The Journal of Systems & Software 195 (2023) 111520

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Applying Inter-Rater Reliability and Agreement in collaborative Grounded Theory studies in software engineering☆

Jessica Díaz [a], Jorge Pérez [a], Carolina Gallardo [a], Ángel González-Prieto [b,c,*]

[a] Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos. ETSI Sistemas Informáticos, C/ Alan Turing s/n (Carretera de Valencia Km 7), 28031 Madrid, Spain
[b] Universidad Complutense de Madrid. Departamento de Álgebra, Geometría y Topología. Facultad de Ciencias Matemáticas, Plaza Ciencias 3, 28040 Madrid, Spain
[c] Instituto de Ciencias Matemáticas (CSIC-UAM-UCM-UC3M), C/ Nicolás Cabrera 13-15, 28049 Madrid, Spain

ARTICLE INFO

Article history:
Received 1 January 2022
Received in revised form 8 July 2022
Accepted 26 September 2022
Available online 1 October 2022

Dataset link: https://doi.org/10.5281/zenodo.5034244, https://es.surveymonkey.com/r/PMWD7ZM

Keywords:
Grounded Theory
Inter-Rater Reliability
Inter-Rater Agreement

ABSTRACT

Context: The qualitative research on empirical software engineering that uses Grounded Theory is increasing (GT). The trustworthiness, rigor, and transparency of GT qualitative data analysis can benefit, among others, when multiple analysts juxtapose diverse perspectives and collaborate to develop a common code frame based on a consensual and consistent interpretation. Inter-Rater Reliability (IRR) and/or Inter-Rater Agreement (IRA) are commonly used techniques to measure consensus, and thus develop a shared interpretation. However, minimal guidance is available about how and when to measure IRR/IRA during the iterative process of GT, so researchers have been using ad hoc methods for years.
Objective: This paper presents a process for systematically measuring IRR/IRA in GT studies, when appropriate, which is grounded in a previous systematic mapping study on collaborative GT in the field of software engineering.
Methods: Meta-science guided us to analyze the issues and challenges of collaborative GT and formalize a process to measure IRR/IRA in GT.
Results: This process guides researchers to incrementally generate a theory while ensuring consensus on the constructs that support it, improving trustworthiness, rigor, and transparency, and promoting the communicability, reflexivity, and replicability of the research.
Conclusion: The application of this process to a GT study seems to support its feasibility. In the absence of further confirmation, this would represent the first step in a de facto standard to be applied to those GT studies that may benefit from IRR/IRA techniques.

1. Introduction

Qualitative data collection and analysis techniques are increasingly used in software engineering research (Stol et al., 2016; Wohlin et al., 2012). Among the most popular are the various flavors of Grounded Theory (Glaser and Strauss, 1967; Strauss and Corbin, 1990; Charmaz, 2014). *Grounded Theory* (GT) refers to a family of predominately qualitative research methods for inductively generating theory based on rounds of interleaved data collection and analysis. GT is particularly well suited to explore how software professionals collaborate and create software (Hoda, 2021; Leite et al., 2021; López-Fernández et al., 2021; Luz et al., 2019).

GT involves one or more human analysts reading textual data (e.g. interview transcripts, documents, emails, discussion forum posts, field notes), interpreting and labeling these data (*coding*), recording their thoughts in notes called *memos*, organizing the data and labels into categories, and constantly comparing and reorganizing these categories until a mature or *saturated* theory emerges. More and more frequently, analytical techniques pioneered in the GT literature, such as open, selective, axial, focused, and theoretical coding (Stol et al., 2016), involve multiple researchers to capitalize on the potential benefits of collaborative data analysis (Hall et al., 2005; Guest and MacQueen, 2008; Cornish et al.), defined as"*the processes in which there is joint focus and dialogue among two or more researchers regarding a shared body of data, to produce an agreed interpretation (Cornish et al.) and*

☆ Editor: Burak Turhan.
* Corresponding author.
E-mail addresses: yesica.diaz@upm.es (J. Díaz), jorgeenrique.perez@upm.es (J. Pérez), carolina.gallardop@upm.es (C. Gallardo), angelgonzalezprieto@ucm.es (Á. González-Prieto).

https://doi.org/10.1016/j.jss.2022.111520

a shared understanding of the phenomenon being studied (Saldaña, 2012)".

The benefits associated with conducting collaborative qualitative analysis have been extensively reported in Hall et al. (2005), Cornish et al., Richards and Hemphill (2018) and include: (i) juxtaposing and integrating multiple and diverse perspectives (Olson et al., 2016) (e.g., insider/outsider, academic/practitioner, senior/junior, interdisciplinary and international perspectives (Cornish et al.)), which is often viewed as one way to counteract individual biases (Olson et al., 2016) and enhance/increase credibility (Olson et al., 2016), trustworthiness (Patton, 1999), and rigor (Dubé and Paré, 2003); (ii) addressing large and complex problems (Hall et al., 2005) by effective management of large datasets (Olson et al., 2016); and (iii) effective mentoring of junior researchers (Cornish et al.). In Patton's words "*Having two or more researchers independently analyze the same qualitative data set and then compare their findings provides an important check on selective perception and blind interpretive bias*" (Patton, 1999). However, collaborative qualitative analysis can also be challenging and time-consuming (Hall et al., 2005).

Thus, the trustworthiness, rigor, and transparency of GT qualitative data analysis may benefit, among others, when multiple analysts juxtapose diverse perspectives (Cornish et al.) and collaborate to develop a common code framework based on a consensus and consistent interpretation (Richards and Hemphill, 2018). *Collaborative coding* is said to enforce systematicity, clarity, and transparency (Hall et al., 2005), and enables the assessment of Inter-Rater Reliability (IRR) and/or Inter-Rater Agreement (IRA), which are commonly used techniques to measure consensus, and thus develop a shared interpretation in qualitative research (Armstrong et al., 1997; Weston et al., 2001; Campbell et al., 2013; MacPhail et al., 2016; McDonald et al., 2019; O'Connor and Joffe, 2020). IRA is the extent to which different raters assign the same precise value to each item being rated, whereas IRR is the extent to which raters can consistently distinguish between different items on a measurement scale (Gisev et al., 2013). Briefly, IRA measures agreement, whereas IRR measures consistency; raters may have high consistency but low (or even no) agreement.

However, deciding whether to use quantitative measures like IRR/IRA in qualitative research has little consensus with a lot of researchers in favor as evidence of the rigor of the analysis (Cornish et al.) and in against, mainly based on epistemological objections (cf. McDonald et al., 2019; O'Connor and Joffe, 2020). Additionally, while there are general guidelines for applying IRR/IRA (MacPhail et al., 2016; O'Connor and Joffe, 2020), minimal guidance specific to grounded theory is available beyond simply describing statistical techniques, often due to journals' limitations on the space for discussing methods (Richards and Hemphill, 2018). Another possible reason may be that measuring IRR/IRA in GT studies is complex due to the iterative nature of the GT process (when and how to apply IRR/IRA during the iterative process of GT) and the multiple coding procedures that GT involves, so researchers have been using ad hoc methods for years, which makes it difficult to use IRR/IRA systematically and extensively.

This paper presents a process for systematically measuring IRR/IRA in GT studies that meets the iterative nature of this qualitative research method and its different coding procedures. This helps researchers to develop a shared understanding of the phenomenon being studied by establishing either consistency or agreement among coders, trustworthiness, and robustness of a code frame to co-build a theory through collaborative team science and consortia, and, thus, rigor in qualitative research. To this aim, this paper (i) examines the use of IRR/IRA techniques in recent GT studies in software engineering to identify the main challenges and gaps through a systematic mapping study, (ii) formalizes a process for systematically measuring IRR/IRA in GT, and (iii) shows its feasibility in a GT study.

The structure of the paper is as follows. Section 2 provides an overview of GT, the factors that could lead to collaborative coding, the criteria for rigorous qualitative research, and the role of IRR/IRA in qualitative research. Section 3 reports a mapping study on the use of IRR/IRA techniques in recent GT studies in software engineering. Section 4 describes a process for systematically measuring IRR/IRA in GT and Section 5 shows the feasibility of this process through its application to a GT study. Section 6 assesses the validity and reliability of these outcomes. Section 7 describes the related work. Finally, conclusions and further work are presented in Section 8.

2. Background

2.1. Grounded theory

GT has been defined in its most general form as "the discovery of theory from data" ((Glaser and Strauss, 1967), p. 1). GT constitutes a set of different families of research methods, originally rooted in social sciences but with applications in different domains (psychology, nursing, medicine, education, computer science, managerial and accounting sciences, and even urban planning). Since the publication of its seminal work (Glaser and Strauss, 1967), GT has branched into different families. The most recognizable families within GT are Classic or Glaserian (Glaser and Strauss, 1967), Straussian (Strauss and Corbin, 1990), and Constructivist or Charmaz GT (Charmaz, 2014). They retain a common core of methods, vocabulary, and guidelines, such as coding, constant comparison methods, memo writing, sorting, theoretical sampling, and theoretical saturation, with the final aim of discovering or developing a substantive theory grounded in data. They present different nuances in coding procedures that have been referred to as open and initial coding; focused, axial, selective coding; and theoretical coding (Saldaña, 2012; Kenny and Fourie, 2015). We can point out the epistemological underpinnings of GT and the concept of theory sensitivity (namely, the role of literature and academic background knowledge in the process of developing the theory) as the causes of the divergence of schools (Kenny and Fourie, 2015; Stol et al., 2016). Hence, the controversial and distinguishing issues of GT can be traced back to the following issues:

- Epistemological position: ranging from (naïve) positivism to constructionism. Although the foundational work of GT (Glaser and Strauss, 1967) does not adhere to any epistemology, it is acknowledged for its underlying positivist position along with a realist ontology in the classical approach. The Straussian variant modifies this view in favor of a post-positivist position, embracing symbolic interactionism; whereas Charmaz explicitly assumes a constructivist epistemology and a relativist ontology (Kenny and Fourie, 2015).
- Theoretical sensitivity is a complex term that in GT denotes both the researcher's expertise in the research area and his/her ability to discriminate relevant data. The role of literature review is also relevant to grasp this slippery concept of theoretical sensitivity. In Classic GT, the researcher is asked not to be influenced by the existing literature in the construction of the new emerging theory, while being aware of it. Furthermore, research should approach the data without a clear research question, which should emerge from the data. The Straussian paradigm allows for a much more flexible role of literature review when posing the research question and during the research process, since it will

enhance theoretical sensitivity. The Charmazian tradition postulates a much more prominent use of literature to be done at the beginning of the research (Kenny and Fourie, 2015).

