studies that are not related to software effort estimation (OR) (EC2) Studies that do not propose or discuss a taxonomy for SEE (OR) (EC3) Studies that are Grey literature (OR) (EC4) Studies not written in English.

It is important to stress that the research goal of this SLR is to present an evidence synthesis of taxonomies for SEE; therefore, despite the large body of research in SEE, only primary studies that have proposed/used a taxonomy for SEE were included herein. Such distinction drove both the formulation of the inclusion and exclusion criteria above, and the detailed screening of titles and abstracts carried out. Furthermore, note that IC4 imposes no temporal restriction, meaning that any taxonomy for SEE published at any point in time, including before the year 2000, would have been included had it been identified in our search. Note that while reading titles and abstracts, if at any point these were not sufficient to identify whether a paper was to be included or excluded, then the full text was also employed.

4.4.2 Quality assessment

The quality assessment of primary studies' taxonomies for SEE was performed using the framework by Unterkalmsteiner & Abdeen [40] (See sub-section 2.3). This framework proposes metrics that can be used to assess both the process employed to build a taxonomy, and the product that is the result of such process – the taxonomy itself. Note that our assessment of existing taxonomies for SEE is mostly based upon the reporting provided in the conference/journal papers describing such taxonomies. The exception is for Comprehensiveness, where we also searched for authors' backgrounds online.

We note that our approach combines quantitative methods (semantic similarity analysis, structural metrics, R(T) and C(T) calculations) with qualitative methods (thematic analysis of taxonomy content, systematic quality assessment, and validation, type classification), constituting a mixed-methods analysis.

Measurement Implementation:

Our evaluation focuses on attributes where reliable structural assessment is possible:

- Robustness: Calculated using semantic similarity analysis with jina-embeddings-v3
- Conciseness: Measured through structural complexity calculations
- Comprehensiveness: Partially assessed through available documentation. If a taxonomy paper did not provide any background information for its author(s), we searched for it online.

Following Unterkalmsteiner & Abdeen [40], we employed natural language processing techniques to measure Robustness via semantic proximity. We used the *jina-embeddings-v3* model, a multilingual, multi-task text embedding model that has demonstrated superior performance compared to proprietary models from OpenAI, Cohere, XLM-R, and mBERT on both monolingual and multilingual tasks [54].

Furthermore, in addition to using Unterkalmsteiner & Abdeen's taxonomy quality assessment framework [40], we also employed the classification by Wieringa et al. [43] (See sub-section 2.4) to analyse the type of taxonomy validation employed by each primary study included in our SLR. In both cases, each paper was classified by the second author and revised by the first author.

4.5 Data collection & Extraction

To define an appropriate search strategy for identifying primary studies, we first constructed a comprehensive search string (detailed in Sub-section 4.5.1). To achieve this, we selected relevant keywords (see Table 3) derived from some previous literature on taxonomies in SE ([12],[14]); and taxonomies in SEE that we were already aware of [27],[28],[29].

Keyword	Source(s)
Taxonomy	[12],[14],[27],[28],[29]
Estimation	[27],[28],[29]
Software	[12],[14],[27],[28],[29]
Classification	[12],[14],[29]
Knowledge Classification	[28]
Effort estimation	[27],[28],[29]
Software Engineering	[12],[14],[27],[28],[29]

Table 3: Table of relevant keywords collected from known literature.

In addition to the abovementioned keywords, and following the SLR guidelines that were used to guide our work, we also included relevant synonyms (see Table 4), identified via the Cambridge Thesaurus [61]. The inclusion of synonyms is critical, as different authors may employ varying terminology when addressing similar or related concepts. By incorporating synonyms, we ensured the broadest possible retrieval of studies, irrespective of the vocabulary used.

Keyword	Synonym(s)
Taxonomy	Classification, grouping, sorting
Estimation	Forecast, forecasting, prediction, assessment
Software	Application, program
Classification	Taxonomy, grouping, sorting
Knowledge classification	Classification
Effort estimation	Effort forecast, effort forecasting, effort prediction, effort assessment, cost forecast, cost forecasting, cost prediction, cost assessment
Software Engineering	—

Table 4: Table of synonyms

The data collection process was performed using two major academic databases: Scopus [62] and Google Scholar [63]. Scopus indexes a wide variety of journals including those published by the ACM [64] and IEEE [65], and contains thus an extensive database of peer-reviewed journals, conference proceedings, and other scholarly content, making it a reliable source for high-quality academic publications [66]. Google Scholar was selected for its broad accessibility and its ability to index a wide range of academic documents published by the ACM, IEEE, and other publishers (e.g. articles, theses, books, and conference papers), thus ensuring a comprehensive review of the literature [67]. Forward snowballing was not performed as we did not fix the search period on any of the two databases used. Backward snowballing was performed by the first author, and no additional studies were included.

4.5.1 Data collection from Scopus

Using the identified keywords and their synonyms we formulated a search query designed to scan titles and abstracts in the Scopus database. The query was constructed to be sufficiently broad to yield relevant results, while also being narrow enough to avoid retrieving an excessive number of false positives:

TITLE-ABS ((taxonomy OR classification OR grouping OR sorting) AND ("effort estimation" OR "Effort Forecast" OR "Effort Forecasting" OR "Effort Prediction" OR "Effort Assessment" OR "Cost Estimation" OR "Cost Forecast" OR "Cost Forecasting" OR "Cost Prediction" OR "Cost Assessment")) AND ("Software Engineering" OR software OR application OR program))

4.5.2 Data collection from Google Scholar

To further ensure a comprehensive coverage, we conducted a parallel search in Google Scholar using the Scholarly package [68] in Python. Given the inherent limitations of Google Scholar, such as restricting the retrieval of results to the first 1,000 entries and offering less refined search functionality, when compared to Scopus, we performed multiple searches using simpler search queries, which were defined jointly by both authors:

"Taxonomy and Effort Estimation"
"Taxonomy and Cost Estimation"
"Taxonomy and Effort Prediction"
"Taxonomy and Cost Prediction"

Duplicates were removed by comparing the titles of the retrieved papers, and Python was used to automate the deduplication process.

Following the collection of relevant papers, we proceeded with data extraction. A data extraction sheet was created (see Table 23 in Appendix A) and used to guide the systematic extraction of key information from each selected study.

5 SLR Results

The Scopus search retrieved a total of 356 articles. Both authors screened separately through all the titles and abstracts for relevant taxonomies in SEE; they also afterwards checked whether all previously known relevant SEE taxonomy papers were included in the search results; only one paper required a discussion until consensus was reached:

Karadimos and Anthopoulos [70] → This study, entitled ‘A taxonomy of machine learning techniques for construction cost estimation’ was selected by one of the authors but later discarded because it investigated effort estimation for building construction projects, not software projects (EC1).

Regarding the Google Scholar search, a total number of 3125 documents was selected, after removing duplicates. Of these, the first 1000 were screened separately by both authors. This screening led to the preliminary selection of 11 primary studies for inclusion in this SLR; however, when comparing both authors’ level of agreement, the following five papers were removed from the preliminary list of 11 papers:

Vera et al. [31] → This study, entitled ‘Survey Of Software Development Effort Estimation Taxonomies’, was removed because it is not a primary study (EC5). However, this technical report, which is a secondary study, is discussed and compared to our SLR in Section 6.6.

Britto et al. [59] → This study, entitled ‘An Extended Global Software Engineering Taxonomy’ [60] was excluded due to being a taxonomy for global software engineering, and not focusing on software effort estimation (EC1).

Usman et al. [12] → This study, entitled ‘Taxonomies In Software Engineering: A Systematic Mapping Study And A Revised Taxonomy Development Method’ was removed because it is not a primary study (EC5).

Sanchez-Morcilio & Quiles-Torres [70] → This study, entitled ‘The Taxonomy Of Estimation In Software Development Projects’ was removed because it did not in fact propose a taxonomy (EC2), but rather a process model on how to do an estimation. Prior to excluding the paper, we contacted Morcilio and Torres so to understand the diagram they presented and claimed to be a taxonomy; However, their reply made it clear that the diagram they used was a process model, not a taxonomy; and the paper was therefore excluded.

Šmite et al. [71] → This study, entitled ‘An Empirically Based Terminology And Taxonomy For Global Software Engineering’ was excluded due to being a taxonomy for global software engineering, and not focusing on software effort estimation (EC1).

The remaining 2125 papers were screened by the second author but no further primary studies were selected for inclusion. Figure 3 illustrates the process that was followed.

The inter-rater agreement was high (Cohen’s $\kappa = 0.83$, observed agreement = 99.6%). Disagreements ($n = 6$) were resolved through discussion.

After the complete screening of papers, six papers were selected for inclusion in this SLR. Backward snowballing was applied on each by the first author, resulting in no additional primary studies to include. Therefore, the final set of primary studies is as follows:

- *Towards a Taxonomy of Hypermedia and Web Application Size Metrics*, by Mendes et al. [29]
- *A Specialized Global Software Engineering Taxonomy for effort estimation*, by Britto et al. [27]
- *A Taxonomy of Web Effort Predictors*, by Britto et al. [28]
- *An effort estimation Taxonomy for Agile Software Development*, by Usman et al. [59]
- *A Software Cost Estimation Taxonomy for Global Software Development Projects*, by Bajta & Idri [72]
- *A Taxonomy of Approaches and Methods for Software Effort Estimation*, by Dashti & Gandomani [73]

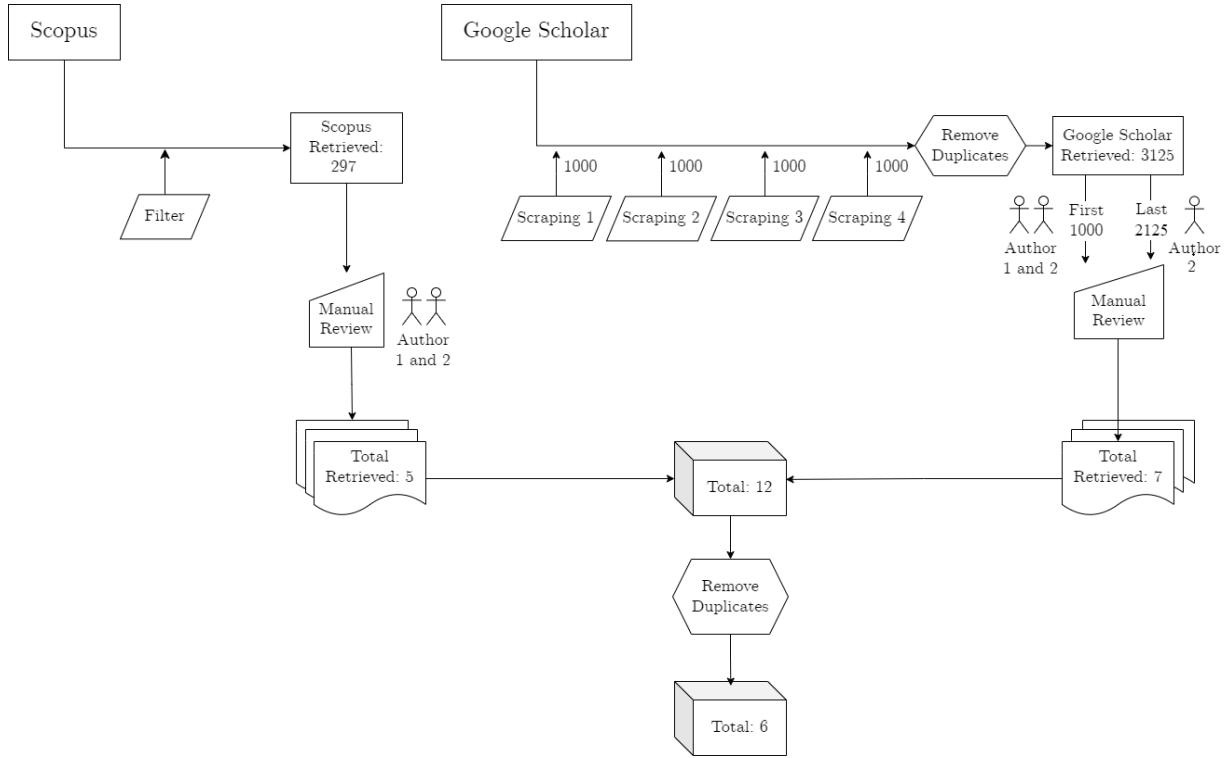


Figure 3: Diagram illustrating the data collection process.

Table 5 shows the year of publication and publication venue for each of the primary studies. The earliest study was published in 2005 by Mendes et al. [29], and the most recent one was published in 2022 by Dashti & Gandomani [73]. These results show that since the publication of the first taxonomy for SEE by Mendes et al., there was a gap of 11 years until other taxonomies for SEE were published. Then there was some sort of steady flow, with taxonomies for SEE being published every year/second year, until 2022. We speculate that the interest in taxonomies for SEE from 2016 onwards might have been related to the increase in the number of SEE studies, which was in part motivated by the use of a wide range of statistical and machine learning techniques for building forecasting models. Furthermore, as Mendes was also a co-author in some of the more recent studies proposing taxonomies for SEE, this also suggests that some of these more recent publications could indeed have been prompted by Mendes work in the SEE area.

The primary studies included herein were published in a range of venues including international conferences (e.g. ICSoft, ICWE, ICGSE, ICICSE) and specialized journals (e.g. Journal of Web Engineering, International Journal of Software Engineering and Knowledge Engineering). Such range may suggest a wider interest in taxonomy research within SEE and SE.

Finally, IC4 imposed no temporal restriction on the search, yet the earliest taxonomy identified dates from 2005, indicating that no taxonomies for SEE, written in English, have been published on peer reviewed venues for the first three decades of research in this field. This could be understood as a research gap related to the systematic organisation of the SEE knowledge.

Taxonomy	Year of publication	Publication venue
Mendes et al. [29]	2005	Web Engineering (ICWE 2005)
Britto et al. [27]	2016	IEEE 11 th International Conference on Global Software Engineering
Britto et al. [28]	2017	Journal of Web Engineering
Usman et al. [59]	2017	International Journal of Software Engineering and Knowledge Engineering Vol.27, No.4
Bajta & Idri [72]	2019	14th International Conference on Software Technologies (ICSOFT)
Dashti & Gandomani [73]	2021	4th International Conference on Interdisciplinary Computer Science and Engineering (ICICSE)

Table 5: Summary Table with the year and publication venue for the primary studies included in this SLR.

5.1 Quality Assessment

5.1.1 Taxonomy quality assessment

We applied the guidelines & corresponding attributes proposed by Unterkalmsteiner & Abdeen [40] (See sub-section 2.3) to measure the quality of the taxonomies for SEE included herein. The results are shown in Table 6 and discussed below.

One of the first observations is that most taxonomies originated from a diverse set of data sources.

The robustness value $R(t)$, which uses the semantic similarity of a taxonomy's contents to indicate varying degrees of reliability, is relatively high throughout all studies. Here, Mendes et al. [29] presented the highest score (0.9094), which translates as the most robust taxonomy according to the specified measurement criteria. In contrast, Britto et al. [27] has the lowest robustness (0.8733), likely due to its construction method, which includes repeated terms to improve readability and conciseness.

Conciseness varies significantly among the taxonomies in terms of dimensions, categories, and characteristics. Britto et al. [27] has the highest number of dimensions (6) while Mendes et al. [29] has the lowest (1); Bajta & Idri [72] has the highest number of categories (23), while Britto et al. [27] and Dashti & Gandomani [73] have the lowest (6); Britto et al. [28] has the highest number characteristics (164), while Dashti & Gandomani [73] has the lowest (12).

Finally, Dashti & Gandomani [73] present the highest taxonomy depth (5), while all remaining studies share the same depth - three.

When it comes to a taxonomy's overall conciseness, taxonomies containing less elements might be more straightforward to present but potentially less detailed, while others may offer extensive detail and nuances but at the cost of clarity. This is reflected in the $C(t)$ values, where taxonomies containing more categories and characteristics score lower than the ones having fewer elements, indicating that simpler taxonomies are easier to understand but may sacrifice robustness.

Extensibility measurements are sparse across taxonomies. Only Britto et al. [28] mentions about specific aspects of change processes and extensions. Mendes et al. [29] and Britto et al. [27] document possible future changes, but guidelines for adaptation or customization are not given.

None of the taxonomies provide structured elements besides dimensions or support for selecting categories, thus highlighting a gap in explanatory capabilities. The absence of data on the classification constraint for mutual exclusiveness points to a weakness in the clarity of taxonomy structures. Here, there are several attributes that could not be measured as they were not documented in the studies (e.g. authors' background, support for choosing dimensions, classification constraint, structuring elements besides dimensions, customization of the taxonomy). The fact that none of the studies provided any information in relation to these attributes may perhaps suggest that these were not seeing by the primary studies' authors as core taxonomy quality attributes. However, such lack of information should not, in our view, be taken to mean that the abovementioned attributes are irrelevant. Future research could include the quality measurement of other taxonomies in SE, so to understand further how existing SE taxonomies have or have not catered for the quality attributes proposed in [40].

With regard to presenting data on real world use, only Usman et al. [58] explicitly states that their taxonomy has been applied in industry, while Britto et al. [27] uses solely a proof of concept. These results suggest that real-world validation is an area that needs further attention.

