
TOWARDS CONTINUOUS SYSTEMATIC LITERATURE REVIEW IN SOFTWARE ENGINEERING

A PREPRINT

Bianca Minetto Napoleão

Université du Québec à Chicoutimi
Chicoutimi, QC, Canada
bianca-minetto.napoleao1@uqca.ca

Fabio Petrillo

École de Technologie Supérieure
Montreal, QC, Canada
fabio@petrillo.com

Sylvain Hallé

Université du Québec à Chicoutimi
Chicoutimi, QC, Canada
shalle@acm.org

Marcos Kalinowski

Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, RJ, Brazil
kalinowski@inf.puc-rio.br

ABSTRACT

New scientific evidence continuously arises with advances in Software Engineering (SE) research. Conventionally, Systematic Literature Reviews (SLRs) are not updated or updated intermittently, leaving gaps between updates, during which time the SLR may be missing crucial new evidence. **Goal:** We propose and evaluate a concept and process called Continuous Systematic Literature Review (CSLR) in SE. **Method:** To elaborate on the CSLR concept and process, we performed a synthesis of evidence by conducting a meta-ethnography, addressing knowledge from varied research areas. Furthermore, we conducted a case study to evaluate the CSLR process. **Results:** We describe the resulting CSLR process in BPMN format. The case study results provide indications on the importance and feasibility of applying CSLR in practice to continuously update SLR evidence in SE. **Conclusion:** The CSLR concept and process provide a feasible and systematic way to continuously incorporate new evidence into SLRs, supporting trustworthy and up-to-date evidence for SLRs in SE.

Keywords Systematic Review Update · Continuous Systematic Review · Systematic Literature Review · SLR Update Process

1 Introduction

Over the last ten years, Systematic Literature Reviews (SLRs) updates have been conducted in the Evidence-Based Software Engineering (EBSE) area. According to Mendes *et al.* [1], an SLR update is a more recent (updated) version of an SLR that includes new evidence (primary studies). They may also include new methods such as new quality criteria to evaluate evidence, different search strategies to detect new evidence and repeat or remake (using a more recent synthesis method, for example) the analysis of the original SLR.

One known challenge in evidence-based disciplines is to keep SLRs updated. As stated in Cochrane's handbook [2], an SLR that is not maintained may become out-of-date or misleading. Furthermore, with the advance of the computer science field and the growth of research publications, new evidence continuously arises. This fact impacts directly on the purpose of keeping SLRs up to date. An SLR that is out-of-date could lead researchers to obsolete conclusions or decisions about a research topic [3].

In the medicine field, SLR update has a consolidated process [4]. Despite the effort of the Software Engineering (SE) community to keep SLRs updated, a recent study [1] showed that by 2019 only 20 SLRs in SE were updated since the start of SLR publication in 2004. In February 2021, the SE area had more than 1000 secondary studies (SLRs and Systematic Mappings (SMs)) published in several SE venues [5].

Creating and maintaining up-to-date SLRs demands a significant effort for reasons such as the rapid increase in the amount of evidence and the limitation of available databases [6]. Furthermore, the lack of detailed protocol documentation and data availability makes the SLR update process even more difficult since most of the tacit knowledge from the SLR conduction is lost [7].

Conventionally, SE SLRs are not updated or updated intermittently [8]. Periodically updating leaves gaps between updates, during which time the SLR may be missing crucial new research, placing it at risk of being inaccurate and wasting the potential contribution of new research to evidence synthesis and decision-making. Research efforts are needed to remedy these knowledge gaps in updating SLRs and ascertain the potential benefits of continuously assessing new evidence and evaluating the need of updating SLRs.

Inspired by the Living Systematic Review (LSR) [9] idea from medicine and considering the DevOps concept [10] from software development, especially the Continuous Integration and Delivery (CI/CD) processes and practices as well as OS practices, in this study, we propose the concept and process of Continuous Systematic Literature Review (CSLR) in SE aiming to support keeping SLRs constantly updated. Therefore, we compared, analyzed, interpreted and translated findings of studies through the conduction of a meta-ethnography [11]. We also conducted a case study to evaluate the feasibility of applying the proposed CSLR concept and process.

The contributions of this study are: (i) the proposition of the CSLR in SE concept; (ii) the CSLR process definition from the meta-ethnography conduction; and (iii) a case study to evaluate the feasibility of the CSLR process. Our main findings indicate that the CSLR concept and process provide an innovative and systematic way that can be applied to help maintaining SLRs, supporting continuously considering trustworthy and up-to-date evidence for SLRs in SE.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes the study design and the meta-ethnography conduction. Section 4 presents the case study. Section 5 discusses our findings and highlights potential implications. The threats to validity are described in Section 6. Section 7 concludes our study.

2 Related Work

In SE, there are initiatives on SLR updates such as on formalizing an SLR update process [12], searching for new/updated evidence [8, 13], selecting updated evidence [3], deciding on whether to update or not [1], and experience reports [7, 14]. However, searching for new evidence, selecting evidence, deciding upon updating, and enacting the update process following lessons learned from experience reports are important pieces of a problem that has not yet been explicitly addressed as a whole: the problem of leaving gaps between the SLR publication and possible updates.

A recent study [15] proposes a sustainability view for SLR and SLR updates. It states the need for SE community efforts to promote the aspects: (i) social – researcher’s communication and participation during the SLR conduction and update; (ii) economic – resources reduction during the SLR conduction and update; and (iii) technical – supporting tools and technologies to conduct and update SLR. Our study proposes and evaluates a process that considers these three aspects.