These inconsistencies have generated a lot of criticism about GT. Charmaz assumes that the researcher cannot evade from this debate, "*epistemological stances are, however, significant because they shape how researchers gather their data and whether they acknowledge their influence on these data and the subsequent analysis*" (Charmaz and Thornberg, 2020); and because it shapes the source of the validity of the obtained knowledge. Thus, the formalization of a new process for GT should be compatible with GT variants mentioned above and flexible enough to different philosophical positions (epistemology and ontology) and theoretical sensitivity.

2.2. Factors leading collaborative coding

In the Auerbachand & Silverstein's words "*All research should be conducted in groups rather than in isolation, particularly when doing qualitative research.*" (Auerbach and Silverstein, 2003).

GT studies may involve multiple researchers in collaborative coding. According to the Empirical Standards for Software Engineering Research (Ralph, 2021), in which some authors of this paper were involved, some factors to consider team coding are as follows:

- Controversiality. The more potentially controversial the judgment, the more multiple raters are needed; e.g., recording the publication year of the primary studies in an SMS is less controversial (i.e., coding requires little interpretation) than evaluating the elegance of a technical solution.
- Practicality. The less practical it is to have multiple raters, the more reasonable a single-rater design becomes; e.g. multiple raters applying an a priori deductive coding scheme to some artifacts may be more practical than multiple raters inductively coding 2000 pages of interview transcripts, although inductive research could also benefit from team coding.
- Philosophy. Involving multiple raters is more important from a realist ontological perspective (characteristic of positivism and falsificationism) than from an idealist ontological perspective (characteristic of interpretivism and constructivism), although idealist ontology perspective could also benefit from team coding.

Due to these factors and the fact that GT studies are becoming larger and more complex, there is a trend toward collaborative coding (Erickson and Stull, 1998; Weston et al., 2001; Guest and MacQueen, 2008), so formalizing a process for GT studies involving multiple raters can make collaborative science and consortia increasingly systematic and broad.

2.3. Criteria for rigorous qualitative research

The appropriate criteria for assessing qualitative research are controversial and often debated not only in the literature (Lincoln and Guba, 1985; Gibbs, 2007; Creswell and Creswell, 2017), but also during peer review and dissertation defense. Epistemological and ontological diversity, as well as differences in research traditions between fields, hinders establishing a broad consensus on these criteria.

Many qualitative researchers claim for *qualitative validity*, which means that the researcher assesses the accuracy of the findings by employing certain procedures, and *qualitative reliability*, which indicates that the researcher's approach is consistent across different researchers and among different projects (Gibbs, 2007; Creswell and Creswell, 2017). In contrast, other qualitative researchers reject validity and reliability altogether in favor of qualitative criteria such as *credibility*, *transferability*, *dependability* (parallel to the conventional criterion of reliability) and *confirmability* (Lincoln and Guba, 1985). Therefore, whether to establish reliability in qualitative research and what reliability means for interpretivists building a theory depends on researchers' traditions in different (sub-)disciplines (Armstrong et al., 1997; Campbell et al., 2013), from health sciences, psychology, sociology, and business, which may expect formal measures of reliability, to education, information management, and software engineering, which rarely rely on these measures but with an increasing interest in the last years (Wohlin et al., 2012; Nili et al., 2017; McDonald et al., 2019). In this regard, it is necessary to analyze the role of IRR/IRA as a criterion in qualitative research.

2.4. IRR/IRA in qualitative research: general guidelines

McDonald et al. (2019) and O'Connor and Joffe (2020) described norms and guidelines for IRR/IRA in qualitative research in the computer and social sciences, respectively. McDonald et al. (2019) examined 251 papers in computer science (specifically, in computer-supported cooperative work and human–computer interaction) and found that most papers described a method of IRR or IRA in which two or more raters were involved, and most of the papers used a process that the authors described as inductive. O'Connor and Joffe (2020) conducted an in-depth analysis about arguments in favor of and objections to IRR/IRA in research based on inductive analysis and interpretativist or constructivist epistemology (see Table 1). They concluded in words of Braun and Clarke (2013), IRR/IRA "*no necessary imply there is a single true meaning inherent in the data which is the concern underpinning most epistemological objections*", "*Rather, it shows that a group of researchers working within a common conceptual framework can reach a consensual interpretation of the data*".

Thus, increasing researchers in different disciplines go beyond IRR/IRA as a statistic or measurement of objectivity and approach IRR/IRA as a tool for improving researcher reflexivity and quality criteria in qualitative research, either for inductive or deductive analysis. One example are Wu S. et al. (2016) who examined various author guidelines for manuscripts reporting qualitative research from a set of journals that recommends the use of IRR/IRA. Later, in 2021 the Empirical Standards for Software Engineering Research (Ralph, 2021), in which some authors of this paper were involved, describes some essential attributes that a study should address when applying IRR/IRA. According to this standard, the study should:

- clearly state what properties were rated,
- clearly state how many raters rated each property,
- describe the process by which two or more raters independently rated properties of research objects,
- describe how disagreements were resolved,
- indicate the variable type (nominal, ordinal, interval, ratio) of the ratings, and
- report an appropriate statistical measure of IRR/IRA.[1]

Therefore, the formalization of a new process for GT should consider these attributes.

[1] IRR is a correlation measure that can be calculated using Cronbach's α, Pearson's r, Kendall's τ, and Spearman's ρ, among others. IRA is a measure of agreement that can be calculated using Scott's π, Cohen's κ, and Krippendorff's α, among others.

Table 1
Arguments in favor and against of IRR/IRA in qualitative research.
Source: Adapted from O'Connor and Joffe (2020)

Arguments in favor
1. Assess rigor and transparency of the coding frame (refinement)
2. Improve communicability and confidence (beyond an individual interpretation of a researcher)
3. Provide robustness (convergence on the same interpretation of the data)
4. Show that analysis is performed conscientiously and consistently
5. Foster reflexivity and dialogue
Arguments against
1. Contradicts the interpretative epistemological stance
2. Reliability is not an appropriate criterion for judging qualitative work
3. Represents a single, objective, external reality instead of the diversity of interpretations

3. Systematic mapping study (SMS): A secondary study on GT and IRR/IRA

This mapping study was carried out by a team of four researchers (referred to as R1, R2, R3, and R4), who coauthored this article. The objective of this secondary study is to verify whether the use of IRR/IRA techniques in GT studies is a common practice. Studies in the field of software engineering have been analyzed from 2016 to 2021. Next, the section is structured following the guidelines by Kitchenham and Charters (2007), Pérez et al. (2020), and the good practices described in the Empirical Standards for Software Engineering Research (Ralph, 2021).

3.1. Planning the SMS

Planning the SMS consists of developing a review protocol that specifies (i) the review objective and research questions; (ii) the search strategy; (iii) the inclusion/exclusion criteria; (iv) the data extraction strategy; and (v) the strategy to synthesize the extracted data. All these steps are described in the following subsections.

3.1.1. Review objective & research questions

This secondary study aims to review the state-of-the-art in the use of IRR/IRA techniques in recent GT studies in software engineering. The following research questions (RQ) lead this review:

- **RQ1** To what extent have IRR/IRA techniques been used in GT studies carried out in the field of software engineering?
- **RQ2** How has IRR/IRA been instrumented in previous GT studies in software engineering?

3.1.2. Search process

A formal search strategy is required to find the entire population of scientific papers that may be relevant to answer the research questions. The formal definition of this search strategy allows us to carry out a replicable review open to external evaluations. The search strategy consists of defining the search space: electronic databases and journals and conference proceedings that are considered key spaces for the review objective. For this work, the search was carried out in the following electronic databases: ScienceDirect, Springer Link, IEEE Xplore, Scopus, and ACM DL. We filtered by year (from 2016 to 2021) and subject area (that is, computing science and, specifically, software engineering). The general search string used for this search is the following.

```
("grounded theory") AND ("inter-rater agreement" OR
↪  "inter-rater reliability" OR "inter-judge
↪  agreement" OR "inter-judge reliability" OR
↪  "inter-coder agreement" OR "inter-coder
↪  reliability")
```

3.1.3. Study selection process

The selection process specifies inclusion criteria (IC) and exclusion criteria (EC) to determine whether each potential study should be considered or not for this systematic study (see Table 2). Specifically, the study selection process we followed was described in a previous work that aimed to reduce bias and time spent in the study selection process (Pérez et al., 2020).