While comprehensiveness and robustness were the two attributes documented the most in the studies included herein, the lack of external evaluations across most taxonomies raises concerns. While some of the included studies propose internal evaluation methods like orthogonality, none have been evaluated by third parties. In addition, another potential area for future work relates to a taxonomy's reliability, which has only been measured in Britto et al. [27].

Limited data on extensibility calls for more flexible taxonomies that can evolve over time. Developing specific guidelines for customization could improve the practical utility of these taxonomies and their resilience through time.

The absence of any information relating to mutual exclusiveness and explanatory attributes suggest the need for both usability studies on the use of such taxonomies (for mutual exclusiveness) and guidelines on how/whether characteristics could be moved within a taxonomy's structure.

Quality Attribute	Measurement	Bajta & Idri [72]	Britto et al. [28]	Britto et al. [27]	Dashti & Gandomani [73]	Mendes et al. [29]	Usman et al. [58]
Comprehensiveness	Does the taxonomy originate from a diverse set of data sources?	Yes	Yes	Yes	Yes	Yes	Yes
	Do the taxonomy creators have a diverse background?	Not mentioned	Not mentioned	No	Not mentioned	Not mentioned	Not mentioned
	Were diverse data analysis methods used to create the taxonomy?	Yes	Yes	Yes	Yes	Yes	Yes
	Has the taxonomy been evaluated by external experts?	No	No	No	No	No	No
Robustness	R(T)	0.8955	0.8956	0.8733	0.8417	0.9094	0.8759
Conciseness	Dimensions	4	2	6	2	1	4
	Categories	23	12	6	6	8	26
	Characteristics	63	164	17	12	26	83
	Maximum Depth	3	3	3	5	3	3
	C(T)	0.2400	0.2077	0.3645	0.3812	0.2815	0.2270
Extensibility	Change process covers the following constructs	Not mentioned	Can be extended to account for predictors that also impact quality, design, and maintenance of Web applications	Dimension Extension	Methods	Web applications	Not mentioned
	Change process covers the following change types	Not mentioned	Web predictors, effort estimation techniques, accuracy metrics, within the Web resource estimation context	Software testing, Software Quality	Methods	Web applications	Not mentioned
	Mandate to change the taxonomy	Not mentioned	Authors	Authors	Not mentioned	Not mentioned	Not mentioned
	Customization of the taxonomy	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
	RoC	Cannot calculate	Cannot calculate	Cannot calculate	Cannot calculate	Cannot calculate	Cannot calculate
Explanatory	Structuring elements besides dimensions?	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
	Support for choosing dimensions or categories?	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
Mutual exclusiveness	Classification constraint	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
Reliability	(Requires the application of the taxonomy)			Yes, taxonomy has been applied	(Requires the application of the taxonomy)		

Table 6: Summary Table with the measured quality attributes

5.1.2 Taxonomies' Type of Validation

The type of validation associated with each of the taxonomies for SEE included herein was categorised according to Wieringa et al.'s classification [43]. Table 7 summarizes the classification given to each paper; each paper was classified by the second author and the classification later checked by the first author.

Taxonomy	Classification of the paper
Bajta & Idri [72]	Solution Proposal
Britto et al. [28]	Solution Proposal
Britto et al. [27]	Evaluation Research
Dashti & Gandomani [73]	Solution Proposal
Mendes et al. [29]	Solution Proposal
Usman et al. [58]	Solution Proposal

Table 7: Summary Table with classification of each of the collected papers.

All papers, except for Britto et al. [27] belong to Wieringa et al.'s solution proposal category, which was identified as such based upon the clarity of the problem stated, the novelty or improvement of the proposed technique, and the logical argumentation for its relevance. Although they do not include a validation in a real setting, they address the criteria included in [40] by clearly stating the problem they intend to solve and detailing their methodology. In solution proposals, Wieringa et al. emphasize the need for sufficient detail in describing the technique so that it can later be validated by others, and these studies achieve such requirement.

Britto et al.'s taxonomy [27] fits in Wieringa et al.'s evaluation research class. Here Wieringa et al. state that practical evaluations should validate the performance of the solution proposed, with criteria focusing on real-world applicability, relevance, and soundness of the method. The study conducts an empirical assessment of their taxonomy in an industrial setting and analyse their impact, thus confirming the practical effectiveness of the proposed solution and complying with Wieringa et al.'s criteria for a paper to be classified as evaluation research.

These assessments are also in line with the taxonomies' quality assessment shown in Table 6, specifically in relation to their reliability.

5.2 Addressing the Research Questions

5.2.1 RQ1: What are the categories and characteristics used in existing taxonomies for SEE?

Figure 4 to Figure 9 below display the hierarchies pertaining to each of the six Taxonomies for SEE included in this SLR). Due to space constraints, only levels 1 and 2 are shown in a hierarchical format. The information shown in Tables 8 to Table 13, Table 22 (see Appendix A), and Figures 4 to 9 were used to answer RQ1.

When observing the structures of these six taxonomies and their categories and characteristics, we can see that they collectively reflect a varied landscape, jointly accounting to a total of 81 categories and 365 characteristics. They vary substantially in structural complexity, where:

- i) The number of dimensions ranges from 1 (Mendes et al. [29]) to 6 (Britto et al. [27]);
- ii) The number of categories ranges from 6 (Britto et al. [27] and Dashti & Gandomani [73]) to 26 (Usman et al. [58]);
- iii) The number of characteristics ranges from 12 (Dashti & Gandomani [73]) to 164 (Britto et al. [28]); and
- iv) The maximum depth ranges from 3 to 5.

Such structural variation reflects fundamentally different design choices: some taxonomies prioritise breadth (many categories and characteristics) while others prioritise depth (more hierarchical levels) or conciseness (fewer elements).

Furthermore, the taxonomies differ in their content focus:

- i) Bajta & Idri [72] and Usman et al. [58] integrate both effort predictors (size measures and cost factors), estimation techniques and estimation context, providing broader coverage;
- ii) Britto et al. [27] and [28] focus solely on cost factors;
- iii) Mendes et al. [29] exclusively on size measures; and
- iv) Dashti & Gandomani [73] exclusively on estimation techniques.

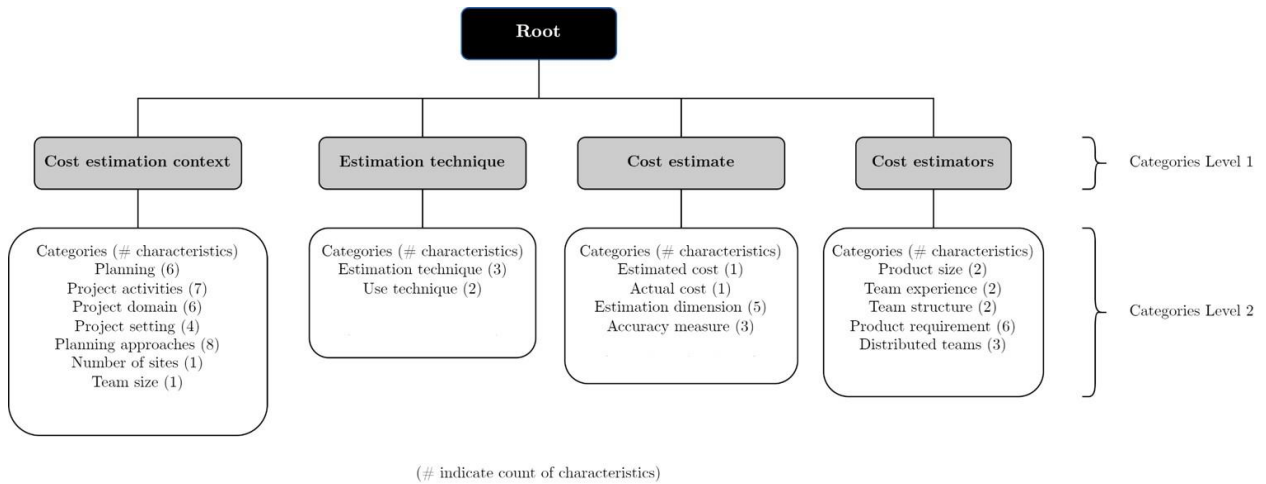


Figure 4: Structure of the taxonomy for SEE proposed by Bajta & Idri [72].

Categories and characteristics from Bajta & Idri [72]

Taxonomy	Categories Level 1	Categories Level 2	Characteristics
Bajta & Idri [72]	Cost estimation context	Planning	Conceptualization, Feasibility study, Preliminary planning, Detail planning, Execution, Commissioning
		Project activities	System investigation, Analysis, Design, Implementation, Testing, Maintenance, Other
		Project domain	SE, Research & dev, Telecommunication, Finance, Healthcare, Other
		Project setting	Close onshore, Distant onshore, Near offshore, Far Offshore
		Planning approaches	Constructive Cost Model, Capability Maturity Model Integration, Agile, Delphi, Quality Assurance, Case-Based Reasoning, Fuzzy similarity, Other
		Number of sites	Number of sites.value
		Team size	N° of team members
	Estimation technique	Estimation technique	Expert judgment, Non-machine learning, Machine learning

	Cost estimate	Use technique	Individual, Group-based estimation
		Estimated cost	Estimate.value
		Actual cost	Actual cost.value
		Estimation dimension	Effort hours, Staff/cost, Hardware, Risk, Portfolio
		Accuracy measure	Baseline comparison, Variation reduction, Sensitivity analysis
	Cost estimators	Product size	Size report, Statistics analysis
		Team experience	Considered, Not considered
		Team structure	Considered, Not considered
		Product requirement	Performance, Security, Availability, Reliability, Maintainability, Other
		Distributed teams distances	Geographical distance, Temporal distance, Socio-cultural distance

Table 8: Table displaying all categories and characteristics of the taxonomy for SEE proposed by Bajta & Idri [72]

Bajta & Idri [72] propose a taxonomy that includes cost factors, cost estimates, estimation techniques and contextual factors. It presents in total 23 categories and 63 characteristics. This taxonomy includes size measures such as *size report*; cost factors including *team experience (considered/not considered)* and *distributed teams distances (Geographical distance, Temporal distance, Socio-cultural distance)*; estimation techniques like *expert judgment* and *Machine learning*, and contextual factors such as *project setting (Close onshore, Distant onshore, Near offshore, far offshore)*.

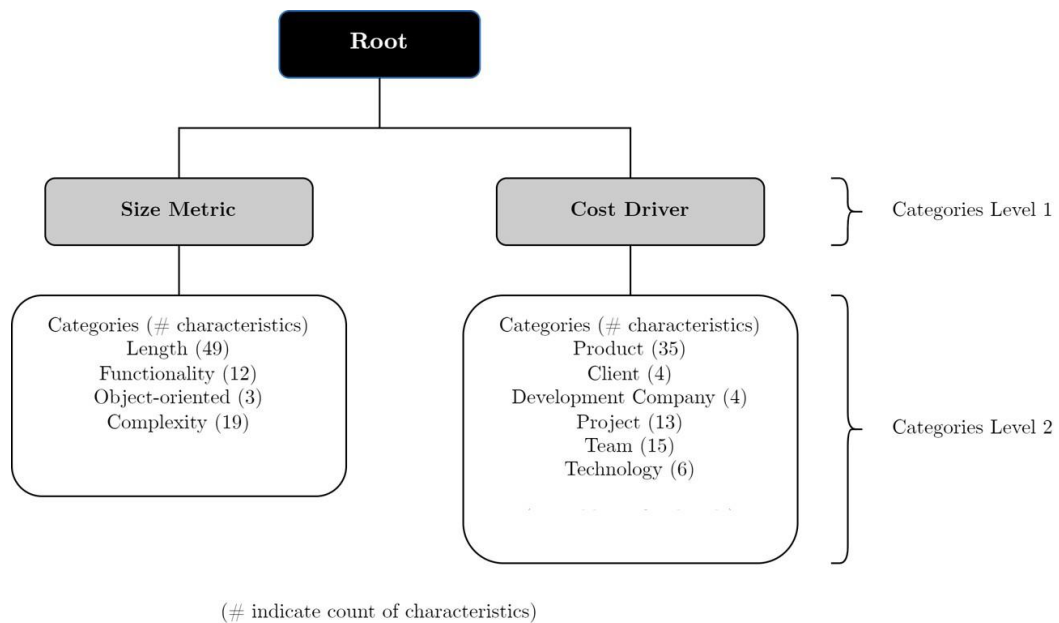


Figure 5: Structure of the taxonomy for SEE proposed by Britto et al. [28]

Categories and characteristics from Britto et al. [28]

Taxonomy	Categories Level 1	Categories Level 2	Characteristics
Britto et al. [28]	Size Metric	Length	Web page count, Media count, New media count, New Web page count, Link count, Program count, Reused component count, Lines of code, Reused program count, Re-used media count, Web page allocation, Reused lines of code, Media allocation, Reused media allocation, Entity count, Attribute count, Component count, Statement count, Node count, Collection slot size, Component granularity level, Slot granularity level, Model node size, Cluster node size, Node slot size, Publishing model unit count, Model slot size, Association slot size, Client script count, Server script count, Information slot count, Association center slot count, Collection center slot count, Component slot count, Semantic association count, Segment count, Slot count, Cluster slot count, Cluster count, Publishing unit count, Section count, Inner/sub concern count, Indifferent concern count, Module point cut count, Module count, Module attribute count, Operation count, Comment count, Reused comment count, Media duration, Diffusion cut count, Concern module count, Concern operation count, Anchor count
		Functionality	High feature count, Low feature count, Reused high feature count, Reused low feature count, Web objects, Common Software Measurement International Consortium, International Function Point Users Group, Object-Oriented Heuristic Function Points, Object-Oriented Function Points, Use case count, Feature count, Data Web points
		Object-oriented	Cohesion, Class coupling, Concern coupling
		Complexity	Connectivity density, Cyclomatic complexity, Model collection complexity, Model association complexity, Model link complexity, Page complexity, Component complexity, Total complexity, Adaptation complexity, New complexity, Data usage

			complexity, Data flow complexity, Cohesion complexity, Interface complexity, Control flow complexity, Class complexity, Layout complexity, Input complexity, Output complexity
	Cost Driver	Product	Type, Stratum, Compactness, Structure, Architecture, Integration with legacy systems, Concurrency level, Processing requirements, Database size, Requirements volatility level, Requirements novelty level, Reliability level, Maintainability level, Time efficiency level, Memory efficiency level, Portability level, Scalability level, Quality level, Usability level, Readability level, Security level, Installability level, Modularity level, Flexibility level, Testability level, Accessibility level, Trainability level, Innovation level, Technical factors, Storage constraint, Reusability level, Robustness level, Design volatility, Experience level, Requirements clarity level
		Client	Availability level, IT literacy, Mapped workflows, Personality
		Development Company	SPI program, Metrics' program, Number of projects in parallel, Software reuse
		Project	Documentation level, Number of programming languages, Type, Process efficiency level, Project management level, Infrastructure, Development restriction, Time restriction, Risk level, Rapid app development, Operational mode, Resource level, Lessons learned repository
		Team	Domain experience level, Team size, Deployment platform experience level, Team capability, Programming language experience level, Tool experience level, Communication level, Software development experience, Work Team level, Stability level, Motivation level, Focus factor, Tool experience level, OO experience level, In-house experience
		Technology	Authoring tool type, Productivity level, Novelty level, Platform volatility level, Difficulty level, Platform support level

Table 9: Table displaying all categories and characteristics of the taxonomy proposed by Britto et al. [28]

Britto et al. [28] focuses primarily on size measures and cost factors with particular emphasis on Web applications, containing the highest number of characteristics (160) among all taxonomies. It presents in total 13 categories and 160 characteristics. The classification reveals a close balance between the number of size metrics (83) and cost drivers (77). No additional categories are included, such as estimation techniques and wider contextual factors.