3 Towards CSLR

In medicine, in order to mitigate the SLR updating issue, Elliot *et al.* [9] introduced the concept of Living Systematic Review (LSR). An LSR is an SLR that is continually updated, incorporating relevant new evidence as it becomes available.

In software engineering, DevOps concepts (mindset, practices, and tools) were introduced to promote several benefits, such as the faster release of features, improved monitoring systems in production, stimulating collaboration among team members, and others [10]. In addition, the CI DevOps practice aims to build and integrate all working versions of the software code, keeping the software updated. It includes automation, for example, a building service, and a cultural mindset such as integrating changes constantly [16]. Maintaining an SLR up to date requires protocol changes, searching for new evidence, and management strategies to support the demand for updating. Thus, concepts inspired in DevOps concepts could represent an alternative to support continuous changes in SLRs to keep them up to date and with reliable results.

Moreover, an important initiative in the SE research community is promoting Open Science (OS). It consists of making research artifacts available to the public addressing open access, open data, and open-source practices [17]. OS practices can directly impact SLR conduction, not limited to the access and availability of primary evidence for the conduction of SLRs, but in their adoption during the conduction of SLRs that reflect on the reproducibility and maintainability of the SLRs. Thus, researchers should address OS practices in any EBSE study.

To address the problem of leaving gaps between the SLR publication and possible updates, we introduce the CSLR concept. CSLR comprises a continuous and systematic surveillance and analysis of potential new relevant evidence for published SLRs, contributing to keeping SLRs up to date. We designed a CSLR process by applying meta-ethnography, considering elements of the traditional SE SLR process [18], concepts from studies addressing supporting activities involved in updating SLRs in SE [1, 8, 19], LSR from medicine [9], DevOps [10, 16] and OS practices in SE [17].

3.1 Study Design

In order to reach and validate our study goals, we divided it into two phases: (A) Creation of the CSLR process; and (B) Evaluation of the CSLR process. Figure 1 illustrates a summary of our study design.

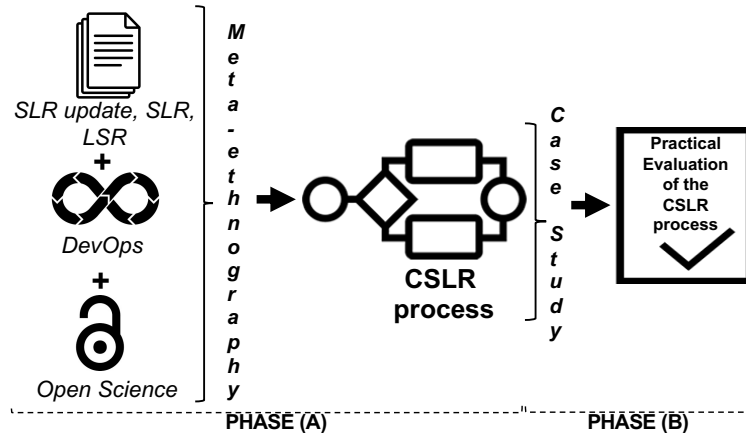


Figure 1: Study design summary

The research method adopted in phase (A) is meta-ethnography [11] since it enables a systematic qualitative synthesis and detailed understanding of how topics and studies are related. According to Hannes & Lockwood [20], the meta-ethnography method provides a picture of the whole phenomenon under investigation from studies and its parts. In this sense, the resulting picture of our meta-ethnography is the CSLR process and its steps. The meta-ethnography method, in summary, involves researchers selecting, analyzing and interpreting qualitative studies through a process of translation, which provides an interpretation of the entire topic, in order to answer questions, gain new insights and/or build knowledge on a specific topic [11].

During phase (B) we performed a case study [21] with a well-known SLR in SE to evaluate the CSLR process observing its contributions to mitigating the SLR intermittent update issues. The details about the conduction of each phase are described in Sections 3.2 and 4 respectively.

3.2 Application of the meta-ethnography method

We present the seven stages of the meta-ethnography [11] with a brief description and the application results from each of the stages.

3.2.1 Getting started

During this first stage, the goal is to identify a topic of interest to be qualitatively explored and then define a Research Question (RQ) that represents the topic and guides the research [11]. In this study, we aim to understand how concepts from the SLR process, SLR update, LSR, DevOps, and OS are related and how they can be integrated to mitigate intermittent SLR update issues in SE.

RQ: How are the SLRs process, SLR update, LSR, DevOps and OS concepts related and how can they be integrated to help mitigate intermittent SLR update issues in SE?

To answer our RQ, we synthesized intersections and relationships of these concepts to create the CSLR process.

3.2.2 Deciding what studies are relevant to the topic of interest

In the second stage, the goal is to find and select relevant studies to the topic of interest. It includes searching and selecting studies to be analyzed [11]. Firstly, we performed a systematic search on Scopus using the terms: (“systematic

literature review process” OR “*systematic review process*” OR “*systematic literature review guideline*” OR “*systematic review update*” OR “*SLR Update*”) AND “*software engineering*”) to detect studies that address the SLR process and SLR update strategies. Secondly, we searched for studies and guidelines that addresses LSR on Scopus using the term: (“*living systematic review*”) and also in two renown medicine databases: PubMed and Cochrane. We performed both searches in February 2022. Thirdly, since our goal is to synthesize information from DevOps practices, we opted to use as a base for our analysis two widely adopted books that describe the DevOps process [10] and the CI/CD pipeline [16]. Finally, for OS practices, we considered the book chapter of Mendez *et al.* [17] since it presents a recent overview of OS practices in SE.