This process was carried out by researchers R2 and R3 (2nd and 3rd authors). They analyzed a set of retrieved studies to determine whether the study is included or excluded, which is reported in Table 3. Both researchers met to compare their results, refine IC/EC (if applicable), and calculate IRA. To calculate IRA, we used Krippendorff's α (binary) (Krippendorff et al., 2016; Krippendorff, 2018) as described in González-Prieto et al. (2020). When $\alpha \geq 0.8$, the dual selection process is stopped and each researcher independently processes half of the remaining primary studies. However, to ensure that the agreement remains in force (the IC/EC are still interpreted in the same way) and as a quality control measure, some control points were carried out. Using these control points, both researchers reviewed a new set of studies and recalculated their agreement.

3.1.4. Data extraction process

This phase aims to recover the information necessary to answer the research questions. According to the protocol we defined, researchers R1 and R4 (1st and 4th authors) performed data extraction independently and without duplicity (duality is not necessary as the data to be extracted are totally objective). For each primary study, we extracted:

- Epistemology/Ontology: ranging from positivism or realism/objectivism to constructivism/interpretivism or relativism.
- GT variant: classic or glausserian, straussian, constructivist/charmazian. If a paper claims to apply a GT approach (mainly coding) but its application is questionable (for instance, the study does not apply theoretical sampling, the constant comparison method, memoing, saturation, or no emerging theories are shown), the study is labeled as "GT-like approach".
- Data gathering method: semistructured or structured interviews, surveys, etc.
- GT coding phases and methods: initial coding, open coding, axial coding, focused coding, selective coding, theoretical coding, constant comparative method, etc.
- Did the paper claim to apply Inter-Rater Agreement (IRA) techniques?
- Did the paper claim to apply Inter-Rater Reliability (IRR) techniques?
- Are Reliability & Agreement terms correctly used?
- IRA Instrument: Scott's π, Cohen's κ, Fleiss's κ, and Krippendorff's α, among others.

Table 2
Selection criteria.

Inclusion criteria	Exclusion criteria
1. GT should be used as a research methodology in the study 2. IRR/IRA is measured	1. The GT methodology is mentioned, but not used in the study 2. IRR/IRA is mentioned but not measured 3. IRR/IRA is not used for the development of the theory during the GT coding process (often used in a previous literature review) 4. IRR/IRA is measured, but the statistical measure is not specified

Table 3
Study selection form template.

Inclusion criteria (of the current iteration)		Exclusion criteria (of the current iteration)	
Reviewer:			
Study ID	Study title	Include? (Y/N)	IC/EC
...	...	...	...

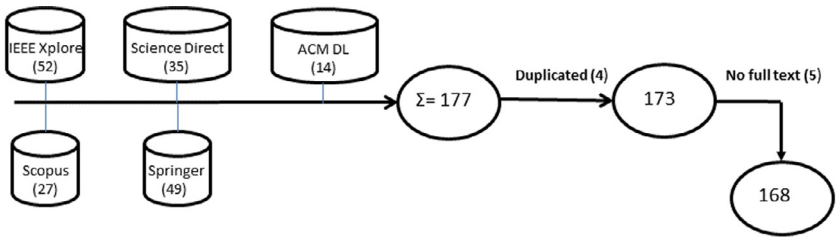


Fig. 1. Results of the Search Process.

- IRR Instrument: Pearson's r, Kendall's τ, and Spearman's ρ, among others.
- Process: Brief description of the IRR/IRA process that the authors applied in their GT study.

These items are used to define a coding scheme that is processed using a computer-assisted qualitative data analysis (CAQDAS) tool named Atlas.ti v9.

3.1.5. Synthesis process

After data extraction, R1 and R3 performed a synthesis process to summarize the main ideas and discoveries from the data. In other words, the synthesis process consists of organizing key concepts to enable high-order interpretation.

3.2. Conducting and reporting the SMS

This section reports the results of the search, selection, extraction and synthesis of the study. The mapping study retrieved 173 unduplicated scientific papers. The references of these papers and the results of the process are available in a public repository (see Data availability), that is, a replication package to motivate others to provide similar evidence by replicating this secondary study.

3.2.1. Results of the search process

Following the review protocol described in Section 3.1.2, a search was carried out for primary studies. We located 177 studies from the databases that we defined in the protocol, of which 4 were duplicates. Additionally, it was not possible to obtain the full text of 5 studies. Therefore, 168 studies are input for the mapping study (see Fig. 1). The search strings used in each electronic database and the 168 studies are listed in the "replication_package.xlsx" file of the repository.

3.2.2. Results of the selection process

Of a total of 168 retrieved studies, R2 and R3 dually analyzed 14 + 8 + 8 using IC/CE (see the green line in Fig. 2 that corresponds to an initial set of studies and two control points). They

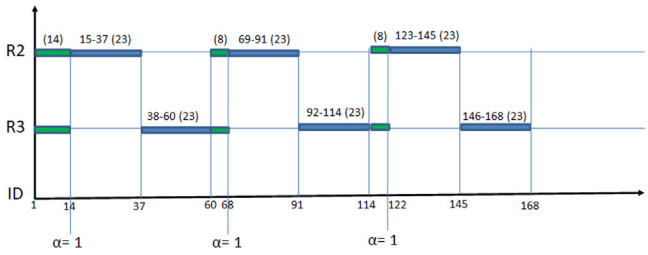


Fig. 2. Results of inter-rater agreement.

obtained no observed disagreement ($D_o = 0$), that is, perfect agreement. Therefore, the value of the Krippendorff's coefficient $\alpha = 1 - D_o/D_e = 1$, where D_e is the expected disagreement. This value indicates that there exists a very high level of reliability in the selection process. Each researcher also reviewed 69 studies individually (23 + 23 + 23) (see the blue line in Fig. 2), then a total of 138 studies were reviewed individually. This process is described in the "replication_package.xlsx" file in the repository. At the end of this process, we selected 49 primary studies, which describe a GT study and use statistical techniques, IRR or IRA, to analyze consensus in collaborative coding.

3.2.3. Results of the data extraction process

R1 and R4 performed data extraction on the 49 primary studies using a coding schema that was implemented using Atlas.ti v9. The results of the data extraction process are shown in Table 4. For the sake of readability, the description of the process by which the authors of the primary studies conducted IRR/IRA in their GT studies is not included, but, for further details, it can be checked in the "replication_package.xlsx" file of the repository.

3.2.4. Results of the data synthesis

R1 and R3 performed the data synthesis. From the 49 primary studies, we concluded as follows. Most of the papers did not mention epistemology or ontology, except for two that explicitly

mentioned constructivism (see column 2 in Table 4). A few more than half of the primary studies (28 papers) conducted a GT study as defined in Stol et al. (2016) and conform to a set of essential (or desirable) attributes as defined by the Empirical Standards for Software Engineering Research (Ralph, 2021). Of these 28 papers, 16 selected the Straussian variant, 3 papers the Charmaz variant, and 1 paper the Classic (or Glaserian) variant, whereas the rest of the papers did not mention any particular GT variant (see column 3 in Table 4). The other 21 papers conducted a GT-like approach, i.e., the authors only superficially mentioned the GT method to justify the use of coding procedures, without referring to iterative and interleaved rounds of qualitative data collection and analysis to lead to core categories and key patterns, and generate a theory, and without referring to specific GT coding procedures such as initial, open, focused, axial, selective, and theoretical coding (see column 5 in Table 4).

Of the 49 primary studies, 33 say to apply IRR, while 16 IRA (see columns 6 and 7 in Table 4). However, according to the formal definitions given by Gisev et al. (2013), many authors indicated to measure IRR but instead measured IRA, that is, the authors indicated to measure the extent to which raters consistently distinguished between different items on a measurement scale but instead measured the extent to which different raters assigned the same precise value for each item being rated (see column 8 in Table 4). To this end, most of them used Cohen (1960) and Krippendorff (2018) (see column 9 in Table 4). Therefore, it seems clear that IRR and IRA are often interchangeable terms.

In addition to this descriptive analysis, the analytical reasoning of the IRA/IRR application in these GT studies shows some shortcomings.

- None of the primary studies provide a reasonable justification for collaborative coding in terms of controversiality, practicality, or philosophy (see Section 2.2).
- However, all of them justify collaborative coding and the use of IRR/IRA as criteria for rigorous qualitative research (see Section 2.3), that is, as a means of avoiding researcher bias (e.g., ID7, ID55); gaining in reliability and consensus (e.g., ID67, ID110, ID151) and robustness (e.g., ID17); testing validity (e.g., ID67, ID110, ID132, ID144); and for refining codebooks (e.g., ID8, ID34) and clarifying definitions (e.g., ID12).
- Almost no primary studies describe the process by which two or more raters independently rated codes in sufficient detail, for example, who defines the code frame, what the code frame is, the units of coding (that is, how data are segmented into meaningful quotes, for example, a paragraph, a line, etc.), how disagreements were resolved, etc. Only two primary studies (ID8 and ID55) mentioned an iterative process, which is described in little more than a paragraph.
- Only eight primary studies explicitly indicated the corpus (data) over which IRR/IRA is applied.
- Only five primary studies explicitly indicate the minimum threshold that indicates acceptable reliability/agreement.
- Most papers report an inappropriate statistical measure of IRR/IRA; hence, 33 primary studies stated using Cohen's kappa and Krippendorff's α to measure IRR when these statistical techniques measure IRA.

Next, we show some excerpts from the primary studies to make the chain of evidence explicit. Hence, ID144 and ID161 pointed out at the use of quantitative techniques to *enhance the quality* of qualitative data:

> *ID144 "To enhance the rigor of the quantitative analysis of qualitative data analysis, a triangulation of analysts (Patton, 1999) was employed. Two researchers coded four interviews with the nine care transition outcomes separately; they met and reviewed their coding to discuss differences and refine outcome definitions. The two researchers then coded two more interviews separately to evaluate inter-rater reliability to strengthen the internal validity of the research".*

ID01, ID11, ID26, ID37, ID58, ID62, ID66, ID74, ID83, ID88, and ID163 superficially described the IRR/IRA process by indicating only the number of coders, and vaguely how disagreements were discussed.