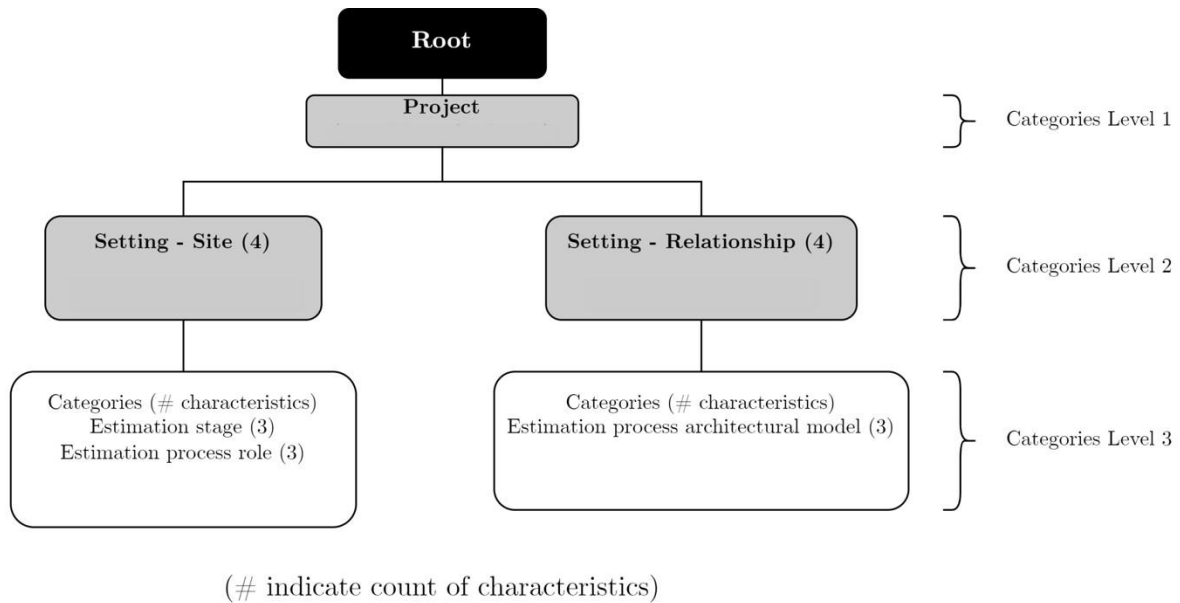


Figure 6: Structure of the taxonomy for SEE proposed by Britto et al. [27]

Categories and characteristics from Britto et al. [27]

Taxonomy	Categories Level 1	Categories Level 2	Categories Level 3	Characteristics
Britto et al. [27]	Project	Setting - Site	-	Location
				Legal Entity
				Geographic Distance
				Temporal Distance
			Estimation stage	Early, Early & Late, Late
			Estimation process role	Estimator, Estimator & Provider, Provider
		Setting - Relationship	-	Location
				Legal Entity
				Geographic Distance
				Temporal Distance

			Estimation process architectural model	Centralized, Distributed, Semi-distributed
--	--	--	--	--

Table 10: Table displaying all categories and characteristics of the taxonomy for SEE proposed by Britto et al. [27]

Britto et al. [27] focuses solely upon project-related factors such as *the estimation process architectural model (Centralized, Distributed, Semi-distributed)* and *at which stage estimates are prepared by each of the participating sites (Early, Early & Late, Late)*. No categories relating to size metrics, cost factors or estimation techniques were included. It presents in total six categories and 17 characteristics.

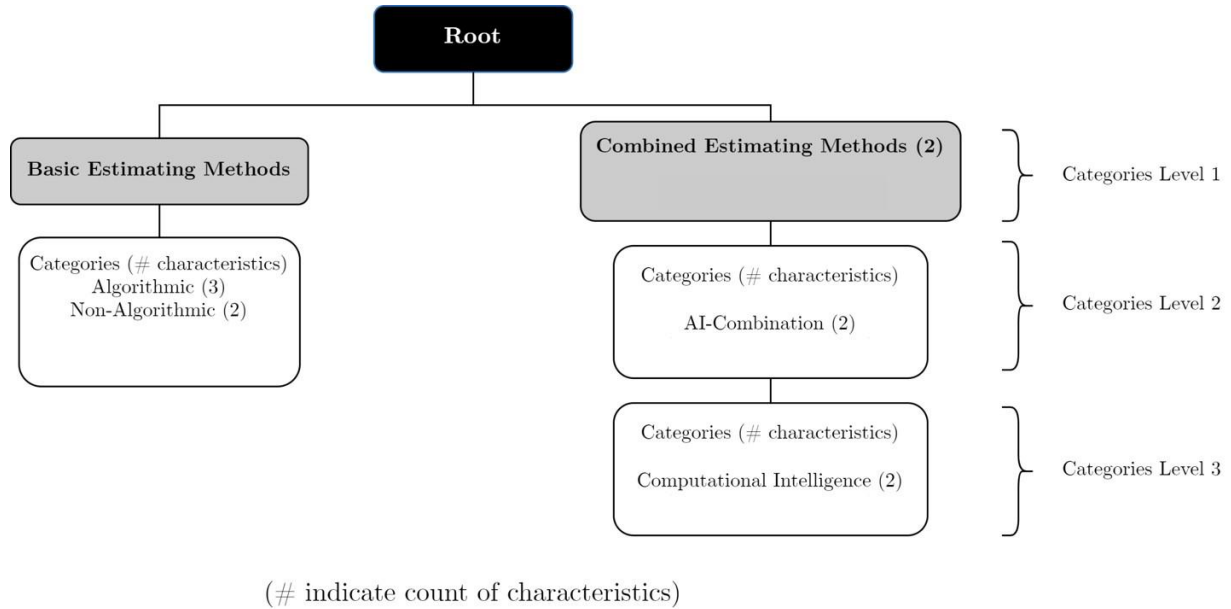


Figure 7: Structure of the taxonomy for SEE proposed by Dashti & Gandomani [73]

Categories and characteristics from Dashti & Gandomani [73]

Taxonomy	Categories Level 1	Categories Level 2	Characteristics
Dashti & Gandomani [73]	Basic Estimating Methods	Algorithmic	Constructive Cost Model
			Software Life Cycle Management
			Software Evaluation and Estimation for Risk
		Non-Algorithmic	Expert Judgment
			Analogy-Based
	Combined Estimating Methods	-	Basic-Combination
		AI-Combination	Fuzzy Logic
			Artificial Neural Networks
			Swarm

			Evolutionary
		-	AI-Combined hybrid

Table 11: Table displaying all categories and characteristics of the taxonomy for SEE proposed by Dashti & Gandomani [73]

Dashti & Gandomani [73] explores estimation techniques exclusively, focusing on methodological aspects rather than predictive factors. It presents in total six categories and 12 characteristics. The taxonomy classification reveals elements that are neither size measures nor cost factors, instead focusing on *estimation techniques at a higher level of granularity (algorithmic, non-algorithmic)* and then its specialisation into a more detailed view that includes *AI-based computational intelligence methods - Artificial Neural Network Swarm algorithms*.

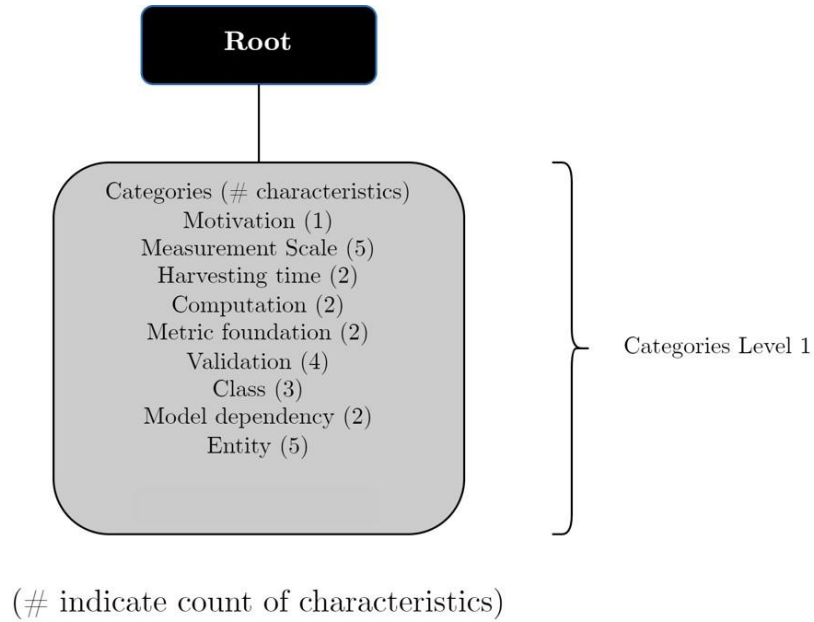


Figure 8: Structure of the taxonomy for SEE proposed by Mendes et al. [29]

Categories and characteristics from Mendes et al. [29]

Taxonomy	Categories Level 1	Characteristics
Mendes et al. [29]	Motivation	Motivation
	Harvesting time	Early size metric, Late size metric
	Metric foundation	Problem-oriented metric, Solution-oriented metric
	Class	Length, Functionality, Complexity
	Entity	Web hypermedia application, Web software application, Web application, Media, Program/Script
	Measurement Scale	Nominal, Ordinal, Interval, Ratio, Absolute
	Computation	Direct, Indirect

	Validation	Validated Empirically, Validated Theoretically, Both, None
	Model dependency	Specific, Nonspecific

Table 12: Table displaying all categories and characteristics of the taxonomy for SEE proposed by Mendes et al. [29]

Mendes et al. [29] specializes exclusively in size measures for Web hypermedia and software applications, representing the most focused taxonomy in terms of scope. It presents in total nine categories and 26 characteristics. The detailed classification shows that this taxonomy does not include categories relating to cost factors, estimation techniques and contextual factors; instead, it solely provides domain-specific coverage of Web application metrics.

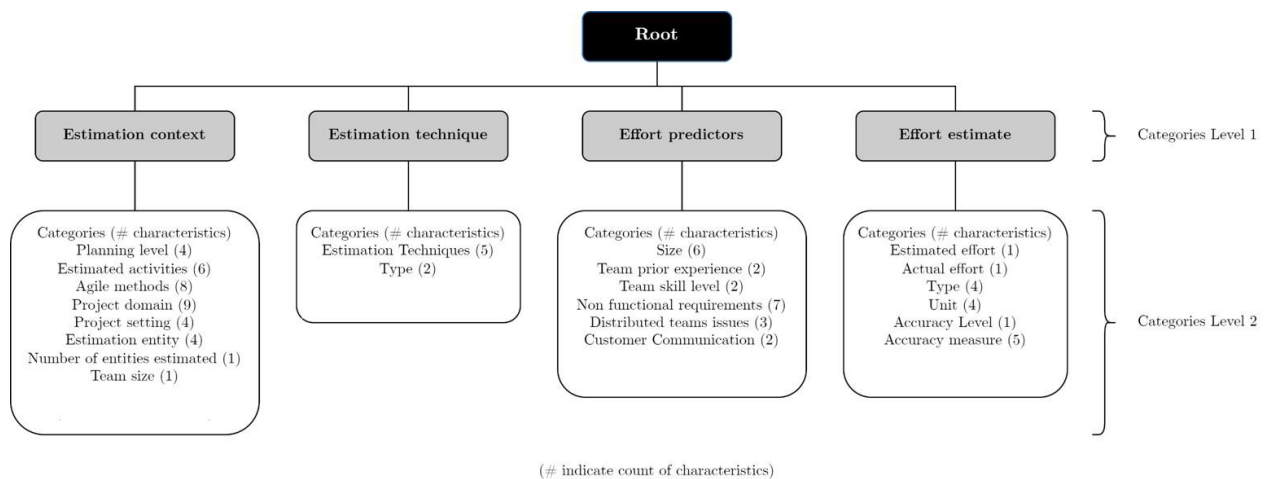


Figure 9: Structure of the taxonomy for SEE proposed by Usman et al. [58]

Categories and characteristics from Usman et al. [58]

Taxonomy	Categories Level 1	Categories Level 2	Characteristics
Usman et al. [58]	Estimation context	Planning level	Release, Sprint, Daily, Bidding
		Estimated activities	Analysis, Design, Implementation, Testing, Maintenance, Estimated activities.All
		Agile methods	Extreme Programming, Scrum, Customized Extreme Programming, Customized Scrum, Dynamic Systems Development Method, Crystal, Feature-Driven Development, Kanban
		Project domain	Communications industry, Transportation, Financial, Education, Health, Retail/Wholesale, Manufacturing, Government/Military, Project domain.Other

		Project setting	Co-located, Distributed: Close Onshore, Distributed: Distant Onshore, Distributed: Near Offshore, Distributed: Far Offshore
		Estimation entity	User story, Task, Use case, Estimation entity.Other
		Number of entities estimated	Number of entities estimated.Value
		Team size	No. of team members
	Estimation technique	Estimation Techniques	Planning Poker, Expert Judgement, Analogy, Use case points method, Estimation Techniques.Other
		Type	Type.Single, Type.Group
	Effort predictors	Size	Story points, User case points, Function points, Size.Other, Size.Not used, Considered without any metric
		Team's prior experience	Team's prior experience.Considered, Team's prior experience.Not Considered
		Team's skill level	Team's skill level.Considered, Team's skill level.Not Considered
		Non-functional requirements	Performance, Security, Availability, Reliability, Maintainability, Non-functional requirements.Other, Non-functional requirements. Not considered
		Distributed teams' issues	Distributed teams' issues.Considered, Distributed teams' issues.Not Considered, Distributed teams' issues.Not applicable
		Customer Communication	Customer Communication.Considered, Customer Communication.Not Considered
	Effort estimate	Estimated effort	Estimate value(s)
		Actual effort	Actual effort.Value
		Type	Point type, Three point, Distribution, Type.Other
		Unit	House/days, Pair days, Ideal hours, Unit.Other
		Accuracy Level	Accuracy Level.Value
		Accuracy measure	Mean Magnitude of Relative Error, Median Magnitude of Relative Error, Bias of Relative Error, Accuracy measure.Other, Accuracy measure.Not used

Table 13: Table displaying all categories and characteristics of the taxonomy for SEE proposed by Usman et al. [58]

Usman et al. [58] similarly adopts a wide coverage, covering size measures (*story points, function points*), cost factors (*team experience, project setting*), estimation techniques, and estimation context, with particular emphasis on Agile methodologies. It presents in total 26 categories and 83 characteristics.

Overall, the classification analysis showed that the taxonomies for SEE detailed here varied in relation to which aspects of an effort estimation phenomenon were included. Usman et al. [58] and Bajta & Idri [72] included effort estimation techniques, effort predictors and contextual factors; Britto et al. [28] and Mendes et al. [29] included only effort predictors; Dashti & Gandomani [73] included only effort estimation techniques and finally, Britto et al. [27] included only contextual factors.

Key take-aways:

1. The six taxonomies differ substantially in structure, ranging from 1 to 6 dimensions, 6 to 26 categories, and 12 to 164 characteristics, with no common design standard or shared structural framework across taxonomies.
2. They also vary in their content focus: Bajta & Idri [72] and Usman et al. [58] integrate both effort predictors (size measures and cost factors), estimation techniques and estimation context, providing broader coverage. In contrast, Britto et al. [27] and [28] focus solely on cost factors; Mendes et al. [29] exclusively on size measures; and Dashti & Gandomani [73] exclusively on estimation techniques.

5.2.2 RQ2: What are the software application domains targeted by the taxonomies identified in this SLR?

The classification of software application domains used herein, as previously stated, is the one proposed by Pressman & Maxim [44], where seven domains were identified, as follows: System software, Application software, Engineering/scientific software, Embedded software, Product-line software, Web/mobile applications, Artificial intelligence software. Their brief description is given in sub-section 2.5. The classification was applied to studies by the second author and validated by the first author.

Table 14 shows that the software application domain ‘Application software’ is targeted by Usman et al. [58], Bajta & Idri [72] and Britto et al. [27]; ‘Web/mobile applications’ is the focus by Mendes et al. [29], and Britto et al. [28]. Dashti & Gandomani [73] do not specify the type(s) of software application domain(s) to which their proposed taxonomies relate to. Finally, Usman et al. [58], Britto et al. [27] and Bajta & Idri [72] proposed taxonomies geared towards ‘Embedded software’.

None of these taxonomies includes characteristics that account for more recent software development contexts, such as those using Artificial Intelligence [77], and sustainability [78]. Therefore, we do believe that such results highlight a gap in the current classification of knowledge pertaining to SEE, given that the software domains of Application software, Embedded software and Web/Mobile software, represent only a fraction of the diverse array of software solutions being developed [79][80][81].

An avenue for future work could be to either extend an existing taxonomy for SEE, or to propose a new taxonomy, catering for the additional software application domains in Pressman & Maxim’s classification [44].

Taxonomy	What is the domain?
Bajta & Idri [72]	Application software, Embedded software
Britto et al. [28]	Web/Mobile applications
Britto et al. [27]	Application software, Embedded software
Dashti & Gandomani [73]	Not specified
Mendes et al. [29]	Web/Mobile applications
Usman et al. [58]	Application software, Embedded software

Table 14: Summary Table with the evidence used to answer RQ2

Key take-aways:

1. Among the seven software application domains proposed by Pressman and Maxim [44], only three (application software, Web/mobile applications, and embedded software) are documented in the taxonomies for SEE identified by our SLR. We believe that the other four remaining application domains have not been accounted for by the existing taxonomies for SEE included in our SLR. They are system software, engineering/scientific software, product-line software, and artificial intelligence software.
2. The likely gap in coverage of software application domains by the existing taxonomies for SEE supports the need for further investigation in this area, so to propose/extend a taxonomy for SEE that includes all seven software application domains.
3. Note that our results and suggestions relating to gaps in existing taxonomies for SEE contain some level of uncertainty given that one study did not disclose the software application domain(s) target of their taxonomies for SEE.

5.2.3 RQ3: (if applicable) Which of the taxonomies identified in this SLR are an evolution of previously published ones? Which aspects were changed?

Our results (Table 15) show that, except for Britto et al. [27], all of the identified taxonomies for SEE were developed independently, building on prior literature but not directly evolving from previous taxonomies. Britto et al. [27] is the only exception, as it used earlier work by Šmite et al. [71] and Britto et al. [59] relating to a taxonomy for global software engineering (GSE), in order to build a taxonomy for SEE within the context of GSE.