In order to select relevant studies on SLR, SLR update and LSR, we defined three Inclusion Criteria (IC) and three Exclusion Criteria (EC): (IC1) The study proposes or discusses a process or elements of a process on SLR Update in SE or LSR in medicine; OR (IC2) The study is a guideline for the conduction of SLR, SLR Update in SE or LSR in medicine; OR (IC3) The study is an experience report on SLR update in SE. (EC1) The study is an SLR, SLR update or LSR, but it does not discuss any step or aspect of the SLR, SLR update and LSR processes; OR (EC2) The study is an experience report on SLR conduction; OR (EC3) The study is an older version of another study already considered; OR (EC3) The study is not written in English.

A total of 298 studies addressing SLR, SLR update (41 studies) and LSR (257 studies) were retrieved during the search process. First, we excluded duplicated studies, and then we applied the IC and EC on the title, abstract and keywords of these studies. As a result, we selected 17 candidate studies in this stage. Next, we read and applied the IC and EC on the full text of the candidate studies, and selected 12 studies in this stage. We also performed an iteration of the backward and forward snowballing [22] to identify additional studies through the list of references and citations of the selected studies. As a result, we selected three more studies. Thus, a total of 15 studies compose our final set of studies. Table 1 compiles in a list all 18 studies (S1 – S18), selected for the following stages of the meta-ethnography process execution. It includes our final set of 15 selected studies categorized by their main topic and the books and books’ chapter previously selected (+3).

Table 1: Selected studies for the next stages of the meta-ethnography

| ID | Main topic | Publication Year | Reference |
|-----|--------------|------------------|-----------|
| S1 | SLR update | 2020 | [3] |
| S2 | SLR update | 2020 | [8] |
| S3 | SLR update | 2020 | [1] |
| S4 | SLR update | 2020 | [7] |
| S5 | SLR update | 2019 | [23] |
| S6 | SLR update | 2018 | [24] |
| S7 | SLR update | 2017 | [14] |
| S8 | SLR update | 2016 | [13] |
| S9 | SLR update | 2008 | [12] |
| S10 | SLR process | 2017 | [25] |
| S11 | SLR process | 2015 | [26] |
| S12 | SLR process | 2007 | [18] |
| S13 | LSR | 2022 | [27] |
| S14 | LSR | 2019 | [28] |
| S15 | LSR | 2017 | [9] |
| S16 | DevOps | 2015 | [10] |
| S17 | DevOps | 2010 | [16] |
| S18 | Open Science | 2020 | [17] |

3.2.3 Reading the studies

During the third stage, the authors read the set of selected studies and perform the data extraction [11]. In the context of this study, all the authors are familiar with the SLR process and the software development process. Therefore, the authors’ experience and their interactions facilitated the understating of the content of the selected studies. The first author extracted the data using a data extraction form, which was revised by the second author.

The data extraction form was built to obtain (i) the general objective of the study; (ii) its main results and contributions; (iii) if the study proposes an approach or the use of a technique, a description of the approach or technique presented and whether it has been validated; and (iv) if the study presented a process, the description of each process activity including roles, inputs, processing, and outputs.

Analyzing the 18 selected studies, as shown in Table 1, the majority of the selected studies address the SLR Update process in SE since our goal was to mitigate intermittent SLR update issues in SE, which requires a deep understanding on the advancements on SLR updates in SE over the past years. Notably, SLR updates have gained the attention of the SE community. In 2020 four studies proposing improvements in conducting SLR updates in SE were published (S1, S2, S3 and S4). Watanabe *et al.* [3] (S1) propose and evaluate the use of text classification to provide automated support in the SLR update studies selection activity. Wohlin *et al.* [8] (S2) propose guidelines on the search strategy to update SLRs in SE. Mendes *et al.* [1] (S3) recommend using a decision framework on when to update SLRs in SE. Felizardo *et al.* [7] (S4) present an experience report on how to transfer the know-how of SLRs to facilitate their updates through the instantiation of a knowledge management model.

Other five studies addressing SLR updates were selected (S5, S6, S7, S8 and S9). Besides S4, two other experience reports on SLR updates were considered in our study (S5 and S6). Nepomuceno & Soares [23] (S5) present a systematic mapping and survey on how researchers are evolving their SLRs and what they think about SLR updates. Garcés *et al.* [14] (S7) relate the authors' experience in updating two SLRs using automated techniques based on VTM (Visual Text Mining). Felizardo *et al.* [13] (S8) introduced the adoption of forward snowballing [22] to search for studies to update SLRs in SE. Two years later, Felizardo *et al.* [24] (S6) evaluated the use of different electronic databases for applying forward snowballing to update secondary studies. Study S2 presents a combined and more recent investigation addressing the approaches described in S5 and S6. Last but not least, Dieste *et al.* [12] (S9) propose a process to perform SLR updates in SE, taking into account lessons learned from updating an SLR.

Regarding the traditional SLR process, we considered the well-known Kitchenham & Charters [18] guidelines (S12) as well as Kitchenham's *et al.* Book [26] (S11) which contains an extended description of the SLRs process in SE as well as the update of the guidelines proposed in S12. Kuhrmann *et al.* [25] (S10) present an experience-based guideline to aid researchers in designing SLRs in SE with emphasis on the studies' search and selection procedures.