> *ID01 "Coding was performed independently by two coders who met frequently to discuss codes in order to ensure high inter-rater reliability".*

> *ID11 "Each tweet was coded independently by two coders. Kappa coefficients measuring inter-coder reliability above chance agreement ranged from fair to good (50% to 88%)".*

ID144 and ID153 acknowledged a minimum threshold to achieve acceptable agreement when using Cohen's κ and ID88 described an acceptance threshold for Krippendorff's $\alpha > 0.8$. However, only ID07 and ID57 seem to describe a minimum threshold of Krippendorff's α as a tool for improving the consistency of a code frame, although it is not explicitly described.

> *ID144 "Cohen's kappa was calculated for all outcomes; all values were above the acceptable value of 0.8 and indicate that the interpretation and coding of interview data are reliable".*

> *ID153 "A further independent inter-rater test was performed which achieved a 75% agreement which according to Landis and Koch is a "substantial agreement".*

> *ID07 "The results of the Krippendorff's α test suggest that there was a 69% agreement between the observers. Because this result was below the commonly accepted threshold of an α of 80%, the first two authors deliberated over the differences to form one consistent initial coding set".*

ID05, ID13, ID17, ID24, ID28, ID89, ID92, and ID128 described the corpus over which IRA is calculated (25%, 20%, approx. 30%, 10%, approx. 10%, 26%, 10% and 20% respectively). Specifically, ID17, although it uses IRR to refer to IRA, indicates the role of the researchers and the percentage of data over which IRA is calculated.

> *ID17 "The bulk of the coding was performed by the first author. In order to ensure the robustness of the coding system, the remaining three authors performed two independent coding passes of a subset of 50 of the 230 artifacts in the first pass, and 25 of the 230 artifacts and 6 of the 60 videos, at two stages in the development of the code books. We calculated the inter-coder reliability ratio as the number of agreements divided by the total number of codes [...]"*

ID128 "To test the inter-coder reliability, the primary researcher coded all 154 records, and subsequently the second coder coded every fifth record in the dataset. Cohen's Kappa coefficient was found to be 0.84, indicating high agreement between the two coders".

ID55, ID59, ID65, and ID89 are the only ones of the few to lightly describe the expertise of the raters involved in the coding process, but none of them mention specific training in the coding process.

ID55 "To avoid the researcher's bias, we have performed an inter-rater reliability test between mapping team and indented experts [...]"

ID59 "The initial team of paper taggers was made up of seven post-docs and graduate students with some association to the University of Trento and some experience with goal modeling".

ID02 "A total of 433 excerpts were extracted from the interview transcripts (excluding answers from Q4 which was quantitative), and 113 excerpts (26%) were randomly selected and double coded by two independent coders who were social science graduate students".

ID110 relies on automated text coding. Data are codified by an algorithm and subsequently validated by human experts.

ID110 "To code the data, we developed an automated coding scheme using the nCodeR package for the statistical programming language R [...]. We used nCodeR to develop automated classifiers for each of the codes in Table 1 using regular expression lists [...]. To create valid and reliable codes, we assessed concept validity by requiring that two human raters achieve acceptable measures of kappa and rho, and reliability by requiring that both human raters independently achieve acceptable measures of kappa and rho compared to the automated classifier".

ID 67 is the one that describes the IRR/IRA process in relation to the different GT coding procedures (i.e., open, axial, and selective coding phases). However, most of the papers use IRR/IRA as a finalist measure like ID87. Hence, ID97 and ID153 explicitly indicate that IRA is calculated after the theoretical saturation was reached.

ID67 "Hence, in order to implement IRR, two coders were involved in independent analysis and coding the transcripts from the interviews and the convergence of their findings was evaluated at the end of each open, axial, and selective coding phases. In cases of conflicts between the decisions made by these two coders, a third coder was involved in the discussions for resolving the conflicts. At the end of each coding phase, we merged the coding files from ATLAS.ti and exported the coding results of each researcher into Microsoft Excel. We used Microsoft Excel to calculate Kappa as a measure of IRR".

ID87 "After developing the coding scheme through grounded theory as described above, we conducted a second phase of analysis to test inter-rater reliability".

ID153 "Selective coding is the final coding process in GTM, and involves the selection of core categories of the data. Selective coding systematically relates the categories identified in axial coding, and integrates and refines them to derive theoretical concepts. After theoretical saturation, we conducted an inter-rater reliability test evaluation using Cohen's kappa".

Only ID08 and ID34 explicitly mentioned an iterative process that aims to improve a code frame (e.g. removing ambiguous codes) and, thus, improve researchers' reflexivity.

ID08 "Four coders independently coded two samples to refine the coding scheme. We then discussed and used affinity diagramming to synthesize emerging themes. Next, we went through several iterations to check another two samples individually. The purpose of this step was to confirm the legitimacy of the coding scheme and to check the inter-rater reliability. After several iterations, four coders reached a suitable level of agreement (Fleiss's kappa, $\kappa = 0.71$)".

ID34 "Initially, the 1st and 2nd author each independently coded a new sample of five analyses (20% of the data), receiving a low Cohen's Kappa of 0.55 (Pérez et al., 2020). Both authors discussed disagreements, refined the code book, and repeated the process on a new sample of five interviews. With a moderate Cohen's Kappa of 0.71 (Pérez et al., 2020), the two authors labeled all remaining interviews together, allowing for multiple labels where needed, as decided through discussion and consensus. Afterwards, our analysis followed 'data-driven' thematic analysis (Cruzes and Dyba, 2011) where we clustered our coded data into themes".

4. A process for IRR/IRA in grounded theory studies

This section presents a process for systematically applying IRR/IRA in GT studies in which multiple researchers are involved in collaborative coding. Before describing this process, it is necessary to highlight two concerns that the process should consider. The first one is about coding. As McDonald et al. (2019) examined in previous literature, coding is sometimes used to describe a process of *inductive interpretation*, and other times is used to describe a process of *deductive labeling* of data with preexisting codes, even sometimes both approaches are integrated as Cruzes and Dyba recommended for thematic analysis in software engineering (Cruzes and Dyba, 2011).

The second one is about the purpose of measuring IRR/IRA. When multiple raters collaboratively code, consensus could be reached through "intensive group discussion, dialogical intersubjectivity, coder adjudication, and simple group consensus as an agreement goal" (Saldaña, 2012). However, *you cannot improve what you cannot measure*, and precisely, the IRR and IRA techniques allow researchers to measure consistency and agreement between multiple coders. Measuring consistency and agreement among raters, where appropriate, promotes "systematicity, communicability, and transparency of the coding process; reflexivity and dialogue within research teams; and helps to satisfy diverse audiences of the trustworthiness of the research" (O'Connor and Joffe, 2020). It is particularly crucial to identify mistakes before the codes are used in developing and testing a theory or model, that is, to ensure robustness before analyzing and aggregating the coding data. Weak confidence in the data only leads to uncertainty in the subsequent analysis and generates doubts on findings and conclusions. In Krippendorff's own words: "If the results of reliability testing are compelling, researchers

may proceed with the analysis of their data. If not, doubts prevail as to what these data mean, and their analysis is hard to justify" (Krippendorff, 2018).

Thus, IRR and IRA provide a key tool for achieving inter-coder consistency and agreement by encouraging consensus and reflexivity (Hammer and Berland, 2014) and a shared understanding of the data, discovering where coders disagree, and revealing weaknesses in coding definitions (McDonald et al., 2019), overlaps in meaning (MacPhail et al., 2016), or difficulties in consensus given the nature of data (Hammer and Berland, 2014). For us, the process of reaching consensus, either consistency or agreement (or both), is more important than its measurement, although measurement is the key to conducting this process.

4.1. The process

The process for GT studies that we propose (see Fig. 3) starts with an initial research question(s) and data collection, using purposive, convenience, or theoretical sampling strategies. In this process, $N > 1$ and $M > 1$ raters are involved, in such a way that the values of N and M can be the same or different and refer to the same or different coders if the statistic used to measure the IRR/IRA allows for it. The number of coders depends on the factors described in Section 2.2. Note that the greater the number of raters (coders), the more difficult it is to reach a consensus (either consistency IRR or agreement IRA), but trustworthiness is improved. The raters are then involved in the coding of a subset of data (e.g., interviews, qualitative survey responses, or any other data subject to qualitative analysis). To make our process flexible to the existing GT variants, it is based on the common stages of these variants according to Stol et al. (2016) and Kenny & Fourie (see Kenny and Fourie (2015)), who analyzed and compared the three main GT variants. We aim to have a fluid process with at least three stages to coding, i.e., open/initial coding, selective coding, and sorting theoretical coding, although axial coding, focused coding, etc. could also be added. These stages are described as follows:

Initial/open coding: This activity involves multiple rounds of coding, constant comparison, and memoing (see Fig. 3), specifically one round per rater involved in collaborative coding. The first rater analyzes the subset of selected data (e.g., between 5–10 instances[2]) by reviewing the data line by line, creating quotations (highlighted segments of text), assigning new codes to the quotations, and writing memos, that is, notes about ideas or concepts potentially relevant to the research. As more data instances are analyzed, the resulting codes (code frame or codebook) are refined by using the constant comparison method, which forces the rater to go back and forth. The following raters analyze the same subset of selected data in which the quotations that the previous raters created are visible (although raters never see the coding of previous raters, i.e., the codes assigned to each quotation). Therefore, the following raters only see the quotations of previous coders (without codes), the current version of the codebook and memos. The following raters can create new quotations (for relevant data that were omitted by previous raters), assign new codes or previously defined codes to the quotations, split one code into two different codes, merge codes, adjust the definition of a code, and write new memos or modify the existing ones, if necessary, while constantly comparing codes with each other, within the data instance and between instances. The next coder will see these modifications in the codebook and in a ***disagreement diary*** and will perform the coding under these new circumstances. Thus, this activity is an integrated approach of inductive and deductive coding.