Taxonomy	Is the taxonomy a revision of a previously published one?	Which taxonomy is it a revision of?	What aspects were changed or added?
Bajta & Idri [72]	No	-	-
Britto et al. [28]	No	-	-
Britto et al. [27]	Yes	Šmite et al. [71], Britto et al. [59]	The proposed taxonomy consists of Šmite et al.'s [71] and Britto et al.'s [59] original dimensions (categories) and characteristics. The new specialized taxonomy added

			the following new dimensions (categories) and characteristics: Estimation stage: Early, Late, Early & Late Estimation process role: Estimator, Provider, Estimator & Provider Estimation process architectural model: Centralized, Distributed, Semi-distributed
Dashti & Gandomani [73]	No	-	-
Mendes et al. [29]	No	-	-
Usman et al. [58]	No	-	-

Table 15: Summary Table with the evidence used to answer RQ3

Key take-aways:

1. Five taxonomies for SEE were developed independently, focusing on specific areas of SEE rather than evolving from an existing framework.
2. Britto et al. [27] is an exception, adopting an integrative approach to defining a more specialized taxonomy targeted at SEE in global software development (GSD), using as basis two previous studies that proposed taxonomies for Global Software Development (Šmite et al.'s [71] and Britto et al.'s [59]).

5.2.4 RQ4: What are the similarities and differences across the identified taxonomies for SEE in terms of their structure and content?

This research question aimed to identify the categories and characteristics sharing a similarity of at least 90% across taxonomies (Table 16 and Table 17). Such analysis was carried out using jina-embeddings-v3 [54], and the justification for this choice is given in Section 2.6.

Note that several safeguards were employed to mitigate the risk of artificial structure in our embedding-based analysis, as follows: (i) the 90% similarity threshold is deliberately conservative, ensuring that only very high-confidence matches are reported; (ii) all matched pairs are explicitly listed in Table 18 to support readers' evaluation and verification; (iii) the methodology follows the established framework of Unterkalmsteiner & Abdeen [40]; and (iv) all the matched pairs were validated independently by another two SEE researchers who have also proposed Taxonomies for SEE. They were contacted separately and provided with the matched pairs for assessment. Both returned their feedback also separately, and their independent assessments were that all pairs represented both semantic & cognitive similarity.

The following main aspects were observed:

Bajta & Idri [72] shares multiple categories with Usman et al. [58], including *Accuracy Measure*, *Estimation Technique*, *Project Domain*, *Project Setting*, and *Team Size*; also shared characteristics such as

Analysis, Availability, Design, Implementation, Maintainability, Maintenance, Performance, Reliability, Security, and Testing; it also shares characteristics such as *Expert Judgment* and *Constructive Cost Model* with Dashti & Gandomani [73], and the characteristic *Temporal Distance* with Britto et al. [27].

Usman et al. [58] shows the most extensive overlap with other taxonomies for SEE because it shares categories (*Accuracy Measure, Estimation Technique, Project Domain, Project Setting, Team Size*) with Bajta & Idri [72] and numerous characteristics (*Analysis, Availability, Design, Implementation, Maintainability, Maintenance, Performance, Reliability, Security, and Testing*) with Bajta & Idri [72] and Britto et al. [28]. The taxonomy also shares *Analogy-based characteristics* with Dashti & Gandomani [73], indicating a similar understanding of estimation methodologies across different research contexts.

Britto et al. [27] shares the same *Project* category with Britto et al. [28]; the *Estimation Process Architectural Model* with Dashti & Gandomani [73]; *Geographic Distance* and *Temporal Distance* with Bajta & Idri [72]; and *Legal Entity* with Dashti & Gandomani [73]. These common elements represent shared focus towards distributed development aspects and process-related factors.

Britto et al. [28] shares Size Metrics categories with Mendes et al. [29]; the *Project* category with Britto et al. [27], along with characteristics such as *Availability Level, Entity Count, Maintainability Level, Motivation Level, New Complexity, Reliability Level, Risk Level, Team Size, and Total Complexity* with Bajta & Idri [72], Usman et al. [58] and Mendes et al. [29]. Such overlap suggests a good level of consensus on technical complexity factors and organizational aspects within Web development contexts.

Mendes et al. [29] shares Size Metrics categories and characteristics with Britto et al. [28], demonstrating some agreement on Web size measures within specialized Web application domains.

Dashti & Gandomani [73] shares *Estimation Process Architectural Model* with Britto et al. [27], *Expert Judgment* and *Constructive Cost Model* with Bajta & Idri [72], *Legal Entity* with Britto et al. [27], and *Analogy-Based* characteristics with Usman et al. [58].

The semantic similarity analysis revealed that 36 categories (44.4%) and only 62 characteristics (17%) are conceptually shared across the different taxonomies (Table 16 and Table 17). When jointly considering all the six different taxonomies for SEE, we had a total of 365 distinct characteristics, of which only 62 had some level of overlap. These results may suggest a high level of fragmentation regarding the characteristics that have been included in the abovementioned taxonomies, and some level of fragmentation in relation to categories.

Is knowledge fragmentation always undesirable? Not necessarily; however, it is important to assess whether the fragmentation identified when comparing the different taxonomies for SEE is useful or not.

When answering RQ2, we showed in Table 14 that five of the six taxonomies identified in our SLR catered to either the domains of Application & Embedded software (three taxonomies) or to the domains of Web/Mobile applications (two taxonomies). One taxonomy did not specify to which software application domain(s) their taxonomy was related to.

The two taxonomies that relate to Web/Mobile applications share nine characteristics; however, one has 26 characteristics (Mendes et al. [29]) and the other has 160 characteristics (Britto et al. [28]). The former was published in 2005, the latter in 2017. We believe that such fragmentation should be addressed, and this could be done, for example, via validating the combined set of characteristics with both industry and SEE researchers so to determine the subset of characteristics that should be kept, and then to use the results from such validation to build a single taxonomy. In this way, there would be one up to date taxonomy for SEE that relates to Web/Mobile applications.

In relation to the three taxonomies that relate to Application & Embedded software, as shown in Table 17, Bajta & Idri [72] share two characteristics with Britto et al. [27] and 33 with Usman et al. [58]; Britto et al. [27] shares two characteristics with Usman et al. [58]. They were published respectively in 2019, 2016 and 2017; so not far apart from one another. The number of characteristics in their taxonomies are respectively 63, 17 and 83. Bajta & Idri [72] and Usman et al. [58] proposed taxonomies that represent effort estimation techniques, predictors and contextual factors, whereas the taxonomy proposed by Britto et al. [27] has characteristics that represent only contextual factors. We believe that such

fragmentation should also be investigated to assess whether it would be beneficial to both practice and research to merge these taxonomies and/or to validate the combined set of characteristics with both industry and SEE researchers.

The validation of such taxonomies with both industry and SEE researchers is one of the goals of our future work. We also aim to assess whether these two communities (industry and research) would favour the development of a single unified SEE Taxonomy, or to have distinct Taxonomies for SEE.

Set 1	Set 2	Shared Categories Count*	Categories from Set 1	Categories from Set 2
Bajta & Idri [72]	Britto et al. [27]	3	Estimation Technique, Project Activities, Project Setting	Estimation Stage, Project
Bajta & Idri [72]	Britto et al. [28]	6	Product Requirement, Product Size, Project Activities, Project Setting, Team Experience, Team Structure	Product, Project, Team
Bajta & Idri [72]	Dahsti & Gandomani [73]	2	Estimation Technique	Basic Estimating Methods, Combined Estimating Methods
Bajta & Idri [72]	Usman et al. [58]	16	Accuracy Measure, Cost Estimation Context, Estimation Technique, Planning, Planning Approaches, Product Requirement, Product Size, Project Activities, Project Domain, Project Setting, Team Experience, Team Size, Team Structure	Accuracy Level, Accuracy Measure, Estimation Context, Estimation Entity, Estimation Technique, Non-Functional Requirements, Planning Level, Project Domain, Project Setting, Size, Team Size, Team's Prior Experience
Britto et al. [27]	Britto et al. [28]	1	Project	Project
Britto et al. [27]	Usman et al. [58]	3	Estimation Stage, Project	Estimation Entity, Estimation Technique, Project Setting
Britto et al. [28]	Usman et al. [58]	3	Project, Size Metric, Team	Project Setting, Size, Team Size
Dahsti & Gandomani [73]	Usman et al. [58]	2	Basic Estimating Methods, Combined Estimating Methods	Estimation Technique

Table 16: Table showing the shared categories between taxonomies, used as additional evidence towards answering RQ4.

Set 1	Set 2	Shared Characteristic Count*	Characteristics from Set 1	Characteristics from Set 2
Bajta & Idri [72]	Britto et al. [27]	2	Estimated Value, Temporal Distance	Estimator, Temporal Distance
Bajta & Idri [72]	Britto et al. [28]	6	Availability, Maintainability, Number Of Team Members, Reliability, Risk, Security	Availability Level, Maintainability Level, Reliability Level, Risk Level, Security Level, Team Size
Bajta & Idri [72]	Dahsti & Gandomani [73]	2	Cocoma, Expert Judgment	Cocoma, Expert Judgment
Bajta & Idri [72]	Usman et al. [58]	33	Analysis, Availability, Bidding, Design, Distant Onshore, Estimated Value, Expert Judgment, Finance, Healthcare, Implementation, Maintainability, Maintenance, Near Offshore, Not Considered, Number Of Team Members, Performance, Reliability, Security, Statistical Analysis, Testing, Value	Analysis, Availability, Bidding, Close Onshore, Considered, Design, Distant Onshore, Estimate Value(S), Expert Judgment, Far Offshore, Financial, Health, Implementation, Maintainability, Maintenance, Near Offshore, No. Of Team Members, Not Applicable, Not Considered, Performance, Reliability, Security, Testing, Value
Britto et al. [27]	Usman et al. [58]	2	Distributed, Estimator	Distribution, Estimate Value(S)
Britto et al. [28]	Mendes et al. [29]	9	Class Complexity, Component Complexity, Cyclomatic Complexity, Input Complexity, Media Allocation, Media Count, New Complexity, Output Complexity, Total Complexity	Complexity, Media
Britto et al. [28]	Usman et al. [58]	5	Availability Level, Maintainability Level, Reliability Level, Security Level, Team Size	Availability, Maintainability, No. Of Team Members, Reliability, Security
Dahsti & Gandomani [73]	Usman et al. [58]	2	Analogy Base, Expert Judgment	Analogy, Expert Judgment
Mendes et al. [29]	Usman et al. [58]	1	Other	Other

Table 17: Table showing the shared characteristics between taxonomies.

Primary study	Word Embedding Relation (Embeddings from different taxonomies sharing 90% similarity)
Bajta & Idri [73]	analysis --> analysis availability --> availability availability --> availability level constructive cost model --> constructive cost model design --> design estimated value --> estimate value(s) expert judgment --> expert judgment expert judgment --> expert judgement geographical distance --> geographic distance implementation --> implementation maintainability --> maintainability maintainability --> maintainability level maintenance --> maintenance number of team members --> team size performance --> performance reliability --> reliability reliability --> reliability level risk --> risk level security --> security team experience.considered --> team's prior experience.considered team experience.considered --> team's prior experience.not considered team experience.not considered --> team's prior experience.not considered temporal distance --> temporal distance testing --> testing
Britto et al. [28]	availability level --> availability availability level --> availability entity count --> number of entities estimated maintainability level --> maintainability maintainability level --> maintainability motivation level --> motivation new complexity --> complexity reliability level --> reliability reliability level --> reliability risk level --> risk team size --> number of team members total complexity --> complexity
Britto et al. [27]	geographic distance --> geographical distance legal entity --> legal entity temporal distance --> temporal distance
Dashti & Gandomani [74]	analogy-based --> analogy constructive cost model --> constructive cost model expert judgment --> expert judgment expert judgment --> expert judgement legal entity --> legal entity
Mendes et al. [29]	complexity --> new complexity complexity --> total complexity motivation --> motivation level
Usman et al. [59]	analogy --> analogy-based analysis --> analysis availability --> availability availability --> availability level design --> design estimate value(s) --> estimated value expert judgement --> expert judgment expert judgement --> expert judgment implementation --> implementation maintainability --> maintainability maintainability --> maintainability level maintenance --> maintenance number of entities estimated --> entity count performance --> performance reliability --> reliability reliability --> reliability level security --> security team's prior experience.considered --> team experience.considered team's prior experience.not considered --> team experience.considered team's prior experience.not considered --> team experience.not considered testing --> testing

Table 18: Table showing through colours, how one taxonomy's characteristic relates to another taxonomy's characteristic. Displaying only the word embeddings with a semantic similarity of over 90%. The

characteristics are coloured according to the primary study it originates from, as follows: In dark green - Bajta & Idri [72], pink - Britto et al. [28], orange - Britto et al. [27], light green - Dashti & Gandomani [73], yellow - Mendes et al. [29], grey - Usman et al. [58].

Key take-aways:

1. Only 36 categories (44.4%) and 62 characteristics (17%) were identified via our semantic similarity technique to be similar, based upon a 90% semantic similarity threshold.
2. The remaining 45 categories and 303 characteristics (83%) are unique to individual taxonomies.
3. Areas of convergence exist in estimation techniques (e.g., expert judgment), quality attributes (e.g., reliability, maintainability, availability), and team-related factors (e.g., team size, team experience).
4. Some of the terms labelled in the same way carried different contextual meanings across taxonomies (e.g., reliability as code stability versus system uptime), suggesting that at least a common terminology should be shared across taxonomies.
5. There is knowledge fragmentation between taxonomies catering to the same software application domain(s); therefore, we believe that such situation merits further investigation, and suggested avenues for future work were given.

6 Discussion

6.1 Quality assessment framework employed to assess the existing taxonomies

The quality attributes proposed by Unterkalmsteiner & Abdeen [40] presented challenges when applied to taxonomies in SEE. One issue arose when computing *robustness*, as it depends on comparing the semantic similarity of a taxonomy's characteristics (leaves). While the proposal in [40] to compute robustness may look promising at first, it is difficult to implement in practice. To implement an algorithm¹ to automatically measure robustness, we had to first define a suitable input method to gather information about the different categories and characteristics. This was complex due to the variability in the taxonomies' design. Furthermore, calculating the semantic similarity also presented its challenges because many categories (inner nodes) and characteristics (leaves) are labelled using multiple words or sentences, instead of single terms.

Furthermore, Unterkalmsteiner & Abdeen [40] suggest using a custom doc2vec model, which introduces two challenges: 1) reproducibility of results, and 2) the complexity of converting taxonomies into machine-readable code. The robustness measure $R(t)$ is directly influenced by the quality of the semantic model used, meaning the accuracy of results is dependent on the effectiveness of the chosen model. Perhaps the use of a semantic model trained specifically on SE concepts and sentences may lead to improved results, compared to using a general semantic model. Therefore, future work in this area is, in our view, still needed.

In relation to measuring the taxonomies' *extensibility*, most of the primary studies included in this SLR only mentioned briefly anything that would relate to a process for updating the taxonomy. Such lack of detailed reporting was also observed when measuring the attributes *explanatory* and *mutual exclusivity*. Therefore, our recommendation is that studies that propose taxonomies to also include detailed reporting that would enable others to measure the abovementioned attributes.

One of the taxonomies included herein, by Usman et al. [58], included numerous characteristics labelled "yes/no". This, in our view, turned the taxonomy into a pseudo checklist, which is not in our opinion what a taxonomy is meant to be used for.

¹ Available at <https://github.com/marcsol96/slrtaxse>

Furthermore, several characteristics are only interpretable in the context of their parent categories. Examples include terms such as “none”, “value”, and “type”. Enhancing these characteristics by simply linking them to their parent categories (e.g., “Project Activities.Other”) significantly improves the taxonomy’s robustness, as shown in Table 19. However, although their robustness is improved, the use of labels such as “none” or “other” does not truly specify any specific size measures, cost factors, or effort estimation techniques and methodologies, and therefore they should not be present in any taxonomy.

Taxonomy	Original R(t)	Modified characteristics R(t)
Bajta & Idri [72]	0.8955	0.8959
Britto et al. [28]	0.8956	0.8960
Britto et al. [27]	0.8733	0.9310
Dashti & Gandomani [73] *	0.8417	0.8278
Mendes et al. [29]	0.9094	0.9371
Usman et al. [58]	0.8759	0.8726

Table 19: Comparison of R(T) when introducing a prefix to ambiguous characteristics. * R(t) decreases

6.2 Additional observations about six taxonomies for SEE

Many differences were observed between the six taxonomies for SEE included in the SLR and we could try to somehow speculate the reasons; however, we chose to do a post hoc analysis, summarised in Table 25 (Appendix C). We can clearly see that both Bajta & Idri [72] and Dashti & Gandomani [73] were oblivious about previously published and relevant studies detailing other taxonomies for SEE. Usman et al. [58] targets at the same software application domains as Britto et al. [27] and is co-author in Britto et al. [28] so there was obviously awareness of previous work. Britto et al. [28] referenced Mendes et al. [29] as both shared the same software application domains and had one co-author in common (Mendes). Except for Mendes et al. [29], which was the first study to be published on taxonomies for SEE, all the other five studies failed either completely (Bajta & Idri [72] and Dashti & Gandomani [73]) or partially (Britto et al. [27], Britto et al. [28] and Usman et al. [58]) in referencing related work.