LSRs were introduced in the medical field in 2017 by Elliott *et al.* [9] (S15), aiming to incorporate relevant new evidence to SLRs as it becomes available. Two years later, the Cochrane DL embraced the LSR concept and published its guidelines for the production and publication of Cochrane LSRs [28] (S14). More recently, Simonmds *et al.* [27] (S13) described the general principles of LSR in a book chapter, when they might be of particular value, and how its procedure differs from conventional SLRs.

The three last studies included in our analysis are the book of Bass *et al.* [10] (S16), which describes the DevOps concept through a software architecture perspective detailing each step of the DevOps process. The book of Humble *et al.* [16] (S17) which presents the whole DevOps, explaining in detail the CI/CD practices. Finally, the recently published book chapter by Mendez *et al.* [17] (S18) on OS for SE includes the OS definition, why SE researchers should engage in it, and how they should do it.

3.2.4 Determining how the studies are related

In this fourth stage, using the extracted data from the stage (3) and revisiting the select studies when needed, we extracted the metaphors of each selected study. These metaphors were phrases, ideas, and concepts that could be relevant to detecting relationships or connections among the studies. Next, a list describing all the identified relationships among the studies is created. We used spreadsheets to support this step. When we started to extract the relationships between studies, we noticed that all studies have some process factor that directly impacts the updating of an SLR. We summarize these relationships in Table 2. The process factors are described in the first column of Table 2.

Table 2: Relationships among the selected studies

| Process factor | Related studies |
|-------------------|--|
| Planning | S3, S4, S5, S7, S9, S10, S11, S12, S13, S14, S15, S16, S17 |
| Study selection | S1 |
| Search strategy | S2, S6, S8 |
| Executing | S1, S2, S3, S4, S5, S6, S7, S9, S10, S11, S12, S13, S14, S15, S16, S17 |
| Reporting | S4, S5, S9, S10, S11, S12, S3, S14, S15, S16, S17, S18 |
| Monitoring | S13, S14, S15, S16, S17 |

As shown in Table 2, we detected studies related to specific SLR update planning factors: study selection and search strategy. Despite the base of the SLR update process being the original SLR process (planning, executing and reporting), i.e. re-execute (and adapt, if necessary) the original review protocol, we identified that the DevOps concepts and process could be seen as a metaphor for building a process of continuous integration and delivery of evidence to support keeping SLRs up-to-date. Monitoring, a DevOps process factor (S16-S17), is connected to the LSR studies (S13-S14-S15)

which are connected to the other three SLR and SLR update factors (planning, executing and reporting). In addition, the OS study (S18) is connected to the reporting factor since it addresses the availability of information.

3.2.5 Translating the studies into one another

The main goal of this stage is to compare the metaphors (text fragments) extracted from the studies. Unfortunately, Noblit and Hare [11] do not describe how to perform this step, but one suggestion is to compare the synthesis of each study progressively [29].

We analyzed the metaphors observed in stage (4), comparing them and analysing their relationships. To construct the relations among the studies, we opted to guide this construction based on the original SLR process activities as demonstrated in Table 2 because it is also the basis process for updating an SLR. In addition, the LSR process follows the same base structure as the SLR process. In this sense, first, we build a table based on the elements of the relationships detected in stage (4). Second, we separated each metaphor into (i) general metaphors: fragments that address the relationship among the studies and procedural elements generally; (ii) specific metaphors: fragments that address specific relations among each activity necessary to perform an SLR update. Finally, we checked our list of metaphors to see if any of them directly mitigated the SLR intermittent update issues. Therefore, we checked if the relationships were found to converge to the objective of our study. We also added the references that support our findings¹.

Analyzing the identified relationships, it is clear that the selected studies' practices, processes, activities, and recommendations are strongly interconnected. Hence, the systematization of the evidence found in this step in the form of a process can contribute to the establishment of a dedicated process for mitigating the intermittent SLR update problem.

3.2.6 Synthesizing the translations

During the sixth stage, we constructed a translation synthesis. The translations of studies result in many metaphors. They are compared to verify similarities and/or if some metaphors can encompass others. The result of this stage is usually represented as diagrams or figures [11]. In the context of our study, we compared and systematized the translations mapped in step (5) as a continuous process for assessing new evidence and evaluating the need of update SLRs (Continuous Systematic Review – CSLR). Figure 2 illustrates the three stages of the CSLR process (Integration, Delivery and Observability) with their phases and activities. The CSLR BPMN diagram is also available online¹.

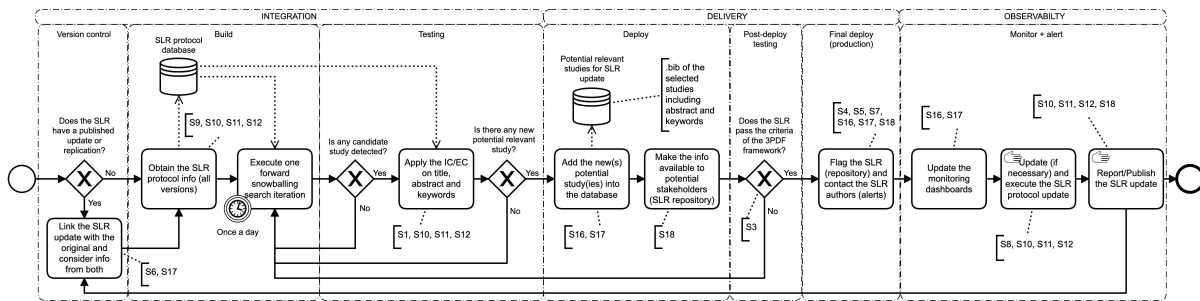


Figure 2: CSLR process

The CSLR process starts with the *Integration* stage and *Version Control* phase (see Figure 2). In this phase, the first activity is to verify if the SLR has an update or replication published. If yes, these studies must be linked and considered in the next process activity.