Once all raters have coded the subset of selected data, collaborative coding is the input for measuring IRR and/or IRA (see activity **Calculate (1) IRR/IRA** in Fig. 3).

- If this measure is less than a minimum threshold (which could vary depending on the statistical technique), a group discussion is followed to collaboratively reach consensus and/or agreement (see activity **group discussion, dialogical inter-subjectivity, coder adjudication, and simple group consensus** in Fig. 3). This activity aims to identify coding disagreements, weaknesses in coding definitions, overlaps in meaning, etc. as mentioned above, which are documented in a ***disagreements diary***. After possible modifications to quotations, codes, and memos, a new iteration of initial/open coding starts over a new subset of selected data (if necessary, new data are collected).
- If this measure is higher than or equal to the minimum threshold, the following activity starts.

Selection of core categories aka. variables (see Fig. 3): raters select core categories from the most relevant and important codes obtained in the previous coding procedure (the usual criteria for selecting core categories can be *groundedness* and *density*, i.e., the number of quotations linked to a code and the number of other codes connected to a code, respectively).

Selective coding: This activity also involves multiple rounds of coding, constant comparison, and memoing (see Fig. 3), specifically one round per rater involved in collaborative coding. All raters analyze the same new subset of selected data by reviewing the data line by line, creating quotations (segments of text), assigning them a core category (i.e., subcodes of a core category), writing memos, and comparing the categories with one another. This coding procedure is an integrated approach of inductive and deductive coding that focuses only on the core categories and subcodes of these categories. Thus, coding is a deductive process of labeling data with preselected core categories and an inductive process of creating and labeling data with possible new subcodes of these preselected core categories. Again, successive raters analyze the same subset of selected data in which the quotations that the previous raters created are visible (however, raters never see the coding of previous raters, i.e., the codes assigned to each quotation).

Once all raters have coded the selected data subset, collaborative coding is the input to measure IRR and/or IRA (see activity **Calculate (2) IRR/IRA** in Fig. 3).

- If this measure is less than a minimum threshold (which could vary depending on the statistical technique), a group discussion is followed to collaboratively reach consensus and/or agreement (see activity **group discussion, dialogical inter-subjectivity, coder adjudication, and simple group consensus** in Fig. 3). This process aims to identify coding disagreements, weaknesses in category definitions, overlaps in meaning, etc. as mentioned above, which are documented in a ***disagreements diary***. After possible modifications to quotations, core categories, and memos, a new iteration of selective coding starts over another subset of selected data (if necessary, new data are collected).
- If this measure is higher than or equal to the minimum threshold, the researchers evaluate whether theoretical saturation is reached. If not, new data are collected via theoretical sampling and a new iteration of selective coding begins.
- If theoretical saturation is reached, researchers can continue towards theoretical coding (see Fig. 3).

[2] This is an arbitrary number selected by the researcher, which depends on the quantity and quality of the data collected and the availability of human resources.

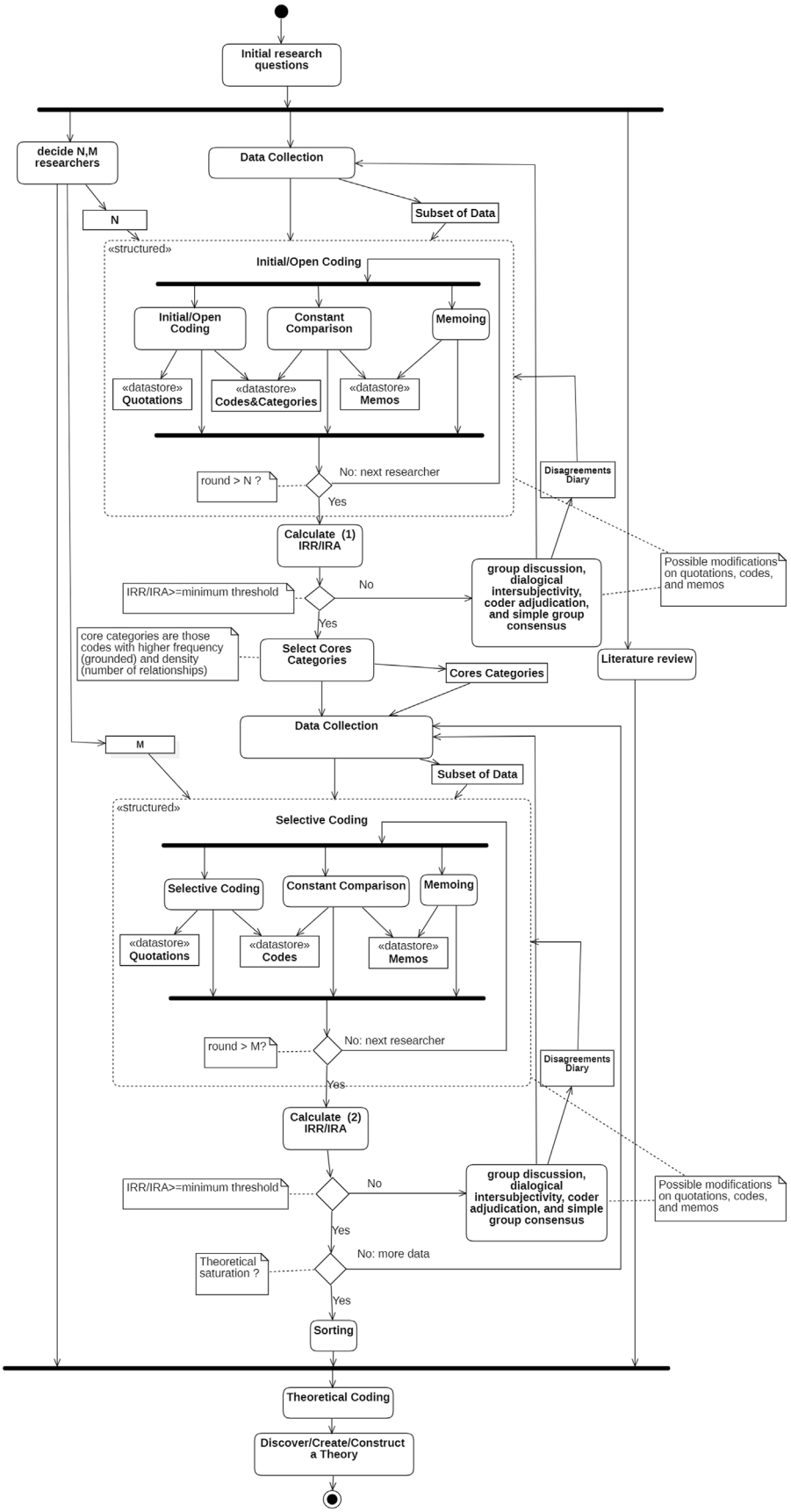


Fig. 3. Process for using IRR/IRA in GT studies (UML Activity Diagram).

When the stop conditions are met, the researchers tackle the subsequent the stages for **sorting core categories and memos** and **theoretical coding** for the sake of **discovering/creating/constructing a theory** (see Fig. 3).

Finally, the **role of the literature** has been formalized to be compatible with the three variants of GT, reviewing the literature to fit the purpose of the GT study (Charmaz, 2014) or delaying the review of the literature until the theory has emerged to validate it (Glaser and Strauss, 1967).

The process described here meets the iterative nature of the GT research method and helps in developing a shared understanding of the phenomenon being studied by establishing either consistency or agreement among coders, and thus, the trustworthiness and robustness of a code frame to co-build a theory through collaborative team science and consortia.

4.2. Discussion

Next, some aspects related to the process are described and discussed.

1. The process does not state anything about how to code. It does state the phases, activities, and milestones of a GT process that incorporates IRR/IRA. Fig. 3 indicates where and when to use these statistics, but says nothing about how to do open coding, axial, or theoretical coding.

2. The process does not impose any restrictions about how many researchers collect data or who collect these data, that is, there can be multiple data collectors or only one, and they can be the coders themselves or different researchers. This flexibility is one of the benefits of the proposed process. Moreover, the number and identity of the coders are allowed to be dynamically changed if the statistic used to measure the IRR/IRA allows for it (e.g., Krippendorff coefficients do). Therefore, the values of N and M (Fig. 3) can be the same or different and refer to the same or different coders.

3. According to the process described here, the coders work sequentially, i.e., one coder does not start his/her coding process until the previous one has finished, and he/she uses the quotations and the code frame (or codebook) generated so far (although the coding of previous coders is not visible). However, perhaps it is worth considering that parallel coding would save time. Thus, why not code in parallel? If several coders work simultaneously, each coder could define a different set of codes from a morphological, lexical, syntactic, or semantic point of view. Only the latter case is relevant when building a theory since the disagreement would be about the constructs themselves (their meaning). The other sources of disagreement only imply a loss of time in meetings to work out the disagreements. Hence, a morphological (ball versus balls) or lexical (kids versus children) disagreement does not involve a disagreement on the meaning of the construct but only on the way of referring to it. The same applies to syntactic disagreements; that is, the meaning of a code can be expressed with a phrase that admits a different order of its constituent parts or with semantically equivalent phrases. To avoid having to resolve these kinds of "format disagreements", we propose that coders work sequentially using the codes and quotations defined by previous coders. This process does not prevent the generation of "format disagreements", but it does avoid an initial explosion of codes (and disagreements) that must be agreed upon.