The differences between Britto et al. [28] and Mendes et al. [29] were intentional as they even shared a co-author (Mendes). The differences between Usman et al. [58] and Britto et al. [27] must also have been intentional as both Britto and Usman work at the same Higher Education Institution. However, as for Bajta & Idri [72] and Dashti & Gandomani [73], we can only speculate their reasons for proposing different taxonomies as they showed no awareness about previous work in taxonomies for SEE. Therefore, we prefer not to add any further comments.

Furthermore, we have also identified issues, such as: (i) the existence of identical terms that present different contextual meanings across taxonomies. For example, a term such as “reliability” may refer to code stability in an agile development context, while in a global software engineering context it may refer to system uptime; (ii) lack of explicitly capturing contextual variations with clear definitions; (iii) no use of standardized terminology with context qualifiers whenever applicable; and (iv) no provision of a common foundation while preserving legitimate contextual distinctions. Such foundations could establish, for example, the core categories that a SEE Taxonomy should present (e.g. effort predictors, effort estimation technique, accuracy measure).

In interpreting these findings, it is important to distinguish between two types of gaps. The first type relates to genuine coverage gaps, where entire software application domains (such as AI software, engineering/scientific software, product-line software, and system software) have no taxonomy representation at all. The second type relates to possible terminological inconsistency, where the same concepts may be named differently across taxonomies targeting overlapping domains; this seems to be the case and suggested by the fact that 303 of 365 characteristics (83%) are unique to individual taxonomies at a 90% similarity threshold. Both types of gaps open avenues for future work where, for example, an existing taxonomy for SEE may be extended, or an entirely new Taxonomy for SEE be proposed to cater for the additional software application domains in Pressman & Maxim’s classification [44]. In addition, surveys may be conducted with both industry and SEE researchers so to obtain feedback on aspects such as: i) the set of characteristics deemed suitable to be part of a taxonomy for SEE; and ii)

their views on the use/development of a single holistic Taxonomy for SEE or to settle for specialised Taxonomies.

6.3 Comparison of our work with Vera et al.

This Sub-section provides a more detailed comparison between this SLR and the technical report by Vera et al. [31]. Such comparison revolves mostly around methodological flaws and lack of peer review. Note that despite the numerous shortcomings in the technical report, which are detailed next, we have included it herein for the sake of transparency and to position our SLR as genuinely the first that presents an evidence synthesis of taxonomies for SEE.

The technical report presents a literature survey, which they argue to have been conducted according to Kitchenham & Charters' SLR guidelines for SE [57]. However, we have identified numerous inconsistencies, which are presented in Table 20. We also added an additional column to Table 20 so to show how we have applied the SLR guidelines to our SLR. Such solution also helps to clearly identify the distinctions between the technical report and our SLR. We have also employed other guidelines; however, for the sake of a proper comparison, only the ones that were claimed to be used in the technical report are included in Table 20.

The formulation of research questions is one of the core components of any SLR. Vera et al.'s RQ1 seeks to identify "main" taxonomies without operationalizing this concept, while RQ2 pursues "relevant" classification criteria without defining relevance parameters. This ambiguity cascades through their entire methodology, affecting the search strategy, selection criteria, and synthesis approaches. In contrast, our SLR complies with the guidelines via presenting and justifying the use of four RQs, and establishing clear, measurable criteria for taxonomy evaluation, which includes explicit definitions of taxonomy content, consistency, and practical utility. In addition to a quality assessment, we also carried out & validated a semantic analysis of the current taxonomies in SEE to identify inconsistencies and overlaps between them.

The use of automated "robots" for search and selection processes, while potentially efficient, introduces opacity that undermines reproducibility. Vera et al.'s reliance on black-box automation without providing implementation details or validation procedures represents a significant methodological concern. Our approach maintains transparency through detailed reporting of search strategies, manual validation, public availability of automated processes, and provision of all search protocols and selection criteria. We conducted the selection of papers transparently and disclosing what each author's role was in that step, with the replication package being available online¹.

Vera et al. did not include any primary study that proposed a Taxonomy for SEE. What they did was to propose themselves taxonomies for each of the primary studies they included and to use their self-created taxonomies as if they had been originally described in the primary studies. Such methodological flaw compromises significantly the validity of their findings as they did not apply a rigorous methodology when identifying primary studies.

Finally, Vera et al. did not include any of the six taxonomies for SEE that we included, as none of their primary studies proposed/used a taxonomy for SEE.

SLR Stages as per Kitchenham & Charters' Guidelines	Vera et al. technical report	Our SLR
PLANNING THE REVIEW		
5.1 The need for a systematic review	Justified the need as to identify software estimation models for their own use.	Section 1 clearly establishes the need. No comprehensive evidence synthesis of existing taxonomies in SEE has been carried out to date
5.2 Commissioning a Review	Not a commissioned review	Danish National Research Foundation Chair Grant DNRF181

5.3 Specifying the Research Question(s)	Two RQs that are not clearly stated.	Four clear RQs with corresponding justification (see Table 2)
5.4 Developing a Review Protocol	No mentioning of a review protocol prepared prior to starting the survey.	A protocol was prepared and it is presented in Section 4 according to three guidelines: Kitchenham & Charters, SEGRESS, and PRISMA guidelines
5.5 Evaluating the Review Protocol	No protocol evaluation mentioned	The two authors developed and reviewed the protocol
CONDUCTING THE REVIEW		
6.1 Identification of Research		
- Search strategy	Search strategy described	Comprehensive search using Scopus (356 articles) and Google Scholar (3125 articles). Keywords derived from literature; synonyms identified
- Electronic sources	Four sources: IEEE Xplore, ACM Digital Library, Science Direct, Springer Link Missed key databases like Scopus and Google Scholar	Scopus and Google Scholar, indexing for IEEE, ACM, among others.
- Other sources	Not mentioned	Grey literature excluded.
- Search documentation	Search string composition not given; Search strings provided, databases specified, dates not documented. Search carried out by an automated bot without transparency.	Search string composition not given; Search strings provided, databases specified, dates documented, process described in detail
6.2 Study Selection		
- Selection criteria	Vague and broad inclusion criteria missing essential aspects (language, publication type)	Clear inclusion (IC1-IC4) and exclusion (EC1-EC4) criteria provided
- Selection process	The lead author rated ALL studies (massive bias risk) Advisors "played the other roles" - what roles exactly? "Robot" automatically applied QS2 filter - no human verification Studies were not independently assessed by 2+ reviewers "relevant" vs "irrelevant" vs "uncertain" used but no decision tree	Multi-stage process: both authors screened first 1000, remaining 2125 by second author, disagreements discussed

- Multiple reviewers	No independent assessment - lead author did all initial ratings, advisors only involved in conflicts	
- Kappa statistics	No Kappa statistics reported	Only 5 disagreements from first 1000 papers needed discussion; consensus reached Cohen’s κ = 0.83, observed agreement = 99.6%
6.3 Study Quality Assessment		
- Quality assessment	Publication venue quality (QA1) Clarity of goals (QA2) Evidence support (QA3) Industry usefulness (QA4)	Comprehensive framework using Unterkalmsteiner & Abdeen for taxonomy quality + Wieringa et al. for paper classification.
- Quality instruments	Quality assessment checklist with Yes/Partially/No scoring.	Quality scores calculated (R(T), C(T)), quality data used in analysis and synthesis
- Quality threshold	Used 50% cutoff	N/A
6.4 Data Extraction		
- Extraction forms	Used generic template; no customized forms for taxonomy-specific data; missing critical data points	Data extraction sheet referenced Table 23 in Appendix A) Comprehensive extraction covering all RQs, taxonomy details, quality metrics
- Multiple extractors	Single author review of robot-extracted data; no independent extraction or validation	Both authors involved, disagreements resolved through discussion
- Piloting	No mention of piloting extraction forms	
6.5 Data Synthesis		
- Synthesis approach	No systematic synthesis; just descriptive listing without integration framework	Comprehensive table (Table 7) showing taxonomy details, categories, characteristics
- Heterogeneity	Not assessed; no analysis of why taxonomies differ or conflict	Robustness R(T) and conciseness C(T) calculations, semantic similarity analysis using jina-embeddings-v3
- Sensitivity analysis	No sensitivity analysis performed	Tables, figures, Thematic analysis of taxonomy characteristics, comparative analysis
- Publication bias	No discussion of publication bias	N/A
REPORTING THE REVIEW		
7.1 Dissemination Strategy	Not mentioned	N/A

7.2 Formatting the Main Report	No structured abstract Did not include "Strengths and Weaknesses" and "Implications for Practice" sections. Short "limitations" section.	Follows comprehensive structure: abstract, introduction, methodology, results, discussion, conclusions
7.3 Evaluating Reports	No mention of external review beyond conference peer review	N/A

Table 20: Compliance of Vera et al.’s work [31] and ours with Kitchenham & Charters’ guidelines for performing a systematic literature review in software engineering [57] (the numbering of the sections refer directly to the sections of Kitchenham & Charters’ guidelines).

7 Implications for Research and Practice

SLRs in general provide a detailed synthesis of evidence pertaining to a research field, which can be used in a number of different ways, such as to identify research gaps, support decision-making and understand in more depth the phenomenon one is interested in [57]. A Scopus search conducted in mid-2025, using a conservative search string (searching for 'systematic literature review' or 'systematic review' in paper titles, combined with software engineering terms, and limited to English-language articles and conference papers in computer science and engineering subject areas) retrieved over a thousand systematic literature reviews now published in software engineering, clearly showing the importance that SLRs bring to the SE research community.

The SLR detailed herein provided, via four Research Questions, a detailed understanding regarding taxonomies in the field of Software Effort Estimation. In addition, gaps, shortcomings, and avenues for future work were also documented, aiming to inform researchers about possible directions of investigation within the context of taxonomies for SEE.

Here, some research directions can include, for example, to carry out investigations in SEE within the context of a wider range of software application domains, such as Engineering/scientific software, Artificial intelligence software, and product-line software. These can lead to the proposal and validation of a new taxonomy for SEE that only caters to these software application domains, or to the extension of an existing taxonomy for SEE to also include these software application domains; or to move in a different direction towards proposing and validating a single holistic SEE taxonomy that caters to all the software application domains identified in Pressman & Maxim [44]. A holistic Taxonomy would enable the SE and specifically the SEE research community to see the forest for the trees and to have a single and properly developed knowledge classification for SEE to support the identification of gaps and future work in this area.

Researchers in SE who wish to propose and validate taxonomies can apply Unterkalmsteiner & Abdeen's quality assessment framework [40] to measure their quality. Furthermore, the limitations identified across the six taxonomies included herein: particularly the lack of external validation, limited extensibility documentation, and insufficient attention to mutual exclusiveness, serve as concrete requirements that future taxonomy development efforts should address.

Finally, taxonomies should be validated in practice, as this was one of the main shortcomings identified when applying the taxonomies’ quality assessment by Unterkalmsteiner & Abdeen [40]. Therefore, future work in this area should also involve the validation of one or more taxonomies for SEE in industrial settings.

Regarding the implications of this SLR’s results for practice, there are several ways in which industry can benefit from the synthesis of evidence detailed herein. For example, the combined list of categories and characteristics pertaining to software effort predictors can be used when estimating effort for software projects. For such scenarios to be effective, it would be important to combine it with tool support, so to enable practitioners to easily navigate such detailed list of categories and characteristics, and to build different ‘views’ containing subsets of categories & characteristics deemed most applicable to a

given development context and software application domain. Such views can later also be used to inform a data collection of past completed projects, in case a company wished to employ, in addition to expert-driven estimation, other means to estimate effort (e.g. using statistical or machine-learning algorithms [36]).

Note that in the Introduction Section we have also listed several advantages of Taxonomies for both research and practice, based upon Ralph [14] and Usman et al. [12].

8 Threats to validity

Several factors had to be considered when assessing the quality of the evidence that is being detailed in this SLR. We employ here the guidelines provided by Ampatzoglou et al. [82] as means to discuss threats to the validity of our SLR, and the corresponding mitigation strategies we used. These are all summarized in Table 21.

Study selection and publication bias. The ways used to construct a search string, selecting digital libraries and determining an appropriate starting year may threaten the validity of the search strategy. To mitigate these threats, we systematically constructed a search string using insights from previous literature and conducted pilot searches to refine it. In addition, broad search strategies were employed to minimize the exclusion of relevant studies.

To address possible threats in the study selection, we employed the inclusion and exclusion criteria previously documented in the SLR's protocol. A subset of studies was independently screened by both authors, and any differences were discussed until consensus was reached. Note that grey literature and inaccessible papers were excluded. Publication bias can also result from negative results not being submitted for publication, or from documents published in more obscure sources. We mitigated this bias by performing the search in two of the biggest indexing sites for scientific literature: Scopus and Google Scholar. The latter indexes a wide variety of sources, which include smaller venues [83]. By using such databases we believe that publication bias was mitigated.

Data validity. Threats to data validity include small sample size, heterogeneity among studies, publication bias, and potential inaccuracies during data extraction. These threats are among the most critical to the validity of the data, as they lead to unreliable results and conclusions. To address the small sample size, the authors conducted broad searches across two large databases and implemented backward snowballing. Such approach helped capture diverse studies and draw conclusions based on observable trends, while reaching consensus about the meaning and content. With regard to obtaining a small sample of primary studies, this is not uncommon in SE, i.e., there have also been other areas in SE for which a small number of taxonomies have been proposed. Examples are SE Models and Methods (7 taxonomies) and SE Processes (9 taxonomies) [12]. Similarly, systematic literature reviews in specialized software engineering domains often yield small sample sizes, as demonstrated by Sulayman and Mendes [84] who identified only 4 relevant studies in their SLR of software process improvement in small and medium Web companies; yet their work successfully identified important research gaps and provided valuable insights to the field. We believe that a small sample should not be a justifiable reason to refrain from carrying out an SLR if such SLR can be used to identify important gaps and avenues for future work in the area. The heterogeneity of primary studies was mitigated by carefully constructing a data extraction criteria through discussions between both authors and by following Kitchenham & Charters [57], SEGRESS [55], and PRISMA's [56] guidelines on how to perform a literature review. These criteria ensured consistent extraction and categorization of key information across all studies, despite differences in their methodologies and reporting styles. Potential biases in data extraction were addressed by using data extraction forms. Here the data was extracted by the second author and validated by the first author. Any disagreements during the whole process were discussed between both authors until consensus was reached.

Research validity. Threats to research validity included concerns about repeatability, choice of research method, deviations in the review process, and the generalizability of the findings. Repeatability was addressed by developing a detailed protocol based primarily on Kitchenham & Charters [57] and PRISMA's [56] guidelines, facilitating the availability of the data, and the involvement of both authors. Brainstorming sessions were conducted to ensure coverage of the research questions to align with the

review process and the study’s objectives. Generalizability threats were addressed through comprehensive searches across two large databases to ensure all the relevant taxonomies in software engineering effort estimation that were published in English were collected. While our findings are specific to the effort estimation domain and cannot be directly generalized to taxonomies in other software engineering areas, the systematic nature of our search provides confidence that our results represent the current state of taxonomy research in software effort estimation. The limited existing work in this area (one unpublished technical report [31]) demonstrates the contribution of our review for the SEE field.

Construct validity. Threats to construct validity relate to whether our measurements accurately capture the intended quality constructs during the taxonomies’ quality assessment.

When measuring a taxonomy’s comprehensiveness, one of the criterion proposed by Unterkalmsteiner & Abdeen [40] for its assessment was to evaluate whether the taxonomy creators had a diverse background. No detailing was provided on how to carry out such evaluation; only Britto et al. [27] provided some information about authors’ background, and their background diversity was evaluated by the first author as she was a co-author in the paper and knows the other two co-authors well.

Additional quality attributes proved difficult to assess reliably through structural analysis alone, further threatening construct validity:

Extensibility: Rate of change calculations require longitudinal data unavailable for static analysis

Explanatory and Mutual exclusiveness: These attributes are rarely documented in the primary studies included here.

Reliability: Requires the use of such taxonomies in practice, which has not been done by most taxonomies.

Therefore, we believe that in our comparative analysis, the attributes robustness and conciseness were the primary differentiating factors between taxonomies.

Note that the quality assessment was initially done by the second author after discussions with the first author, and later validated by the first author. Therefore, the fact that such assessment has not been done independently by both authors can be seen as a limitation.

With regard to measuring the semantic similarity between categories and characteristics, we used a single threshold of 90% and used jina-embeddings-v3 [54] in such analysis. No sensitivity analysis was done to compare different thresholds and/or different embedding models. This is a limitation of our work and is a topic to be investigated as future work.

Threat to Validity Encountered	Mitigating Actions Taken
<i>Study Selection Validity Threats</i>	
Construction of search string	Systematic search string construction through previous literature. Additionally, pilot searches were conducted. Selected known large digital libraries. Used broad searches.
Selection of digital libraries	
Selection of publication venues	
Selection of arbitrary starting year	
Search engine inefficiencies	
Limited number of journals and conferences	
Missing non-English papers	Checked the number of identified papers’ low occurrence amongst the sample.
Paper inaccessibility	
Handling of duplicate articles	Developed a strategy for handling duplicate articles.
Inclusion/exclusion of grey lit.	Checked against study goals, decided to exclude all grey literature.
Study inclusion/exclusion	Defined an inclusion/exclusion criteria.