In the second phase of the *Integration* stage, the *Build* phase, the first activity is to obtain the protocol information of all considered studies (original SLR, updates, replications – if it exists) and store them in a database. These protocols information include: Addressed research questions, the period covered by the SLR execution, list of included studies, inclusion and exclusion criteria, and quality criteria (if adopted).

In the next activity of the *Build* phase, using the original SLR and its list of included studies, one iteration of the forward snowballing using the Google Scholar as DL is performed [8] once a day. If there is an SLR update and/or replications linked to this SLR, these studies and their list of included studies must be considered in the snowballing execution.

¹<https://doi.org/10.5281/zenodo.6503143>

The last phase of the *Integration* stage, the *Testing* phase, starts checking if any study was detected by the snowballing process (candidate studies). If no study was detected, the process returns to forward snowballing execution activity. If one or more studies were detected, the IC and EC criteria must be applied to the candidate studies' title, abstract, and keywords. The study of Watanabe *et al.* [3] proposed an automated alternative for this step using text classification techniques. The next activity is to verify if there is any potential relevant study according to the IC and EC. If not, the process returns to the activity of snowballing execution. If yes, the process moves to the *Delivery* stage.

The *Delivery* stage starts with the *Deploy* phase. The first activity in this phase is to add the new potential relevant studies to the database. In order to deal with automation [3] and to allow using reference management tools (such as Jabref and Zotero), the studies should be stored in the database in .bib format. The next activity is to make these potential relevant studies available to potential stakeholders. For that, we suggested uploading this information into a public repository such as Zenodo or ArXiv [17].

The next phase, the *Post-deploy testing*, has one activity: apply the 3PDF framework proposed in [1] to analyse if the SLR needs to be updated. The 3PDF framework consists in answering 7 questions (steps) as described in [1]. If the results of the 3PDF show that the SLR is suitable for an update, in the *Final deploy* phase the SLR must be flagged as "outdated" and the authors of the original SLR, update and/or replication (if it exists) will be contacted. Otherwise, the process returns to the activity of defining the snowballing forward frequency.

The last stage of the CSLR process is the *Observability* stage. This stage has an unique phase: *Monitor + Alert*. The two first activities of this stage are conditioned to process automation, i.e., the development of a dedicated SLR repository. It consists of updating the monitoring dashboards that report the SLR update data gathered during the CSLR process execution. The last two activities consist in performing the SLR update. It includes updating protocol items (if necessary) [1] and executing it; and reporting the results of the SLR update (e.g., publishing) [26]. Finally, with the first process cycle finished, the new and completed SLR updated must be linked to its other version(s), re-starting the CSLR process from its beginning.

3.2.7 Expressing the synthesis

In this last stage, the synthesis findings are disseminated to interested parties. In our case this concerns making the CSLR process available as a direction for future research on continuously keeping SLRs in SE up-to-date.

4 Applying CSLR to a Published SLR

To evaluate the CSLR process, we selected a suitable SLR as an instrument to execute the process through the conduction of a case study. Our goal is to evaluate the feasibility of applying the proposed CSLR process and observe its contributions to mitigating the SLR intermittent update issues. We translated our case study goal into two Research Questions (RQs): **RQ1:** *How do the steps of the proposed CSLR process perform in practice?* **RQ2:** *Can the CSLR process help to mitigate the intermittent SLR update issue in SE?*

We follow the five main steps for conducting case studies proposed by [21]: Design, preparation, collecting data, analysis and reporting. These steps are described hereafter.

4.1 Design

Our design consists in selecting an SLR to be the instrument (input) of the CSLR process and then executing the CSLR process steps manually.

We chose the SLR [30] which addresses the topic of cross-company vs within-company effort estimation, for the following reasons: (i) The SLR is published as a conference paper [31], and after as a journal paper [30] in renowned SE venues; (ii) The SLR has being used as evaluation instrument by several other studies [1, 8, 13, 22]; (iii) The SLR was last updated in 2014 (over seven years ago) [32] – with this, we can also evaluate the CSLR process when the SLR already has a published update; and (iv) The SLR update has as co-author one of the authors of this study.

4.2 Preparation

We distributed the conduction of the case study between the first and last authors of this study. Both authors have experience in conducting SLRs and updates. The last author is co-author of the SLR Update [32]. According to SLR experience reports [7, 14, 23], the participation of a member who already participated in the previous review can facilitate the update process besides contribute to avoiding bias.

4.3 Collecting data

Data collection is based on the SLR chosen for the CSLR process evaluation. The CSLR process has several data collection activities that must be carried out according to the process execution (e.g. check if there is a published SLR update, obtain the list of included studies).

4.4 Analysis

We report our analysis on how the process steps perform in practice by describing the case of applying each CSLR process activity to the selected SLR.

The first activity of CSLR involves verifying if our SLR candidate has a published update. For that, we checked the citations of the SLR on Google Scholar (428 citations). As a result, we identified an update [32] (SLR-Update) published in 2014, as well as two other studies published in 2016 that replicated the SLR update investigating different search strategies [22] (SLR-UR1) and [13] (SLR-UR2). Therefore, SLR-Update, SLR-UR1 and SLR-UR2 were selected in the version control phase (*Integration* stage) of the CSLR process execution. It is important to mention that these same three studies were also identified in [8] that used this same SLR as an investigation instrument.