4. GT methodology prescribes an iterative process that ends when saturation is reached. When a single researcher is involved in the process, this condition is necessary and sufficient. However, when there are several coders, saturation is necessary, but not sufficient. What happens if saturation is reached and there is no agreement among the coders? If we go on to build the theory with no agreement on the semantics of the constructs, we may lose the benefits we are pursuing with the use of IRR/IRA. The opposite case, in which the agreement is reached but not the saturation, involves new iterations until both are reached.

5. In some areas, such as social science, collaborative coding and IRR/IRA techniques are widely used (Wu S. et al., 2016). In computing science, the use of GT is still immature, even some authors discourage the use of quantitative techniques, such as IRR/IRA, in qualitative research, specifically GT studies. Despite this fact, the SMS (see Section 3) shows that there is also a trend in the use of these techniques in recent years. What is clear is that there is no process (according to our best knowledge and conscience) that formalizes how to do this, neither in computer science nor in other areas.

6. Our process pursues of *continuous improvement* of the codebook. However, not every time all resources/authors are available/willing to do the tedious and time-consuming job of coding the data. To address this issue, it is possible for a single researcher to continue with the coding of the data after an IRR/IRA higher than or equal to the minimum threshold, but one must be aware that, from that time on, he/she is failing to incorporate the benefits described in this paper when collaborative coding and IRR/IRA are used in qualitative research.

5. Application of the process for IRR/IRA in a GT study

This section describes part of a GT study conducted by the authors in the domain of *Edge Computing* and *DevOps* in industry (*EdgeOps*) over the last while, aimed at illustrating the application of the proposed process for IRR/IRA in GT studies. This process has involved simultaneous data collection and analysis as described in Section 4.

According to *purposive sampling strategy*, we initially collected data from a set of participants from leading organizations in the Internet of Things domain, which are currently committee members of the Master's Degree in Distributed and Embedded Systems Software[3] and Master's Degree in IoT[4] at the Universidad Polit'ecnica de Madrid, Spain. Then we moved on to *theoretical sampling* and iteratively collected more data based on those concepts or categories that were relevant to the emerging theory until the value of inter-coder agreement (ICA) exceeded a given threshold and theoretical saturation was reached. A total of 27 responses were collected from an open-ended questionnaire available in https://es.surveymonkey.com/r/PMWD7ZM.

This GT study involved three researchers in the coding process (denoted by R1, R2, and R3) because of the controversiality of the terms around EdgeOps, whose definition, characterization, benefits, implications, and challenges have little consensus among the community due to its novelty. As multiple coders were involved, we applied ICA, and specifically Krippendorff coefficients (Krippendorff, 2018; González-Prieto et al., 2020), to improve the quality of our qualitative analysis – i.e., discover disagreements and reveal weaknesses in coding definitions, overlaps in meaning, etc. – and gain in researchers' reflexivity and a shared understanding of the data. Qualitative analysis was instrumented through Atlas.ti v9, which includes specific functionality to calculate Krippendorff coefficients.

The next subsections describe the main notions about Krippendorff coefficients we have used, and the multiple iterations that have been necessary during both initial/open coding and selective coding to exceed a certain threshold – that the community has approved – and during which codes and categories (and memos) were improved and clarified as disagreements revealed

[3] http://msde.etsisi.upm.es/

[4] https://masteriot.etsist.upm.es/?lang=en

weaknesses in coding definitions, lack of understanding, overlaps in meaning, among others. The results of the application of the process to this GT study on EdgeOps (including the different versions of the codebook and all statistical calculations of the Krippendorff coefficients) are available in a public repository (see Data availability).

5.1. Inter-coder agreement (ICA)

ICA is assessed to a raw matter of data (typically, transcripts of interviews, answers to surveys, video data, etc.), over which various coders highlight relevant parts (known as quotations) and label these quotations through a collection of codes (known as codebook) that represent different aspects of the reality that researchers want to understand. Additionally, codes are typically gathered into some meta-categories, called semantic domains. These semantic domains represent a facet of reality that researchers want to understand in a broad sense. Thus, it is typical to have some semantic domains $S_1, S_2, \ldots, S_n$ and each of these domains S_i contains several codes $C_{i1}, C_{i2}, \ldots, C_{in_i}$. This division cannot be arbitrarily made and must satisfy a property known as *mutual exclusiveness*. This means that the semantics of the different codes within a domain must be disjoint, or, in other words, it cannot be possible to assign to the same quotation two codes of the same semantic domain (C_{ij} and C_{ik} with $j \neq k$). To illustrate these concepts, we will use the EdgeOps study. Then we may have a semantic domain S_1 = conceptualization, with inner codes C_{11} = distributed architecture and C_{12} = limited device capability; as well as a semantic domain S_2 = functionality, with inner codes C_{21} = data collection and processing, C_{22} = video processing, C_{23} = artificial intelligence, and C_{24} = cloud shadowing. For each quotation, we can assign one code from S_1 and one code from S_2, but it is not possible to apply two codes of the same semantic domain to the same quotation (mutual exclusiveness). If necessary, the quotations should be split.

Therefore, at the end of the coding process, each of the coders has labeled a collection of quotations with one or more codes from the semantic domains according to the mutual exclusiveness rule. However, it is perfectly possible that the codings provided by the different coders do not agree; i.e., different subjects are interpreting the reality in different ways, maybe due to inconsistencies or fuzziness of the definition of the codes. To correct this issue, it is necessary to evolve the codebook by refining both codes and meanings until all coders interpret it in the same way and agree on its application. The detection of these flaws in agreement is precisely the aim of ICA techniques. These techniques are a collection of quantitative coefficients that allow us to measure the amount of disagreement in the different codings and to determine whether it is acceptable (so we can rely on the output of the coding process) or not (so we must refine the codebook and repeat the coding with new data).

For this purpose, in González-Prieto et al. (2020), a unified framework for measuring and evaluating ICA was established based on a new interpretation of Krippendorff's coefficients α. Krippendorff's α coefficients (Hayes and Krippendorff, 2007; Krippendorff, 2004, 2011; Krippendorff et al., 2016) are part of a standard tool used for quantifying the agreement in content and thematic analysis due to its well-established mathematical properties and probabilistic interpretation. In our research, we use the following Krippendorff's α coefficients, as described in Appendix.

5.2. Initial/open coding

Recall from Section 4 that this activity aims to discover the concepts underlying the data and to instantiate them in the form of codes. Thus, at each open coding iteration, n documents of the survey (a document is a set of responses to the survey of one of the participants) are analyzed by R1, R2, and R3, i.e., cut into quotations that are assigned to a previously discovered code or a new one that emerges to capture a new concept.

The process was carried out as follows. R1 analyzed the n documents, that is, identified quotations, created a codebook (codes and semantic domains), and handled the coding. When R1 ended, R2 analyzed the same n documents using the codebook created by R1, that is, he analyzed previous quotations, identified new ones, labeled these quotations with a code previously proposed by R1, and added new codes and memos, adjusted the definition of codes and memos, and merged/split codes. When R2 modified quotations and codes, these changes were reported in a *disagreements diary*. After R2 finished the coding process, the new codebook was delivered to R3 who repeated the process. Therefore, according to our process, the coders used the codes previously proposed by a researcher or generated new ones if they thought that some key information was missing. Hence, the process is flexible enough to allow coders to add their points of view in the form of new codes, but the existence of a common codebook also increases the chances of achieving a consensus.

After an iteration ends (that is, n documents have been coded by R1, R2, and R3), the ICA is calculated. In particular, we used Krippendorff's coefficient *Cu-α* as a quality control. *Cu-α* is a measure that considers consensus on the semantic domains irrespective of the codes within them. Two scenarios are possible:

- *Cu-α* is below an acceptable threshold (in this study, we fixed the standard *Cu-α* < 0.8). This evidences that there exist significant disagreements in the interpretation of the codes among the coders. In that situation, R1, R2, and R3 meet to discuss their interpretation of the codes. This *review meeting* delivers a *disagreements diary* and a *refined codebook* in which the definitions and the range of application of the codes are better delimited. With this new codebook as a basis, a new iteration starts with the next n' documents of the corpus.
- *Cu-α* is above or equal to the threshold (*Cu-α* ≥ 0.8). This means that there exists a consensus among the coders on the meaning of the codes. At this point, the open coding process stops and the generated codes (actually, the entire codebook) are used as input for the following activities, that is, the selection of the core categories (Section 5.3) and the selective coding (Section 5.4).

Additionally, the value of the Krippendorff coefficient *cu-α* is also calculated per semantic domain. As explained in Section 5.1, a low value of *cu-α* in a particular domain means that the coders do not interpret the codes of that domain in the same way. This provides a valuable clue about the conflicting codes so that the discussion of the meaning of the codes can be focused on these codes. Thus, a small value of *cu-α* points out potentially problematic codes, so that, during the review meeting, the coders can focus on the codes of these domains. Hopefully, this will lead to a more effective refinement of the codebook, which improves the ICA value of the next iteration more markedly.

The following sections describe the evolution of the agreement during the open coding activity of our GT study on EdgeOps. As we shall see, after the first iteration of the coding, there was no consensus on the meaning of the codes (*Cu-α* < 0.8). However, after refining the codebook and conducting a second iteration, the agreement improved to reach an acceptable threshold (*Cu-α* ≥ 0.80) so the initial coding was concluded.