	Discussion between authors until consensus was reached. To mitigate the threat relating to investigator bias, both authors screened separately through all the 356 titles & abstracts from a Scopus search; and through another 1k titles & abstracts from the set of 3125 titles & abstracts retrieved via a Google scholar search. Disagreements ($n = 6$) were resolved through discussion, and the inter-rater agreement was high (Cohen's $\kappa = 0.83$, observed agreement = 99.6%).
<i>Data Validity Threats</i>	
Small sample size	Drew conclusions based on evidence.
Heterogeneity of primary studies	Used broad searches.
Choices of variables to be extracted	Discussion between authors until consensus was reached and an extraction form was co-designed
Publication bias	Comprehensive literature search performed on two large databases.
Potential lack of relationships	
Validity of primary studies	Developed a concise and transparent inclusion/exclusion criteria.
Data extraction bias	Involved both authors. Used expert opinion. Discussion between authors until consensus was reached.
Quality assessment subjectivity	
Data extraction inaccuracies	
Unverified data extraction	
Misclassification of primary studies	
Lack of statistical analysis	Given the qualitative and conceptual focus of the data rather than quantitative, empirical measurements, traditional statistical analysis (e.g., hypothesis testing, regression) is not applicable or meaningful in this context.
Robustness of initial classification	Established a clear classification through the research questions. Both authors were involved. Classification revisited periodically.
Construction of attribute framework	Comprehensive attribute identification through research questions. Collected attributes proposed in [40].
Researcher bias	Predefined inclusion and exclusion criteria. Both authors conducted the review. Both authors designed and used a protocol containing an extraction sheet for concise data extraction.
<i>Research Validity Threats</i>	
Repeatability	Involved both authors. Made data available¹. Developed a protocol.
Chosen research method	Developed a concise protocol agreed by both authors, and based primarily on Kitchenham & Charters [57], SEGGRESS [55], and PRISMA guidelines [56].
Review process deviation	
Coverage of research questions	Brainstorming.
Lack of comparable studies	
Unfamiliar with research field	First author is an expert in SEE and has co-authored three of the six papers in taxonomies for SEE.

Generalizability	Used a broad search.
Research not applicable to other domains/organizations	

Table 21: Threats to validity and mitigation strategies

9 Conclusions and Future work

This paper presented the results of a systematic literature review on taxonomies for software effort estimation (SEE), carried out based primarily on Kitchenham et al.'s [55] and PRISMA's [56] guidelines. Four research questions were used to drive the SLR's evidence synthesis.

A total of respectively 356 and 3125 articles were retrieved via searches on Scopus and Google Scholar; regarding the review process, both authors independently screened all 356 Scopus results and the first 1,000 Google Scholar results; of these, only five papers required discussion until consensus was reached, indicating a high level of agreement between the two reviewers; after applying the inclusion/exclusion criteria, six studies presenting taxonomies for SEE were selected.

The first taxonomy for SEE was published in 2005 and it took another 11 years for a second taxonomy for SEE to be published (2016); these were followed by another two taxonomies published in 2017, a fifth one in 2019 and the sixth and last one to date to be published in 2021.

These taxonomies for SEE have been published across diverse venues, including international conferences (Bajta & Idri [72], Britto et al. [27], Dashti & Gandomani [73], Mendes et al. [29]) and specialised journals (Britto et al. [28], Usman et al. [58]) demonstrating the topic's research value across a number of different venues.

The quality of those taxonomies was assessed using the taxonomies' quality assessment framework by Unterkalmsteiner & Adbeen [40], and later these studies were classified according to Wieringa et al.'s evaluation [43] criteria. In relation to the quality of the taxonomies, there were a few quality attributes that we were unable to assess as no information was reported in the studies. This applied specifically to three attributes: Extensibility, Explanatory and Mutual exclusiveness. Furthermore, only one taxonomy for SEE, by Britto et al. [27], had been applied in practice.

In relation to attributes that we were able to measure, the main findings were as follows:

The robustness value $R(t)$, which uses the semantic similarity of a taxonomy's contents to indicate varying degrees of reliability, is relatively high throughout all studies. Here, Mendes et al. [29] presented the highest score (0.9094), which translates as the most robust taxonomy according to the specified measurement criteria. In contrast, Britto et al. [27] has the lowest robustness (0.8733), likely due to its construction method, which includes repeated terms to improve readability and conciseness.

Conciseness varies significantly among the taxonomies in terms of dimensions, categories, and characteristics. Britto et al. [27] has the highest number of dimensions (6) while Mendes et al. [29] has the lowest (1); Bajta & Idri [72] has the highest number of categories (23), while Britto et al. [27] and Dashti & Gandomani [73] have the lowest (6); Britto et al. [28] has the highest number characteristics (164), while Dashti & Gandomani [73] has the lowest (12).

Finally, Dashti & Gandomani [73] present the highest taxonomy depth (5), while all remaining studies share the same depth - three.

Most of the primary studies included here are of type solution proposal, according to Wieringa et al.'s evaluation [43].

The four RQs included in this SLR and their corresponding main findings are summarized next.

RQ1: What are the categories and characteristics used in existing taxonomies for SEE?

The detailing of categories and characteristics for the six taxonomies for SEE included herein is given throughout Figures 3 to 8, and Tables 8 to 13. Many categories and characteristics were identified, jointly accounting to a total of 81 categories and 365 characteristics.

The six taxonomies differ substantially in structure, ranging from 1 to 6 dimensions, 6 to 26 categories, and 12 to 164 characteristics, with no common design standard or shared structural framework across taxonomies. In addition, they also vary in their content focus: Bajta & Idri [72] and Usman et al. [58] integrate both effort predictors (size measures and cost factors), estimation techniques and estimation context, providing broader coverage. In contrast, Britto et al. [27] focus on contextual factors, while Britto et al. [28] focus exclusively on cost factors; Mendes et al. [29] exclusively on size measures; and Dashti & Gandomani [73] exclusively on estimation techniques.

RQ2: What are the software application domains targeted by the taxonomies identified in this SLR?

A limited coverage is observed in the domains the retrieved SEE taxonomies cover. The identified taxonomies focus predominantly on Application software and Embedded software (three taxonomies), followed by Web/Mobile applications (two taxonomies). One of the primary studies did not specify the software domain(s) focus of their taxonomy for SEE.

Of the seven proposed software application domains four are not represented: system software, engineering/scientific software, product-line software, and artificial intelligence software.

RQ3: (if applicable) Which of the taxonomies identified in this SLR are an evolution of previously published ones? Which aspects were changed?

Only one taxonomy, that by Britto et al. [27], is an evolution of another two taxonomies, and aspect that was extended related to SEE within a distributed software development context.

RQ4: What are the similarities and differences across the identified taxonomies for SEE in terms of their structure and content?

The semantic similarity analysis showed that 36 categories (44.4%) and only 62 characteristics (17%) are conceptually shared across the different taxonomies (see Table 16 and Table 17). When jointly considering all the six different taxonomies for SEE, we had a total of 365 distinct characteristics, of which only 62 had some level of overlap.

The two taxonomies that relate to Web/Mobile applications share nine characteristics; however, one has 26 characteristics (Mendes et al. [29]) and the other has 160 characteristics (Britto et al. [28]).

In relation to the three taxonomies that relate to Application & Embedded software, as shown in Table 26, Bajta & Idri [72] share two characteristics with Britto et al. [27] and 33 with Usman et al. [59]; Britto et al. [27] shares two characteristics with Usman et al. [58].

This study identifies an important gap in Software Effort Estimation (SEE) taxonomies, via the inclusion of six taxonomies in SEE that show a significant knowledge fragmentation. We believe that one of the reasons for such fragmentation was the unawareness of more recent studies proposing taxonomies for SEE about previously published taxonomies for SEE.

The detrimental effect of knowledge fragmentation in general has been discussed in a wide range of fields (e.g. international law [85]; social sciences [86]; educational context [87]; knowledge synthesis for scientific management [88]; psychology [89]).

Such high amount of fragmentation, compounded with the absence of taxonomies for emerging application categories, such as mobile apps and artificial intelligence, supports the need for future work in this area, either by proposing a single SEE taxonomy, or specialised taxonomies per software application domains, for example.

Any of the abovementioned future directions would benefit both practitioners and researchers by establishing a common vocabulary facilitating clearer communication of findings, concepts, and best practices across all SEE methods.

This in turn can reduce ambiguity and enhance collaboration between researchers and practitioners, promoting a more unified approach to addressing challenges in software effort estimation, and simplifying the process of comparing different methods and tools. By providing a clear structure, it would make it easier to identify overlapping methods or differences, thereby supporting meta-analyses and cumulative research that can build on existing knowledge.

As stated in the Introduction section, a solution, being it a single SEE taxonomy, or different Taxonomies for SEE, is the focus of our future work, and part of a larger research project on evidence-based SEE being currently funded by the Danish National Research Foundation [90]. The decision as to which option we will employ – one or more SEE Taxonomy(ies), will be driven by the feedback obtained through two surveys that we will carry out with respectively industry and SEE researchers. These will be the focus of future research publications.

10 Authors' contributions

The first author designed the SLR and both authors contributed to its execution. The first author led the analysis of results, wrote and revised previous versions of the manuscript and was the only author revising and changing significantly the latest version of the manuscript. The second author contributed towards the previous versions of this manuscript and the results.

11 Acknowledgements

We would like to thank Professor Barbara Kitchenham for her significant feedback on a previous version of this manuscript. We would also like to thank JSS' area editor and the reviewers for their comments and suggestions.

Finally, this work is funded by a Danish National Research Foundation Chair Grant DNR181.

References

- [1] E. Mendes, 'Using knowledge elicitation to improve Web effort estimation: Lessons from six industrial case studies', in *2012 34th International Conference on Software Engineering (ICSE)*, Jun. 2012, pp. 1112–1121. doi: 10.1109/ICSE.2012.6227108.
- [2] E. Mendes and N. Mosley, 'Bayesian Network Models for Web Effort Prediction: A Comparative Study', *IEEE Transactions on Software Engineering*, vol. 34, no. 6, pp. 723–737, Nov. 2008, doi: 10.1109/TSE.2008.64.
- [3] S. Al-Qudah, K. Meridji, and K. T. Al-Sarayreh, 'A Comprehensive Survey of Software Development Cost Estimation Studies', in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, Batna Algeria: ACM, Nov. 2015, pp. 1–5. doi: 10.1145/2816839.2816913.
- [4] E. Carbonera, C.K. Farias, and V. Bischoff, 'Software development effort estimation: A systematic mapping study.' *IET Software* 14.4 (2020): 328-344e, doi: 10.1049/iet-sen.2018.5334.
- [5] M. Jorgensen and M. Shepperd, 'A Systematic Review of Software Development Cost Estimation Studies', *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 33–53, Jan. 2007, doi: 10.1109/TSE.2007.256943.
- [6] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra, 'Research patterns and trends in software effort estimation', *Information and Software Technology*, vol. 91, pp. 1–21, Nov. 2017, doi: 10.1016/j.infsof.2017.06.002.
- [7] C. H. Rashid *et al.*, 'Software Cost and Effort Estimation: Current Approaches and Future Trends', *IEEE Access*, vol. 11, pp. 99268–99288, 2023, doi: 10.1109/ACCESS.2023.3312716.
- [8] K. Molokken and M. Jorgensen, 'A review of software surveys on software effort estimation', in *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, Sep. 2003, pp. 223–230. doi: 10.1109/ISESE.2003.1237981.
- [9] B. Boehm, C. Abts, and S. Chulani, 'Software development cost estimation approaches — A survey', *Annals of Software Engineering*, vol. 10, no. 1, pp. 177–205, Nov. 2000, doi: 10.1023/A:1018991717352.

- [10] S. P. Pillai, S. D. Madhukumar, and T. Radharamanan, ‘Consolidating evidence-based studies in software cost/effort estimation — A tertiary study’, in *TENCON 2017 - 2017 IEEE Region 10 Conference*, Nov. 2017, pp. 833–838. doi: 10.1109/TENCON.2017.8227974.
- [11] B. A. Kitchenham, E. Mendes, and G. H. Travassos, ‘Cross versus Within-Company Cost Estimation Studies: A Systematic Review’, *IEEE Transactions on Software Engineering*, vol. 33, no. 5, pp. 316–329, May 2007, doi: 10.1109/TSE.2007.1001.
- [12] M. Usman, R. Britto, J. Börstler, and E. Mendes, ‘Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method’, *Information and Software Technology*, vol. 85, pp. 43–59, 2017.
- [13] F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, 20th Anniversary Edition. Boston, MA: Addison-Wesley, 1995.
- [14] P. Ralph, ‘Toward methodological guidelines for process theories and taxonomies in software engineering’, *IEEE Transactions on Software Engineering*, vol. 45, no. 7, pp. 712–735, 2018.
- [15] L.A. Hug, B.J. Baker, K. Anantharaman, C.T. Brown, A.J. Probst, C.J. Castelle, C.N. Butterfield, A.W. Hernsdorf, Y. Amano, K. Ise, Y. Suzuki, ‘A new view of the tree of life’, *Nat Microbiol*, vol. 1, no. 5, pp. 1–6, Apr. 2016, doi: 10.1038/nmicrobiol.2016.48.
- [16] H. D. Parks, Maria Chuvochina, David W. Waite, Christian Rinke, Adam Skarszewski, Pierre-Alain Chaumeil, and Philip Hugenholtz, ‘A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life’, *Nat Biotechnol*, vol. 36, no. 10, pp. 996–1004, Nov. 2018, doi: 10.1038/nbt.4229.
- [17] L. A. Hug, ‘The ever-changing tree of life’, *Nat Microbiol*, vol. 9, no. 8, pp. 1906–1908, Aug. 2024, doi: 10.1038/s41564-024-01768-w.
- [18] H. Washizaki, ‘Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)’, *IEEE Computer Society*, 2024.
- [19] ‘ISO/IEC 25002:2024’, ISO. Accessed: Feb. 23, 2026. [Online]. Available: <https://www.iso.org/standard/78175.html>
- [20] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, ‘Capability maturity model, version 1.1’, *IEEE Software*, vol. 10, no. 4, pp. 18–27, Jul. 1993, doi: 10.1109/52.219617.
- [21] Y.-Y. Cheng and Y. Xia, ‘A systematic review of methods for aligning, mapping, merging taxonomies in information sciences’, *Journal of Documentation*, vol. 79, no. 6, pp. 1413–1439, 2023, doi: 10.1108/JD-01-2023-0003.
- [22] G. Rodríguez-Pérez, R. Nadri, and M. Nagappan, ‘Perceived diversity in software engineering: a systematic literature review’, *Empir Software Eng*, vol. 26, no. 5, p. 102, Jul. 2021, doi: 10.1007/s10664-021-09992-2.
- [23] P. T. Metzger, W. M. Grundy, M. V. Sykes, A. Stern, J. F. Bell III, C. E. Detelich, K. Runyon, and M. Summers., ‘Moons are planets: Scientific usefulness versus cultural teleology in the taxonomy of planetary science’, *Icarus*, vol. 374, p. 114768, Mar. 2022, doi: 10.1016/j.icarus.2021.114768.
- [24] N. Erasmus, A. McNeill, M. Mommert, D. E. Trilling, A. A. Sickafoose, and K. Paterson, ‘A Taxonomic Study of Asteroid Families from KMTNET-SAAO Multiband Photometry’, *ApJS*, vol. 242, no. 2, p. 15, May 2019, doi: 10.3847/1538-4365/ab1344.
- [25] S. Dodd, M. Clarke, L. Becker, C. Mavergames, R. Fish, and P. R. Williamson, ‘A taxonomy has been developed for outcomes in medical research to help improve knowledge discovery’, *Journal of Clinical Epidemiology*, vol. 96, pp. 84–92, Apr. 2018, doi: 10.1016/j.jclinepi.2017.12.020.
- [26] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya, ‘Quantum computing: A taxonomy, systematic review and future directions’, *Software: Practice and Experience*, vol. 52, no. 1, pp. 66–114, 2022, doi: 10.1002/spe.3039.
- [27] R. Britto, E. Mendes, and C. Wohlin, ‘A specialized global software engineering taxonomy for effort estimation’, in *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, IEEE, 2016, pp. 154–163.
- [28] R. Britto, M. Usman, and E. Mendes, ‘A taxonomy of Web effort predictors’, *Journal of Web Engineering*, pp. 541–570, 2017.
- [29] E. Mendes, S. Counsell, and N. Mosley, ‘Towards a taxonomy of hypermedia and web application size metrics’, in *International Conference on Web Engineering*, Springer, 2005, pp. 110–123.