The next phase of the CSLR process (*Build*) begins with obtaining protocol information from the studies selected in the version control phase. With the support of a spreadsheet, we extracted the following information from the original SLR, SLR-Update, SLR-UR1 and SLR-UR2: (i) *Research questions* – all studies investigated the same research questions; (ii) *IC, EC and quality criteria* – the IC, EC and quality criteria were the same for all studies; (iv) *Search strategy* – the original SLR performed automated search on six SE Digital Libraries (DLs), manual search on individual journals and conference proceedings and reference checking (a.k.a backward snowballing). The SLR-Update used the same method as the original SLR except for adding Scopus as an extra DL and not performing a manual search. SLR-UR1 adopted backward and forward Snowballing [22] and SLR-UR2 only forward snowballing; (v) *Search strategy coverage period* – the original SLR covered the period from 1990 to November 2006, the SLR-Update from December 2006 to end 2013 and SLR-UR1 and SLR-UR2 both from 2006 to 2013; and (vi) *List of included studies* – The original SLR included ten primary studies, the SLR-Update 11 additional primary studies. The SLR-UR1 did not include two studies included in the SLR-Update, but it included three other studies that did not include in the SLR-Update. The SLR-UR2 included all the studies included in the SLR-Update except for one study, and it also identified two other studies. Considering this scenario, for the CSLR, we considered all unique included studies from the original SLR, SLR-Update, SLR-UR1 and SLR-UR2, totalling a final list with 25 included studies published by the end of 2013.

The second activity in the *Build* phase is to execute one interaction of forward snowballing search technique on Google Scholar to identify new potential relevant studies. Using the 25 included studies and the original SLR, SLR-Update and SLR-UR1 and SLRU-R2 as seed, we performed an interaction of the forward snowballing technique we obtained a total of 2392 returned studies. Since the SLR-Update and both replications covered the search until 2014, we limited our search results from 2014 to February 2022, resulting in 858 returned studies. We exported the bibliographical data, including keywords and abstracts of all studies in .bib and .csv format.

Moving to the *Testing* phase, the first step is to perform the initial cleaning of our set of returned studies. Since this process has been executed manually, we used the .csv file to remove duplicated studies and conference announcements. Thus, we arrived at a list of 444 unique candidate studies. Next, we applied the inclusion and exclusion criteria on the title, abstract and keywords of each study arriving in a set of 24 potential candidate studies. It is worth mentioning that all 24 studies contain at least one or more keywords that would allow an automated selection method such as [3] to identify these studies. The first and last author carefully analyzed the title, abstract and keywords of each study, and in a synchronous meeting, they decided through consensus what studies have a strong potential for inclusion. As a result, five new studies showed significant potential to be included in the SLR, and other five studies presented a new trend of investigations using cross-company and within-company mixed together to estimate software project effort (which could lead to updating the SLR protocol, including its research questions, to properly consider this new identified trend). The list with the 24 potential candidate studies and the ten selected potential studies to be included in a new update is available online¹.

The *Deploy* phase starts by adding the potential selected studies into a database and next making them available online in a repository. Since we are performing both activities manually, we created a .bib file with the ten potential included studies. We made them available at Zenodo¹, an open dissemination research data repository.

The *Post-deploy testing* phase consists of using the 3PDF framework [1] to verify if the SLR needs to be updated. Thus, we performed the seven steps (questions) of the 3PDF method. The answer to each question is available online¹. As a result, the SLR needs to be updated.

In the *Delivery* stage, during the final deployment, we must flag the SLR with the status “SLR suitable for update” and communicate the SLR and SLR-Update (if exists) authors about the findings. As this is a feasibility study, we did not contact anyone with regard to these results yet.

The last stage of the CSLR process is the *Observability* stage. The first activity described in the *Monitor + alert* phase, flag the SLR and contact the authors, is not suitable for performing manually. The activities of performing and publishing the SLR update can be performed without these activities. However, updating the SLR is out of the scope of this study. Hence, since the SLR is not updated yet, the process returns to the *Build* phase to search for new potential relevant studies.

4.5 Reporting

RQ1: *How do the steps of the proposed CSLR process perform in practice?*

In general, the CSLR process flow seemed to be coherent. The stages, phases and activities were manually applicable in practice, addressing relevant aspects to help mitigating intermittent SLR update issues in SE.

However, when it comes to automation aspects, while some activities are clear candidates for automation (e.g., the whole CSLR systematic surveillance and analysis of new potentially relevant studies), not all of them seem to be ready for complete automation yet. For instance, the application of some decision steps of the 3PDF framework, which still require human intervention and reasoning. Also, the update itself, including the rigorous full-text-based assessment, application of the quality assessment criteria, data extraction, and research synthesis, is seen as a manual activity to be conducted by researchers, as depicted in Figure 2.

RQ2: *Can the CSLR process help to mitigate the intermittent SLR update issue in SE?*

During the execution of the case study, (manually) applying the CSLR process activities showed being feasible and enabled systematic surveillance and analysis of new potentially relevant studies. Furthermore, its application revealed the need to update an already published SLR. Hence, we conclude that the CSLR process can help to mitigate the intermittent SLR update issue.

With proper automation (e.g., including SLR repositories, continuous integration facilities, and availability of information in an open and accessible way, monitoring dashboards), CSLR could help to systematically identify the need for updates and unnecessary SLR updates, as reported in [1], could be avoided. Hence, we perceive making the CSLR process available as important to foster research in this direction.