Iteration 1

In the first iteration of the open coding process, R1, R2, and R3 analyzed 6 documents. R1 created a codebook with 29 codes that was subsequently refined by R2 and R3. As a by-product of this process, 40 codes were discovered and divided into 7 semantic domains (denoted by S1, S2, …, S7). After completion of the coding process, the *Cu-α* and *cu-α* ICA coefficients were calculated and their values are shown in Table 5.

As we can observe from this table, the value of the global coefficient *Cu-α* did not reach the acceptable threshold of 0.8. For this reason, a review meeting was necessary to discuss disagreements and the application criteria of the different codes. The results of this meeting are documented in the *disagreements diary* file in the *open coding* folder in the public repository.

To highlight problematic codes, we considered the coefficients *cu-α* computed per semantic domain. For Table 5, we observe that domain S3 got a remarkably low value of the coefficient *cu-α*. A detailed look at the particular codes within S3 shows that this domain includes codes related to the functionality of the system. This is particularly a fuzzy domain, in which several concepts can be confused. During the review meeting, clarifications about these codes were necessary to avoid misconceptions. After this, a new codebook was released. In this new version, memos and comments were added, and a code was removed, so 39 codes (and 7 semantic domains) proceeded to the second iteration of open coding.

Iteration 2

R1, R2, and R3 analyzed other 6 documents. Since the coders agreed on a common codebook in the previous iteration, we can expect a greater agreement that materializes as a higher value of ICA. As a by-product of this second iteration, 8 new codes emerged, leading to a new version of the codebook with 47 codes and 7 semantic domains. The ICA values for this second iteration are shown in Table 6.

From the results of this table we observe that, after this refinement of the codebook, *Cu-α* reached the acceptable threshold of agreement. In this way, the open coding process can stop: There exists consensus in the interpretation of the codes presented in the codebook, and we can proceed with the selection of core categories and selective coding.

5.3. Selection of core categories

In this activity, R1 and R2 selected the core categories, that is, the most relevant codes from the 47 codes obtained in open coding. To this end, we focused on the groundedness of the codes and semantic domains (i.e., the number of quotations linked to a code) and the density of the codes and semantic domains (i.e., the number of other codes connected to a code). The detailed analysis is documented in the *selection of core categories* file in the *selection of core categories* folder in the public repository. Fig. 4 shows an example of the multiple tables and graphics obtained from Atlas.ti that were analyzed during this activity. In this figure, the code "F01 local processing" is related to 11 codes 15 times. The code F01, on the left, is related to 11 codes (on the right) on 15 occasions (each link between the code on the left and one on the right has an associated number – not visible – that represents the width of the link and its sum is 15). As a result of the analysis, four semantic domains (S1, S2, S3, and S6) and 29 codes were selected for the next activity. This codebook is available in the *selection of core categories—codebook* file of the public repository.

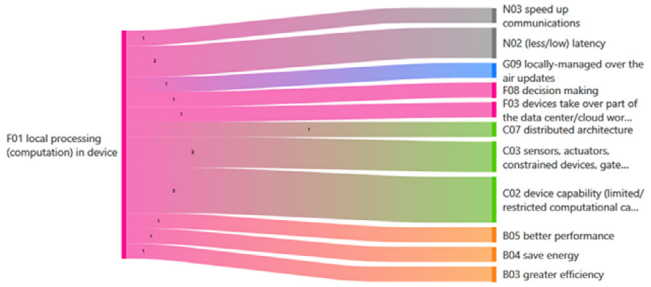


Fig. 4. Illustrative example of density analysis.

5.4. Selective coding

Recall from Section 4 that this is an inductive–deductive process in which new data are labeled with the codes of selected categories (semantic domains). Three coders (R1, R2, and R3) were again involved in this activity. The coders focused only on the core categories, but the number and definition of their inner codes were modified according to the analysis of new data.

After an iteration ends (that is, n documents have been coded by R1, R2, and R3), the ICA is calculated. If the value of *Cu-α* is below the acceptability threshold of 0.8, the coders meet as in Section 5.2 to refine the codebook. After polishing this new version of the codebook, a new iteration of selective coding is conducted to check whether they reach an acceptable agreement.

However, even if *Cu-α* passes the acceptability threshold, it can happen that some extra iterations of the coding process are needed. Indeed, to proceed with the following activity (sorting), it is mandatory that the new data analyzed do not introduce new information to the theory (the so-called theoretical saturation). For this reason, even if $Cu\text{-}\alpha \geq 0.8$, the coders must have a meeting to discuss whether theoretical saturation has been reached. If they decide that the saturation is not yet fulfilled, an additional iteration of selective coding must be conducted. After completing this new iteration, both the ICA (via *Cu-α*) and the theoretical saturation are analyzed. Only when both the ICA and saturation are satisfactory, the GT process can proceed to the next activity.

In the GT study described here, only one iteration was needed to fulfill both the ICA and the saturation criteria.

Iteration 1

In this iteration, R1, R2, and R3 analyzed 6 documents using S1, S2, S3, and S6, which encompass a total of 29 codes. After coding, 9 codes were added to the codebook accounting for a total of 38 core codes. This codebook is available in the *codebook* file in the *selective coding* folder in the public repository. The results of the ICA coefficients obtained after coding are shown in Table 7.

As we can observe from this table, the value of *Cu-α* reached the acceptable reliability threshold of 0.8. This shows that there exists a consensus among the coders on the meaning and limits of the codes within the core categories. Additionally, the coders also agreed that adding new data did not lead to new information, so the theoretical saturation had been reached. Therefore, since after this first iteration, the value of *Cu-α* was compelling and the coders agreed that theoretical saturation had been reached, the GT process could proceed to the next activity.

At this point, the proposed GT process coincides with the existing approaches in the literature: a sorting procedure followed by theoretical coding during which a theory emerges. Since the focus of this work is to improve the rigor and consensus of the codes elicited during open and selective coding procedures, for the sake of simplicity, we skip these subsequent standard GT phases.

Table 4
Data extraction.

ID	Epistemology Ontology	GT variant	Data collection method	GT Coding Phases	IRA	IRR	Correctly used?	Coefficient
01	Not mentioned	Straussian GT	Interviews data	Open coding Axial coding	–	IRR	No (IRA)	Cohen's κ
05	Not mentioned	Straussian GT	Video data	Not mentioned	–	IRR	No (IRA)	Fleiss κ
07	Not mentioned	Straussian GT	Literature	Open coding Axial coding Selective coding Const. comparison	–	IRR	No (IRA)	Krippendorff's α
08	Not mentioned	Straussian GT (inductive analysis)	Screen recordings Survey data Interviews data	Open coding Affinity diagramming Memoing	–	IRR	No (IRA)	Fleiss's κ
11	Not mentioned	Inductive analysis	Twitter data	Open coding	–	IRR	No (IRA)	κ (unknown version)
12	Not mentioned	Straussian GT	Interviews data	Open coding Axial coding	–	IRR	No (IRA)	Cohen's κ
13	Not mentioned	Not mentioned	Interviews data	Axial coding	IRA	–	Yes	Cohen's κ
17	Not mentioned	Not mentioned	Video data Text specifications	Not mentioned	–	IRR	No (IRA)	Percent agreement Cohen's κ
24	Not mentioned	Classic GT	Discourse files	Open coding Axial coding Selective coding	–	IRR	No (IRA)	κ (unknown version)
26	Constructivism	Straussian GT	Survey data		–	IRR	No (IRA)	Krippendorff's α
28	Not mentioned	Straussian GT	Survey data	Open coding	–	IRR	No (IRA)	Krippendorff's α
31	Not mentioned	Not mentioned	Focus group results	Not mentioned	–	IRR	Yes	Not mentioned
34	Not mentioned	Straussian GT	Interviews data	Open coding	–	IRR	No (IRA)	Cohen's κ
37	Not mentioned	Straussian GT	Interviews data	Open coding	IRA	–	Yes	Cohen's κ
41	Not mentioned	GT-like approach	Logs	Open coding Thematic analysis	IRA	–	Yes	Cohen's κ
44	Not mentioned	GT-like approach	Interview data	Open coding	IRA	–	Yes	Cohen's κ
55	Not mentioned	GT-like approach	Literature	Not mentioned	–	IRR	Yes	Kendall's
57	Not mentioned	GT-like approach	Interview data	Inductive analysis	–	IRR	No (IRA)	Cohen's κ
58	Constructivism	Charmaz GT	Interview data Survey data	Inductive analysis Memoing	IRA	–	Yes	Cohen's κ
59	Not mentioned	GT-like approach	Literature data		IRA	–	Yes	Krippendorff's α
60	Not mentioned	Straussian GT	Text specifications (users reviews)	Open coding Axial coding Selective coding	IRA	–	Yes	Cohen's κ
61	Not mentioned	GT-like approach	Text specifications		IRA	–		Fleiss's κ
62	Constructivism	Straussian GT	Survey data	Not mentioned	IRA	–	Yes	Cohen's κ
65	Not mentioned	Charmaz GT	Text specifications (functional requirements)	Not mentioned	IRA	–	Yes	Cohen's κ
66	Not mentioned	Straussian GT	Video data Text data (comments)	Not mentioned	IRA	–	Yes	Cohen's κ
67	Not mentioned	Straussian GT	Interview data	Open coding Axial coding Selective coding Const. comparison	–	IRR	No (IRA)	Cohen's κ
74	Not mentioned	GT-like approach	Interview data	Not mentioned	–	IRR	No (IRA)	Cohen's κ
78	Not mentioned	GT-like approach	Interview data	Content Analysis	–	IRR	No (IRA)	Cohen's κ

(*continued on next page*)

Table 4 (*continued*).