- [30] A. Trendowicz and R. Jeffery, *Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success*. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-03629-8.
- [31] T. Vera, S. F. Ochoa, and D. Perovich, ‘Survey of software development effort estimation taxonomies’, *Computer Science Department, University of Chile: Santiago, Chile*, 2017.
- [32] M. Dixon-Woods, D. Cavers, S. Agarwal, E. Annandale, A. Arthur, J. Harvey, and A. J. Sutton, ‘Conducting a critical interpretive synthesis of the literature on access to healthcare by vulnerable groups’, *BMC Medical Research Methodology*, vol. 6, no. 1, p. 35, Jul. 2006, doi: 10.1186/1471-2288-6-35.
- [33] C.-U. Pae, ‘Why Systematic Review rather than Narrative Review?’, *Psychiatry Investig*, vol. 12, no. 3, pp. 417–419, Jul. 2015, doi: 10.4306/pi.2015.12.3.417.
- [34] A. Sutton, M. Clowes, L. Preston, and A. Booth, ‘Meeting the review family: exploring review types and associated information retrieval requirements’, *Health Information & Libraries Journal*, vol. 36, no. 3, pp. 202–222, 2019, doi: 10.1111/hir.12276.
- [35] M.D.J. Peters, Christina M. Godfrey, Hanan Khalil, Patricia McInerney, Deborah Parker, and Cassia Baldini Soares. "Guidance for conducting systematic scoping reviews." *JBI Evidence Implementation* 13, no. 3 (2015): 141-146.
- [36] E. Mendes, *Cost Estimation Techniques for Web Projects*. IGI Global, 2007.
- [37] G. De Vito, S. Di Martino, F. Ferrucci, C. Gravino, and F. Palomba, ‘LLM-Based Automation of COSMIC Functional Size Measurement From Use Cases’, *IEEE Transactions on Software Engineering*, vol. 51, no. 5, pp. 1500–1523, May 2025, doi: 10.1109/TSE.2025.3554562.
- [38] V. Tawosi, R. Moussa, and F. Sarro, ‘Agile Effort Estimation: Have We Solved the Problem Yet? Insights From the Replication of the GPT2SP Study’, in *2024 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Mar. 2024, pp. 1034–1041. doi: 10.1109/SANER60148.2024.00111.
- [39] G. J. Baratto, E. M. De Bortoli Fávero, D. Casanova, and J. T. Oliva, ‘Z-Se2: A Model for Software Effort Estimation Using Llm Gpt3.5’, Dec. 02, 2024, *Social Science Research Network, Rochester, NY*: 5041669. doi: 10.2139/ssrn.5041669.
- [40] M. Unterkalmsteiner and W. Adbeen, ‘A compendium and evaluation of taxonomy quality attributes’, *Expert Systems*, vol. 40, no. 1, p. e13098, 2023.
- [41] A. Kaplan, Thomas Kühn, Sebastian Hahner, Niko Benkler, Jan Keim, Dominik Fuchß, Sophie Corallo, and Robert Heinrich, ‘Introducing an Evaluation Method for Taxonomies’, in *The International Conference on Evaluation and Assessment in Software Engineering 2022*, Gothenburg Sweden: ACM, Jun. 2022, pp. 311–316. doi: 10.1145/3530019.3535305.
- [42] N. E. Fenton and S. L. Pfleeger, *Software metrics: A rigorous approach*, 2nd edn. London, OH: PWS, 1998.
- [43] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, ‘Requirements engineering paper classification and evaluation criteria: a proposal and a discussion’, *Requirements Eng*, vol. 11, no. 1, pp. 102–107, Mar. 2006, doi: 10.1007/s00766-005-0021-6.
- [44] R. S. Pressman and B. R. Maxim, *Software engineering: a practitioner’s approach*, 8th edn. New York, NY: McGraw-Hill Education, 2014.
- [45] A. Giabelli, L. Malandri, F. Mercorio, M. Mezzanzanica, and N. Nobani, ‘Embeddings Evaluation Using a Novel Measure of Semantic Similarity’, *Cogn Comput*, vol. 14, no. 2, pp. 749–763, Mar. 2022, doi: 10.1007/s12559-021-09987-7.
- [46] C. Zhang *et al.*, ‘TaxoGen: Unsupervised Topic Taxonomy Construction by Adaptive Term Embedding and Clustering’, Dec. 22, 2018, *arXiv*: arXiv:1812.09551. doi: 10.48550/arXiv.1812.09551.
- [47] Y. Lu *et al.*, ‘Self-supervised Topic Taxonomy Discovery in the Box Embedding Space’, Aug. 27, 2024, *arXiv*: arXiv:2408.15050. doi: 10.48550/arXiv.2408.15050.
- [48] Z. S. Harris, ‘Distributional Structure’, *WORD*, vol. 10, no. 2–3, pp. 146–162, Aug. 1954, doi: 10.1080/00437956.1954.11659520.
- [49] J. Firth, (1957). A Synopsis of Linguistic Theory, 1930-55. In *Studies in Linguistic Analysis* (pp. 1-31). Special Volume of the Philological Society. Oxford Blackwell. [Reprinted as Firth (1968)] - References - Scientific Research Publishing’. Accessed: Feb. 23, 2026. [Online]. Available: <https://www.scirp.org/reference/ReferencesPapers?ReferenceID=1846447>

- [50] T. O. B. Odden, H. Tyseng, J. T. Mjaaland, M. F. Kreutzer, and A. Malthé-Sørenssen, 'Using Text Embeddings for Deductive Qualitative Research at Scale in Physics Education', Feb. 28, 2024, *arXiv*: arXiv:2402.18087. doi: 10.48550/arXiv.2402.18087.
- [51] L. van der Maaten and G. Hinton, 'Visualizing Data using t-SNE', *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.
- [52] 'Principal component analysis', *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, Aug. 1987, doi: 10.1016/0169-7439(87)80084-9.
- [53] L. McInnes, J. Healy, and J. Melville, 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction', Sep. 18, 2020, *arXiv*: arXiv:1802.03426. doi: 10.48550/arXiv.1802.03426.
- [54] S. Sturua *et al.*, 'jina-embeddings-v3: Multilingual Embeddings With Task LoRA', Sep. 19, 2024, *arXiv*: arXiv:2409.10173. doi: 10.48550/arXiv.2409.10173.
- [55] B. Kitchenham, L. Madeyski, and D. Budgen, 'SEGRESS: Software Engineering Guidelines for REporting Secondary Studies', *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 1273–1298, Mar. 2023, doi: 10.1109/TSE.2022.3174092.
- [56] M.J. Page, J.E. McKenzie, P.M. Bossuyt, I. Boutron, T.C. Hoffmann, C.D. Mulrow, L. Shamseer, J.M. Tetzlaff, E.A. Akl, S.E. Brennan, and R. Chou., 'The PRISMA 2020 statement: an updated guideline for reporting systematic reviews', *BMJ*, p. n71, Mar. 2021, doi: 10.1136/bmj.n71.
- [57] B. Kitchenham and S. Charters, 'Guidelines for performing Systematic Literature Reviews in Software Engineering', vol. 2, Jan. 2007.
- [58] M. Usman, J. Börstler, and K. Petersen, 'An effort estimation taxonomy for agile software development', *International Journal of Software Engineering and Knowledge Engineering*, vol. 27, no. 04, pp. 641–674, 2017.
- [59] R. Britto, C. Wohlin, and E. Mendes, 'An extended global software engineering taxonomy', *Journal of Software Engineering Research and Development*, vol. 4, pp. 1–24, 2016.
- [60] S. P. Pillai, S. Madhukumar, and T. Radharamanan, 'Consolidating evidence based studies in software cost/effort estimation—A tertiary study', in *TENCON 2017-2017 IEEE Region 10 Conference*, IEEE, 2017, pp. 833–838.
- [61] C. U. Press, 'Cambridge Thesaurus'. 2024. [Online]. Available: <https://dictionary.cambridge.org/thesaurus/>
- [62] Elsevier, 'Scopus: Abstract and citation database of peer-reviewed literature'. 2024. [Online]. Available: <https://www.scopus.com>
- [63] G. LLC, 'Google Scholar: Search engine for scholarly literature'. 2024. [Online]. Available: <https://scholar.google.com>
- [64] 'ACM partnerships with indexing services'. Accessed: May 28, 2025. [Online]. Available: <https://authors.acm.org/journals/journals-indexing-list>
- [65] 'Abstracting & Indexing (A&I) Databases', IEEE Author Center Journals. Accessed: May 28, 2025. [Online]. Available: <https://journals.ieeeauthorcenter.ieee.org/when-your-article-is-published/abstracting-indexing-ai-databases/>
- [66] M. E. Falagas, V. A. Pitsouni, G. A. Malietzis, and G. Pappas, 'Comparison of PubMed, Scopus, Web of Science, and Google Scholar: Strengths and Weaknesses', *The FASEB Journal*, vol. 22, no. 2, pp. 338–342, 2008, doi: 10.1096/fj.07-9492LSF.
- [67] J.-F. Gehanno, A. Rollin, and W. Darmoni, 'Is it Possible to Use Google Scholar to Search for Scientific Publications?', *European Journal of Clinical Investigation*, vol. 43, no. 1, pp. 1–5, 2013, doi: 10.1111/j.1365-2362.2012.02650.x.
- [68] R. Miller, 'Scholarly: A Python package for accessing and processing scholarly articles'. 2023. [Online]. Available: <https://github.com/OrganicIrradiation/scholarly>
- [69] P. Karadimos and L. Anthopoulos, 'A taxonomy of machine learning techniques for construction cost estimation', *Innov. Infrastruct. Solut.*, vol. 9, no. 11, p. 420, Oct. 2024, doi: 10.1007/s41062-024-01705-0.
- [70] R. Sánchez-Morcilio and F. Quiles-Torres, 'The taxonomy of estimation in software development projects', *Issues in Information Systems*, vol. 18, no. 3, 2017.
- [71] D. Šmite, C. Wohlin, Z. Galviņa, and R. Prikladnicki, 'An empirically based terminology and taxonomy for global software engineering', *Empirical Software Engineering*, vol. 19, pp. 105–153, 2014.

- [72] M. El Bajta and A. Idri, 'A Software Cost Estimation Taxonomy for Global Software Development Projects.', in *ICSOF*, 2019, pp. 218–225.
- [73] M. Dashti and T. J. Gandomani, 'A Taxonomy of Approaches and Methods for Software Effort Estimation', in *Innovations in Computer Science and Engineering: Proceedings of the Ninth ICICSE, 2021*, Springer, 2022, pp. 97–105.
- [74] Ritu and P. Bhambri, 'Software Effort Estimation with Machine Learning – A Systematic Literature Review', in *Agile Software Development*, John Wiley & Sons, Ltd, 2023, pp. 291–308. doi: 10.1002/9781119896838.ch15.
- [75] A. Ali and C. Gravino, 'A systematic literature review of software effort prediction using machine learning methods', *Journal of Software: Evolution and Process*, vol. 31, no. 10, p. e2211, 2019, doi: 10.1002/smr.2211.
- [76] H. R. Moechtar, S. Sulisty, and R. Ferdiana, 'A Review on Improving the Accuracy of Effort Estimation in Software Development with Agile Method', in *2024 11th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Aug. 2024, pp. 31–36. doi: 10.1109/ICITACEE62763.2024.10761956.
- [77] V. Terragni, A. Vella, P. Roop, and K. Blincoe, 'The Future of AI-Driven Software Engineering', *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 5, p. 120:1-120:20, May 2025, doi: 10.1145/3715003.
- [78] O. Danushi, S. Forti, and J. Soldani, 'Carbon-Efficient Software Design and Development: A Systematic Literature Review', *ACM Comput. Surv.*, vol. 57, no. 10, p. 267:1-267:35, May 2025, doi: 10.1145/3728638.
- [79] 'Types of Application Software: The Complete List (2021 Update)'. Accessed: Nov. 04, 2024. [Online]. Available: <https://www.bridge-global.com/blog/types-of-application-software/>
- [80] A. Forward and T. C. Lethbridge, 'A taxonomy of software types to facilitate search and evidence-based software engineering', in *Proceedings of the 2008 conference of the center for advanced studies on collaborative research meeting of minds - CASCON '08*, Ontario, Canada: ACM Press, 2008, p. 179. doi: 10.1145/1463788.1463807.
- [81] C. McMillan, M. Linares-Vásquez, D. Poshyvanyk, and M. Grechanik, 'Categorizing software applications for maintenance', in *2011 27th IEEE International Conference on Software Maintenance (ICSM)*, Sep. 2011, pp. 343–352. doi: 10.1109/ICSM.2011.6080801.
- [82] A. Ampatzoglou, S. Bibi, P. Avgeriou, M. Verbeek, and A. Chatzigeorgiou, 'Identifying, categorizing and mitigating threats to validity in software engineering secondary studies', *Information and Software Technology*, vol. 106, pp. 201–230, Feb. 2019, doi: 10.1016/j.infsof.2018.10.006.
- [83] A. Martín-Martín, M. Thelwall, E. Orduna-Malea, and E. D. López-Cózar, 'Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations', *Scientometrics*, vol. 126, no. 1, pp. 871–906, Jan. 2021, doi: 10.1007/s11192-020-03690-4.
- [84] M. Sulayman and E. Mendes, 'A Systematic Literature Review of Software Process Improvement in Small and Medium Web Companies', in *Advances in Software Engineering*, D. Ślęzak, T. Kim, A. Kiumi, T. Jiang, J. Verner, and S. Abrahão, Eds, Berlin, Heidelberg: Springer, 2009, pp. 1–8. doi: 10.1007/978-3-642-10619-4_1.
- [85] S. Thathong, 'Lost in Fragmentation: The Traditional Knowledge Debate Revisited', *Asian Journal of International Law*, vol. 4, no. 2, pp. 359–389, Jul. 2014, doi: 10.1017/S2044251313000350.
- [86] R. D. Beam, 'Fragmentation of Knowledge: An Obstacle to Its Full Utilization', in *The Optimum Utilization Of Knowledge*, Routledge, 1980.
- [87] M. F. Norese and F. Salassa, 'Structuring fragmented knowledge: a case study', *Knowledge Management Research & Practice*, vol. 12, no. 4, pp. 454–463, Nov. 2014, doi: 10.1057/kmrp.2013.22.
- [88] V. Z. Chen and M. A. Hitt, 'Knowledge Synthesis for Scientific Management: Practical Integration for Complexity Versus Scientific Fragmentation for Simplicity', *Journal of Management Inquiry*, vol. 30, no. 2, pp. 177–192, Apr. 2021, doi: 10.1177/1056492619862051.
- [89] J. Valsiner, 'Dangerous Curves in Knowledge Construction within Psychology: Fragmentation of Methodology', *Theory & Psychology*, vol. 16, no. 5, pp. 597–612, Oct. 2006, doi: 10.1177/0959354306067439.

- [90] ‘DNRF Chair: Professor Emilia Mendes’, Danmarks Grundforskningsfond. Accessed: Nov. 04, 2024. [Online]. Available: <https://dg.dk/centers/dnrf-chair-professor-emilia-mendes/>
- [91] S. Gavin, K. Mengersen, N. Meader, 2014, Potential uses of Bayesian networks as tools for synthesis of systematic reviews of complex interventions. *Research Synthesis Methods*. 5. 10.1002/jrsm.1087.
- [92] L. Kiss, D. Fotheringham, J. Mak, A. McAlpine, C. Zimmerman, 2020, The use of Bayesian networks for realist evaluation of complex interventions: evidence for prevention of human trafficking. *J. Comp. Social Scien.* 10.1007/s42001-020-00067-8.
- [93] J. Higgins, et al., 2019, Synthesising quantitative evidence in systematic reviews of complex health interventions. *BMJ G. Health*. 10.1136/bmjgh-2018-000858.
- [94] T. Rique, E. Dantas, M. Perkusich, K. Gorgônio, H. Almeida and A. Perkusich, "Insights into the Applications of Bayesian Networks in Software Engineering," *2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 2024, pp. 1-6, doi: 10.23919/SoftCOM62040.2024.10721646.
- [95] Guarino, N., Oberle, D., Staab, S. (2009). What Is an Ontology?. In: Staab, S., Studer, R. (eds) *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-92673-3_0.
- [96] Stewart, G. B., Kerrie Mengersen, and Nicholas Meader. "Potential uses of Bayesian networks as tools for synthesis of systematic reviews of complex interventions." *Research synthesis methods* 5.1 (2014): 1-12.
- [97] Huang, Jing, et al. "Interventions to increase the uptake of postpartum diabetes screening among women with previous gestational diabetes: a systematic review and Bayesian network meta-analysis." *American Journal of Obstetrics & Gynecology MFM* 5.10 (2023): 101137.
- [98] Afflitto, Gabriele Gallo, et al. "Cataract and risk of fracture: a systematic review, meta-analysis, and Bayesian network meta-analysis." *Ophthalmology* 132.8 (2025): 921-934.
- [99] E. Mendes, V. Freitas, M. Perkusich, J. Nunes, F.B.A. Ramos, A.A.M. Costa, R.M. Saraiva, A.S. Freire, 2019, Using BN to Estimate the Value of Decisions within the Context of Value-Based SE: A Multiple Case Study. *Int. J. Softw. Eng. Knowl. Eng.* 29(11&12): 1629-1671.
- [100] E. Mendes, N. Mosley, 2008, Bayesian Network Models for Web Effort Prediction: A Comparative Study, *Transactions on Software Eng.*, 34(6):723-737.
- [101] A.T. Misirli, A. B. Bener, 2014, Bayesian networks for evidence-based decision-making in software engineering, *IEEE Transactions on*, 40(6):533–554.
- [102] Guven, Z. A., Mendes, E., & Baker, S. (2026a). Multi-label text classification using prompting strategies via LLMs in Software Effort Estimation. *IEEE Access*, 14, 52753–52775.
- [103] Guven, Z. A., Mendes, E., & Lefteris, S. (2026b). Applying Multi-Label Text Classification to the Software Effort Estimation Primary Studies Using Large Language Models and Prompting strategies. *Transactions on Software Engineering and Methodology*.
- [104] International Organization for Standardization, *Project, programme and portfolio management: guidance on project management*. Nihon kikaku kyōkai, 2020.