5 Discussion

The motivation for the elaboration of the CSLR process is to provide a systematic process to help mitigating the intermittent SLR update problem contributing to avoiding missing new potential new relevant research in evidence-syntheses or decision-making. According to Nepomuceno *et al.* [23], actions to keep SLR updated are of great importance to the SLR research field.

The CSLR process unifies several pieces of knowledge on SE SLR updates that have been investigated separately, integrating them with DevOps and OS concepts. As a consequence, we observed benefits such as (i) facilitating the identification if the SLR has been updated or not (ii) assisting the identification (search and selection) of potentially relevant evidence; (iii) making potentially relevant evidence available in open repositories that are freely accessible by the SE community; (iv) supporting the decision on the need to update an SLR; and (v) supporting SLR authors throughout the update process.

One may question the similarities between LSR and CSLR. Indeed, both have the same aim of keeping the SLR up to date [9, 27]. LSR is an medicine review approach to update an SLR continually (fully manually) requiring authors to make explicit commitments as to the frequency of search and screening execution as well as publication updates [2]. On the other hand, inspired in SE DevOps and OS practices, the CSLR process explores automation opportunities based on study findings investigated in the SE area on SE SLR updates (e.g. [1, 8]) aiming to provide continuous and systematic surveillance and analysis of potential new relevant evidence. The need for an update is then decided by using the 3PDF framework. Furthermore, the idea of making the SLR protocol information and intermediate results openly available is not explored within the LSR context. LSR relates to continual SLR updates, while CSLR aims at continuous SLR updates.

In a more general view, the CSLR could also be an instrument to direct the SE community on research subjects that have been investigated – if an SLR is often cited, it gives evidence that the subject of study addressed by it is constantly evolving. This fact leads to questions such as: Does the SLR remain relevant? Is the SLR up to date? Does it needs to

be updated? – As observed in our case study, new research trends can be identified, leading to the proposition of other research directions (questions) on a research subject.

Besides, the CSLR process showed to be effective during our evaluation and opens avenues for automating its activities and pipeline. Researchers have investigated automation of the SLR process over the years [33]. However, to the best of our knowledge, only two studies are addressing automation alternatives for SLR updates [3, 19]. Moreover, both studies are focused only on the study selection activity. Therefore, there is a lack of approaches that automate or semi-automate other SLR update activities or approaches that integrate the update activities.

6 Threats to Validity

Construct validity. We followed well-known guidance and advice on designing and conducting meta-ethnography studies [11] and case studies in SE [21]. We also carefully selected the SLR to apply CSLR (cf. Section 4.1).

Internal validity and reliability. Since meta-ethnography is an interpretive approach to synthesis, we addressed the validity and reliability of our synthesis by performing discussions among the authors during the conduction of all seven steps of the method, and we evaluated the outcome (CSLR process) through a case study. Regarding the case study, it was conducted by the first and last author of this study. The last author is co-author of the SLR Update [32] to which CSLR was applied, which avoids misunderstandings with regard to the SLR protocol [23].

External validity. Similar case studies could have been applied to more SLRs to improve the generalizability of our results. However, manually applying the process involves significant effort. Nevertheless, we intend to extend this study by addressing more SE SLRs, also counting on the participation of external researchers to investigate the CSLR process.

7 Conclusion

In this study, we proposed and evaluated the Continuous Systematic Literature Review (CSLR) concept and process to support SLRs updates in SE. We structured the CSLR process by synthesizing evidence through a meta-ethnography integrating knowledge from varied research areas.

The CSLR concept and process comprises a continuous and systematic surveillance and analysis of potential new relevant evidence for published SLRs in a timely manner, contributing to keeping SLRs up to date. The results of our evaluation suggest that CSLR can help to mitigate the relevant SLR intermittent update issue for SLRs in SE. Besides, the CSLR process provides a systematic pipeline towards automating and managing SLR updates activities.

Hence, we believe that making the CSLR process available can foster future work investigating more effective ways for continuously summarizing existing SE evidence. For instance, further evaluating the CSLR process and investigating automating its process pipeline.

References

- [1] E. Mendes, C. Wohlin, K.R. Felizardo, and M. Kalinowski. When to update systematic literature reviews in software engineering. *Journal of Systems and Software*, 167:110–167, 2020.
- [2] Julian Higgins, James Thomas, Jackie Chandler, Miranda Cumpston, Tianjing Li, Matthew Page, and Vivian Welch. Cochrane handbook for systematic reviews of interventions. *Cochrane Handbook for Systematic Reviews of Interventions*, 09 2019.
- [3] Willian Massami Watanabe, Katia Romero Felizardo, Arnaldo Candido, Erica Ferreira de Souza, Jo ao Ede de Campos Neto, and Nandamudi Lankalapalli Vijaykumar. Reducing efforts of software engineering systematic literature reviews updates using text classification. *Information and Software Technology*, 128, 2020.
- [4] David Moher, Alexander Tsertsvadze, Andrea Tricco, Martin Eccles, Jeremy Grimshaw, Margaret Sampson, and Nick Barrowman. When and how to update systematic reviews. *Cochrane database of systematic reviews*, 1, 02 2008.
- [5] Bianca M. Napoleão, Katia R. Felizardo, Érica F. de Souza, Fabio Petrillo, Sylvain Hallé, Nandamudi Vijaykumar, and Elisa Nakagawa. Establishing a search string to detect secondary studies in software engineering. In *(SEAA)*, pages 9–16, 2021.
- [6] L. Zhang, Jia-Hao Tian, Jing Jiang, Y. Liu, Meng-Yuan Pu, and Tao Yue. Empirical research in software engineering — a literature survey. *Journal of Computer Science and Technology*, 33:876–899, 2018.