Appendix A

Section	Sub-Section	PRISMA Item	Summary	Checklist
Full Report	n/a	n/a	Consider referencing material in the protocol, publishing material in supplementary material, and reporting any large-scale model building exercise separately from the SR report.	☒ Yes. Guidelines mentioned in methodology, checklist in supplementary material (Appendix A)
Title	Title	1	Identify the type of review and specify the topic being reviewed.	☒ Yes. Type of review was identified in the title.
Abstract	Structured abstract	2	Provide a structured summary, including Background, Objective, Methods, Results, and Conclusions.	☒ Yes. Structured with Context, Objective, Method, Results, Conclusions
Introduction	Opening	n/a	Introduce the larger problem the paper is targeting, lay out a broad context for the work, and highlight the importance of the work to a large audience.	☒ Yes. Section 1 introduces SEE importance and context
Introduction	Rationale	3	Describe the rationale of the study and how it contributes to the larger problem.	☒ Yes. Explained need for taxonomies in SEE (Section 1)
Introduction	Objectives	4	Specify the research questions and explain how they contribute to the larger problem.	☒ Yes. Four RQs specified in Section 4.2 (Table 2)
Methods	Eligibility criteria	5	Use the study characteristics to define eligibility criteria based on the intervention or topic of interest. Criteria used to restrict the search must be specified and justified.	☒ Yes. Inclusion/exclusion criteria in Section 4.4
Methods	Information sources	6	Describe all information sources, databases, primary study references, and others (e.g., researchers) with search end dates.	☒ Yes. Scopus and Google Scholar described in Section 4.5
Methods	Search Strategy	7	Present full search strategy, e.g., electronic search strings, manual search, method(s) used to assess achieved completeness.	☒ Yes. Full search strings provided for both databases
Methods	Selection Process	8	State the process for selecting studies, i.e., phases, assessors, handling disagreements, and tools used.	☒ Yes. Section 4.3 describes screening by both authors
Methods	Data Collection Process	9	Specify the method used to collect data from reports, i.e., processes, reviewers, whether they worked independently, tools used.	☒ Yes. Data extraction sheet provided (Table 32)
Methods	Data items	10a	List, define, and justify all outcomes for which data was sought, explaining their relationship to the research questions.	☒ Yes. Data items linked to RQs
Methods	Data items	10b	List and define all non-outcome variables for which data was sought.	☒ Yes. Comprehensive list in extraction sheet
Methods	Study Risk Of Bias Assessment	11	Specify the methods used to assess risk of bias in the included studies.	☒ Yes. Unterkalmsteiner & Abdeen [41] framework used
Methods	Effect Measures	12	Specify for each outcome the effect measure(s) used in the synthesis.	☐ N/A for this type of review
Methods	Analysis and Synthesis methods	13a	Describe the process used to decide which studies were eligible for each synthesis.	☒ Yes. Eligibility based on proposing SEE taxonomies
Methods	Analysis and Synthesis methods	13b	Describe any methods required to prepare the data for presentation or synthesis.	☒ Yes. Semantic similarity analysis described
Methods	Analysis and Synthesis methods	13c	Describe any methods used to tabulate or visually display results.	☒ Yes. Multiple tables and visualizations described

Results	Results of Analyses and Syntheses	20c	Present results of all sensitivity analysis conducted.	☒ Yes. (Table 6) robustness analysis
Results	Results of Analyses and Syntheses	20d	Present results of all investigations of possible causes of heterogeneity.	☒ Yes. Domain coverage, evolution analysis
Results	Reporting Biases	21	Report results of assessing publication bias for each synthesis.	Presented in Results 6
Results	Certainty of Evidence	22	Present assessment of certainty in the body of evidence for each finding.	☐ N/A for this review type
Discussion	Discussion	23a	Provide a general interpretation of the results in the context of other evidence.	☒ Yes. Section 6 compares with Vera et al.
Discussion	Discussion	23b	Discuss any limitations of the evidence included in the review.	☒ Yes. Quality attribute limitations discussed
Discussion	Discussion	23c	Discuss any limitations of the review process used.	☒ Yes. Threats to validity (Table 30)
Discussion	Discussion	23d	Discuss implications of the results for practice, policy and future research.	☒ Yes. Section 7 implications, future work
Registration and Protocol		24a	Provide registration information or state that the review was not registered.	Review was not registered
Registration and Protocol		24b	Indicate where the review protocol can be accessed.	☒ GitHub repository mentioned in Appendix C
Registration and Protocol		24c	Describe and explain any amendments to information provided at registration.	Review was not registered
Support		25	Describe sources of financial and non-financial support for the review.	☒ Yes. Danish National Research Foundation Chair Grant DNRF181
Competing Interests		26	Declare competing interests of the review authors.	No conflicting interests
Availability Of Data, Code and Other Materials		27	Report which materials are publicly available and where they can be found.	☒ Yes. GitHub repository mentioned in Appendix C

Table 22: SEGRESS Checklist

Taxonomy	Value	Additional Notes
Data Extractor Name		
Study/Paper name		
Year of publication		
Site of publication		
Is the taxonomy a revision of a previously published one?		
Which taxonomy is it a revision of?		
What aspects were changed or added?		
What are the categories of the taxonomy?		
What are the characteristics of the taxonomy?		
What are the contexts? (e.g. Conventional, Web, Global)?		
Was the paper validated according to its classification?		
<i>Related to the quality of the taxonomy</i>		
Does the taxonomy originate from a diverse set of data sources?		
Do the taxonomy creators have a diverse background?		
Were diverse data analysis methods used to create the taxonomy?		
Has the taxonomy been evaluated by external experts?		
R(T)		
Number of Dimensions		
Number of Categories		
Number of Characteristics		
Maximum Depth		
C(T)		
Change process covers the following constructs		
Change process covers the following change types		
Mandate to change the taxonomy		
Customization of the taxonomy		
RoC		
Structuring elements besides dimensions?		
Support for choosing dimensions or categories?		
Classification constraint		
Is the taxonomy reliable?		

Table 23: Data extraction sheet

Taxonomy	Categories	Characteristics
Bajta & Idri [73]	A: Accuracy measure, Actual cost C: Cost estimate, Cost estimation context, Cost estimators D: Distributed teams distances E: Estimated cost, Estimation dimension, Estimation technique N: Number of sites P: Planning, Planning approaches, Product requirement, Product size, Project activities, Project domain, Project setting R: Research & dev T: Team experience, Team size, Team structure U: Use technique T: Team experience, Team size, Team structure U: Use technique	A: Actual cost.value, Agile, Analysis, Availability B: Baseline comparison C: Capability maturity model integration, Case-based reasoning, Close onshore, Commissioning, Conceptualization, Constructive cost model D: Delphi, Design, Detail planning, Distant onshore E: Effort hours, Estimated value, Execution, Expert judgment F: Far offshore, Feasibility study, Finance, Fuzzy similar G: Genetic algorithms, Geographical distance, Group-based estimation H: Hardware, Healthcare I: Implementation, Individual M: Machine learning, Maintainability, Maintenance N: Near offshore, Non-machine learning, Number of sites.value, Number of team members P: Performance, Planning approaches.other, Portfolio, Preliminary planning, Product requirement.other, Project activities.other, Project domain.other R: Reliability, Risk S: Security, Sensitivity analysis, Size report, Socio-cultural distance, Staff/cost, Statistics analysis, System investigation, Systems engineering T: Team experience.considered, Team experience.not considered, Team structure.considered, Team structure.not considered, Telecommunication, Temporal distance, Testing V: Variation reduction
Britto et al. [28]	C: Client, Complexity, Cost driver D: Development company F: Functionality L: Length O: Object-oriented P: Product, Project S: Size metric T: Team, Technology	A: Accessibility level, Adaptation complexity, Anchor count, Architecture, Association center slot count, Association slot size, Attribute count, Authoring tool type, Availability level C: Class complexity, Class coupling, Client script count, Client.personality, Cluster count, Cluster node size, Cluster slot count, Cohesion, Cohesion complexity, Collection center slot count, Collection slot size, Comment count, Common software measurement international consortium, Communication level, Compactness, Component complexity, Component count, Component granularity level, Component slot count, Concern coupling, Concern module count, Concern operation count, Concurrency level, Connectivity density, Control flow complexity, Cyclomatic complexity D: Data flow complexity, Data usage complexity, Data web points, Database size, Deployment platform experience level, Design volatility, Development restriction, Difficulty level, Diffusion cut count, Documentation level, Domain experience level E: Entity count, Experience level F: Feature count, Flexibility level, Focus factor H: High feature count I: In-house experience, Indifferent concern count, Information slot count, Infrastructure, Inner/sub concern count, Innovation level, Input complexity, Installability level, Integration with legacy systems, Interface complexity, International function point users group, It literacy L: Layout complexity, Lessons learned repository, Lines of code, Link count, Low feature count M: Maintainability level, Mapped workflows, Media allocation, Media count, Media duration, Memory efficiency level, Metrics' program, Model association complexity, Model collection complexity, Model link complexity, Model node size, Model slot size, Modularity level, Module attribute count, Module count, Module point cut count, Motivation level N: New complexity, New media count, New web page count, Node

		count, Node slot size, Novelty level, Number of programming languages, Number of projects in parallel O: Object-oriented function points, Object-oriented heuristic function points, Oo experience level, Operation count, Operational mode, Output complexity P: Page complexity, Platform support level, Platform volatility level, Portability level, Process efficiency level, Processing requirements, Product.type, Productivity level, Program count, Programming language experience level, Project management level, Project.type, Publishing model unit count, Publishing unit count Q: Quality level R: Rapid app development, Readability level, Reliability level, Requirements clarity level, Requirements novelty level, Requirements volatility level, Resource level, Reusability level, Reused comment count, Reused component count, Reused high feature count, Reused lines of code, Reused low feature count, Reused media allocation, Reused media count, Reused program count, Risk level, Robustness level S: Scalability level, Section count, Security level, Segment count, Semantic association count, Server script count, Slot count, Slot granularity level, Software development experience, Software reuse, Spi program, Stability level, Statement count, Storage constraint, Stratum, Structure T: Team capability, Team size, Technical factors, Testability level, Time efficiency level, Time restriction, Tool experience level, Total complexity, Trainability level U: Usability level, Use case count W: Web objects, Web page allocation, Web page count, Work team level
Britto et al. [27]	E: Estimation process architectural model, Estimation process role, Estimation stage P: Project R: Relationship S: Site	C: Centralized D: Distributed E: Estimation stage.early, Estimation stage.early & late, Estimation stage.late, Estimator, Estimator & provider G: Geographic distance L: Legal entity, Location P: Provider S: Semi-distributed T: Temporal distance
Dashti & Gandomani [74]	A: Algorithmic B: Basic estimating methods C: Combined estimating methods, Computational intelligence E: Estimation process architectural model N: Non-algorithmic	A: Ai-combined hybrid, Analogy-based, Artificial neural networks B: Basic-combination C: Constructive cost model E: Evolutionary, Expert judgment F: Fuzzy logic L: Legal entity S: Software evaluation and estimation for risk, Software life cycle management, Swarm
Mendes et al. [29]	C: Class, Computation E: Entity H: Harvesting time, Hypermedia and web application size metrics M: Measurement scale, Metric foundation, Model dependency, Motivation V: Validation	A: Absolute C: Complexity D: Direct E: Early size metric F: Functionality I: Indirect, Interval L: Late size metric, Length M: Media, Motivation N: Nominal, Nonspecific

		O: Ordinal P: Problem-oriented metric, Program/sript R: Ratio S: Solution-oriented metric, Specific V: Validated empirically, Validated theoretically, Validation.both, Validation.none W: Web application, Web hypermedia application, Web software application
Usman et al. [59]	A: Accuracy level, Accuracy measure, Actual effort, Agile methods C: Customer communication D: Distributed teams' issues E: Effort estimate, Effort predictors, Estimated activities, Estimated effort, Estimation context, Estimation entity, Estimation technique, Estimation techniques N: Non functional requirements, Number of entities estimated P: Planning level, Project domain, Project setting S: Size T: Team size, Team's prior experience, Team's skill level, Type U: Unit	A: Accuracy level.value, Accuracy measure.not used, Accuracy measure.other, Actual effort.value, Analogy, Analysis, Availability B: Bias of relative error C: Communications industry, Considered without any metric, Crystal, Customer communication.considered, Customer communication.not considered, Customized extreme programming, Customized scrum D: Design, Distributed teams.considered, Distributed teams.not applicable, Distributed teams.not considered, Distributed: close onshore, Distributed: distant onshore, Distributed: far offshore, Distributed: near offshore, Distribution type, Dynamic systems development method E: Education, Estimate value(s), Estimated activities.all, Estimation entity.other, Estimation technique.other, Expert judgement, Extreme programming F: Feature-driven development, Financial, Function points G: Government/military H: Health, House/days I: Ideal hours, Implementation K: Kanban M: Maintainability, Maintenance, Manufacturing, Mean magnitude of relative error, Median magnitude of relative error N: Non functional requirements.not considered, Non functional requirements.other, Not used effort predictors, Number of entities estimated O: Other effort predictors, Other type P: Pair/days, Performance, Planning level.bidding, Planning level.daily, Planning level.release, Planning level.sprint, Planning poker, Point type, Project setting.co-located, Project somain.other R: Reliability, Retail/wholesale S: Scrum, Security, Story points T: Task, Team size.value, Team's prior experience.considered, Team's prior experience.not considered, Team's skill level.considered, Team's skill level.not considered, Testing, Three point type, Transportation, Type.group, Type.single U: Unit.other, Use case, Use case points method, User case points, User story

Table 24: Summary Table with the evidence used to answer RQ1

Appendix B

The following text is quoted verbatim from Pressman & Maxim [45] (p. 7) to provide a detailed overview of software application domains:

“Software Application Domains

Today, seven broad categories of computer software present continuing challenges for software engineers:

System software. A collection of programs written to service other programs. Some system software (e.g., compilers, editors, and file management utilities) processes complex, but determinate, information structures. Other systems applications (e.g., operating system components, drivers, networking software, telecommunications processors) process largely indeterminate data.

Application software. Stand-alone programs that solve a specific business need. Applications in this area process business or technical data in a way that facilitates business operations or management/technical decision making.

Engineering/scientific software. A broad array of “number-crunching” or data science programs that range from astronomy to volcanology, from automotive stress analysis to orbital dynamics, from computer-aided design to consumer spending habits, and from genetic analysis to meteorology.

Embedded software. Resides within a product or system and is used to implement and control features and functions for the end user and for the system itself. Embedded software can perform limited and esoteric functions (e.g., key pad control for a microwave oven) or provide significant function and control capability (e.g., digital functions in an automobile such as fuel control, dashboard displays, and braking systems).

Product-line software. Composed of reusable components and designed to provide specific capabilities for use by many different customers. It may focus on a limited and specific marketplace (e.g., inventory control products) or attempt to address the mass consumer market.

Web/mobile applications. This network-centric software category spans a wide array of applications and encompasses browser-based apps, cloud computing, server-based computing, and software that resides on mobile devices.

Artificial intelligence software. Makes use of heuristics to solve complex problems that are not amenable to regular computation or straightforward analysis. Applications within this area include robotics, decision-making systems, pattern recognition (image and voice), machine learning, theorem proving, and game playing.

Millions of software engineers worldwide are hard at work on software projects in one or more of these categories. In some cases, new systems are being built, but in many others, existing applications are being corrected, adapted, and enhanced. It is not uncommon for a young software engineer to work on a program that is older than he is! Past generations of software people have left a legacy in each of the categories we have discussed. Hopefully, the legacy to be left behind by this generation will ease the burden on future software engineers.”

Appendix C

Taxonomy	Pub. Year	Software application Domain(s) targeted	References
(A) Mendes et al. [29]	2005	Web/Mobile applications	None; first one to be published
(B) Britto et al. [27]	2016	Application software, Embedded software	None; different focus from (A) but ideally should have referenced (A) in the related work
(C) Britto et al. [28]	2017	Web/Mobile applications	Referenced (A) but not (B); different focus from (B) but ideally should also have referenced (B) in related work
(D) Usman et al. [58]	2017	Application software, Embedded software	None; same applications as (B) and ideally should have referenced (A)-(C) in the related work.
(E) Bajta & Idri [72]	2019	Application software, Embedded software	None; same applications as (B) and (D) and not even these two were referenced in the related work. Complete ignorance of previously published taxonomies for SEE.
(F) Dashti & Gandomani [73]	2021	Not specified	None; complete ignorance of previously published taxonomies for SEE.

Table 25: Analysis that used The References on each primary study