- [7] Katia R. Felizardo, Erica F. de Souza, Tamiris Malacrida, Bianca M. Napoleao, Fabio Petrillo, Sylvain Hallé, Nandamudi L. Vijaykumar, and Elisa Y. Nakagawa. Knowledge management for promoting update of systematic literature reviews: An experience report. In *(SEAA)*, pages 471–478, 2020.
- [8] Claes Wohlin, Emilia Mendes, Katia Romero Felizardo, and Marcos Kalinowski. Guidelines for the search strategy to update systematic literature reviews in software engineering. *Information and Soft. Technology*, 127, 2020.
- [9] Julian Elliott, Anneliese Synnot, Tari Turner, Mark Simmonds, Elie Akl, Steve McDonald, and et. al. Living systematic review 1: Introduction - the why, what, when and how. *Journal of Clinical Epidemiology*, 91, 2017.
- [10] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 1st edition, 2015.
- [11] G.W. Noblit and R.D. Hare. *Meta-Ethnography*. Sage Publications Inc, Newbury Park, Calif., 1988.
- [12] O. Dieste, M. López, and F. Ramos. Formalizing a systematic review updating process. In *(SERA)*, pages 143–150, 2008.
- [13] Katia Felizardo, Emilia Mendes, Marcos Kalinowski, Erica Ferreira Souza, and Nandamudi Vijaykumar. Using forward snowballing to update systematic reviews in software engineering. In *(ESEM)*, 2016.
- [14] Lina Garcés, Katia Felizardo, Lucas Oliveira, and Elisa Nakagawa. An experience report on update of systematic literature reviews. pages 91–96, 07 2017.
- [15] Vinicius dos Santos, Anderson Yoshiaki Iwazaki, Katia Romero Felizardo, Érica Ferreira de Souza, and Elisa Yumi Nakagawa. Towards sustainability of systematic literature reviews. In *(ESEM)*, New York, NY, USA, 2021. ACM.
- [16] Jez Humble and David Farley. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional, 1st edition, 2010.
- [17] Daniel Mendez, Daniel Graziotin, Stefan Wagner, and Heidi Seibold. *Open Science in Software Engineering*, pages 477–501. Springer Int. Publishing, 2020.
- [18] B.A. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University, UK, 2007.
- [19] K.R. Felizardo, E.Y. Nakagawa, S. MacDonell, and J.C. Maldonado. A visual analysis approach to update systematic reviews. In *(EASE)*, pages 1–10. ACM, 2014.
- [20] Karin Hannes and Craig Lockwood. *Synthesizing Qualitative Research: Choosing the Right Approach*. 02 2012.
- [21] P. Runeson, M. Host, and A. Rainer. *Case Study Research in Software Engineering: Guidelines and Examples*. 2012.
- [22] Claes Wohlin. Second-generation systematic literature studies using snowballing. In *(EASE)*, pages 15:1–15:6. ACM, 2016.
- [23] Vilmar Nepomuceno and Sergio Soares. On the need to update systematic literature reviews. *Inf. and Soft. Technology*, 109:40–42, 2019.
- [24] Katia Romero Felizardo, Anderson Y. Iwazaki da Silva, Érica Ferreira de Souza, Nandamudi L. Vijaykumar, and Elisa Yumi Nakagawa. Evaluating strategies for forward snowballing application to support secondary studies updates: Emergent results. In *(SBES)*, page 184–189, New York, NY, USA, 2018. ACM.
- [25] Marco Kuhrmann, Daniel Méndez Fernández, and Maya Daneva. On the pragmatic design of literature studies in software engineering: An experience-based guideline. *Empirical Soft. Eng.*, 22(6):2852–2891, dec 2017.
- [26] B.A. Kitchenham, D. Budgen, and P. Brereton. *Evidence-Based Software Engineering and Systematic Reviews*. Chapman & Hall/CRC, 2015.
- [27] Mark Simmonds, Julian H. Elliott, Anneliese Synnot, and Tari Turner. *Living Systematic Reviews*, pages 121–134. Springer, New York, 2022.
- [28] J Brooker, A Synnot, S McDonald, J Elliott, T Turner, R Hodder, et al. Guidance for the production and publication of cochrane living systematic reviews: Cochrane reviews in living mode. 2019.
- [29] F.Q.B. Silva, S.S.J.O. Cruz, T.B. Gouveia, and L.F. Capretz. Using meta-ethnography to synthesize research: A worked ex. of the relations between personality and soft. team processes. In *(ESEM)*, pages 153–162, 2013.
- [30] B. A. Kitchenham, E. Mendes, and G. H. Travassos. Cross versus within-company cost estimation studies: A systematic review. *IEEE Transactions on Software Engineering*, 33(5):316–329, 2007.
- [31] Barbara Kitchenham, Emilia Mendes, and Guilherme H. Travassos. A systematic review of cross- vs. within-company cost estimation studies. *EASE'06*, page 81–90, Swindon, GBR, 2006. BCS Learning; Development Ltd.

- [32] E. Mendes, M. Kalinowski, D. Martins, F. Ferrucci, and F. Sarro. Cross- vs. within-company cost estimation studies revisited: An extended systematic review. In *(EASE)*, pages 12:1–12:10, 2014.
- [33] Katia R. Felizardo and Jeffrey C. Carver. *Automating Systematic Literature Review*, pages 327–355. Springer International Publishing, 2020.