ID	Epistemology Ontology	GT variant	Data collection method	GT Coding Phases	IRA	IRR	Correctly used?	Coefficient
83	Not mentioned	GT-like approach	Survey data	Not mentioned	–	IRR	No (IRA)	Cohen's κ
87	Not mentioned	Charmaz GT	Case studies data: Text specifications and images (diagrams)	Memoing	IRA	–	Yes	Cohen's κ
88	Not mentioned	Straussian GT	Survey data	Open coding Axial coding	–	IRR	No (IRA)	Krippendorff's α
89	Not mentioned	GT-like approach	Interview data	Content analysis Thematic analysis	–	IRR	No (IRA)	Cohen's κ
92	Not mentioned	GT but variant is not specified	Instagram data	Open coding Axial coding Const. comparison	IRA	–	Yes	Cohen's κ
97	Not mentioned	GT but variant is not specified	Interview data	Initial coding Open coding Axial Coding	–	IRR	No (IRA)	Cohen's κ Scott's π
102	Not mentioned	GT-like approach	Logs Text specifications	Initial coding Axial coding	–	IRR	No (IRA)	Cohen's κ
110	Not mentioned	GT-like approach	Audio data	Not mentioned	–	IRR	No (IRA & IRR)	Cohen's κ Shaffer's ρ
111	Not mentioned	GT-like approach	Interview data Focused group data	Not mentioned	–	IRR	No (IRA)	Cohen's κ
128	Not mentioned	GT-like approach	Image data	Open coding	–	IRR	No (IRA)	Cohen's κ
132	Not mentioned	GT-like approach	Text specifications	Not mentioned		IRR	No (IRA)	Percent agreement Fleiss' κ
138	Not mentioned	Straussian-GT	Text specification	Open coding Axial coding Selective coding Content analysis	IRA	–	Yes	Cohen's κ
140	Not mentioned	GT but variant is not specified	Interview data Text specifications	Content analysis	–	IRR	No (IRA)	Kripendorff's α
144	Not mentioned	GT-like approach	Interview data	Not specified	–	IRR	No (IRA)	Cohen's κ
147	Not mentioned	GT-like approach	Literature data	Not specified	–	IRR	No (IRA)	Cohen's κ
151	Not mentioned	GT-like approach	Text specifications	Not specified	–	IRR		Pearson's r
153	Not mentioned	Straussian-GT	Text specifications	Open coding Axial coding Selective coding	–	IRR	No (IRA)	Cohen's κ
154	Not mentioned	GT-like approach	Text specifications	Not specified	IRA	–		Fleiss' κ
158	Not mentioned	GT but variant is not specified	Text specifications	Open coding Axial coding Selective coding Content analysis	–	IRR	No (IRA)	κ (unknown version)
161	Not mentioned	GT-like approach	Interview data	Not mentioned	IRA	–		Percent agreement
163	Not mentioned	GT-like approach	Interviews data Surveys (questionnaires)	Open coding Axial coding	–	IRR	No (IRA)	Cohen's κ

Table 5
Values of the different Krippendorff's α coefficients in the iteration 1 of the open coding. In bold, the values above the acceptability threshold (≥ 0.80).

cu-α per semantic domain							*Cu-α*
S1	S2	S3	S4	S5	S6	S7	
0.81	**0.98**	0.59	**0.80**	**1.00**	**1.00**	**1.00**	0.56

Table 6
Values of the different Krippendorff's α coefficients in the iteration 2 of the open coding. In bold, the values above the acceptability threshold ($\geq$ 0.80).

cu-α per semantic domain							*Cu-α*
S1	S2	S3	S4	S5	S6	S7	
0.72	**0.97**	**0.88**	**1.00**	**1.00**	**1.00**	**1.00**	**0.80**

Table 7
Values of the different Krippendorff's α coefficients in Iteration 1 of the selective coding phase. In bold, the values above the acceptability threshold ($\geq$ 0.80).

cu-α per semantic domain				*Cu-α*
S1	S2	S3	S6	
1.00	**0.95**	**0.87**	**1.00**	**0.80**

6. Threats to validity and limitations

The meta-science standard (see Empirical Standards for Software Engineering Research (Ralph, 2021)) guided us to analyze the issues and challenges of the GT method when various raters are involved in coding procedures and formalize a process to improve collaborative team science and consortia. To this end, we previously performed an SMS (see Section 3), and subsequently applied the process to a GT study (see Section 5). This section describes the threats to validity and limitations in both the SMS and the application case we addressed.

There are some techniques to mitigate sampling and publication bias in SMS that we did not address, such as backward and forward snowballing searches, searching on indexes (e.g. Google Scholar) in addition to formal databases, and searching for relevant dissertations or preprint servers (e.g. arXiv). However, on the basis of the results obtained, we consider that the narrative synthesis and empirical evidence from the 49 primary studies selected (those that met the inclusion and exclusion criteria of 168 unduplicated scientific papers) were sufficient to answer RQ1 and RQ2. A larger sample would not have provided new findings, but would have strengthened the evidence.

Quantitative quality criteria such as internal validity and construct validity do not apply, as this is not the kind of SMS that conducts meta-analysis to aggregate data for causal relationships between constructs. However, we do provide *replication package* including search terms and results, selection process results, coding examples, and complete synthesis results. The selection process (that is, the application of inclusion and exclusion criteria) was sufficiently rigorous for the mapping study goals, since two researchers participated in a dual selection process (as described in Section 3.1.3) and the IRA (specifically, the Krippendorff's α binary) was iteratively analyzed (as described in Section 3.2.2) to improve inclusion and exclusion criteria.

Conclusion validity concerns the relationship between treatment and outcomes (Wohlin et al., 2012), for example, how different researchers might have addressed data extraction and data synthesis differently. In our case, two researchers with different backgrounds independently extracted the data from primary papers without duplicity, as we considered that duality was not necessary as the data to be extracted are totally objective. We provide some coding examples using Atlas.ti available in the replication package. Additionally, we have extensively used quotations to establish credibility in the qualitative sense of chain-of-evidence (see Section 3.2.4).

Finally, with respect to the process described here and its application to a GT study in the domain of *EdgeOps in industry*, the main concern is external validity and generalizability, which typically does not apply to case studies in which the effort to demonstrate feasibility is enormous, as it requires the execution of multiple cases (ie, multiple GT studies) from data collection to theory generation. Thus, we can only assert that the application of this process to a GT study seems to support its feasibility. In the absence of further confirmation, this would represent the first step of a de facto standard to be applied to those GT studies that require IRR/IRA.

7. Related work

The inclusion of quantitative techniques in qualitative research and the need to follow clear guidelines and a sound methodology is not new (Creswell and Creswell, 2017). Specifically, in the discipline of Information Systems, Venkatesh et al. (2013) analyzed the use of intercode reliability to measure validity and developed a meta-tool to guide researchers in combining both quantitative and qualitative methods, aiming at high-level epistemological issues that researchers should approach when combining both methods. However, the number of publications that test the reliability of coding is notably higher in areas such as health sciences, social psychology, education and business than in computer science and, specifically, in information management research (Nili et al., 2020).

Specifically, and sharing our objectives, several authors have tried to systematize the role of multiple coders and IRR/IRA in the qualitative research paradigm, ranging from phenomenology (Marques and Mccall, 2005) to constant comparative analysis (Olson et al., 2016) and content analysis (Nili et al., 2020) methods.

Olson et al. (2016) addressed the inclusion of a positivist term such as "reliability" in the qualitative paradigm. Specifically, they proposed a 10-step method for applying the constant comparative method of GT when multiple researchers perform data analysis by measuring inter-coder reliability through Fleiss' κ coefficient as follows:

1. Each researcher performs open-coding of a subset of data,
2. Collaborates to unify codes, and
3. Recodes the subset of data using unified codes.
4. The inter-coder reliability is calculated.
5. Researchers collaborate to discuss each code and identify areas lacking agreement, and
6. Repeat the above process for more subsets of data, producing a unified codebook.
7. Researchers recode all data, producing themes,
8. Select themes for further analysis,
9. Conduct co-occurrence analysis, and
10. Construct an exploratory model – the findings of the study.

These authors also reported that they felt so constrained by the use of inter-coder reliability during coding that it led to loss of meaning. The search for a *good* value for inter-coder reliability distorted the purpose of coding to the extent of being more concerned with coincidence with other researchers than with meaningful coding. This is why the authors shifted the

interpretation of Fleiss' κ from a quantitative verification tool to a solidification tool, i.e., a tool to "guide collaboration and identify nuances in the data brought to light by our prior experience, knowledge, and perspectives". Thus, the authors ponder upon the role of inter-coder reliability as a *solidification* tool, which is a concept borrowed from the use of IRR/IRA into the constructivist phenomenological paradigm (Marques and Mccall, 2005), and that we also adopt.

Closer to our interest is the work of Nili et al. (2020), which focuses on the practical issues of applying IRR/IRA to qualitative methods (also circumscribed to Information Systems discipline). In this work, the authors provide guidelines to decide on the most suitable statistical instrument for IRR/IRA and a 5-step approach to perform IRA/IRR in qualitative studies, namely:

1. Selecting an inter-coder reliability method
2. Developing a coding scheme
3. Selecting and training independent coders,
4. Calculating the inter-coder reliability coefficient (which may lead to continuing the training session and iteratively coding the entire dataset),
5. Reporting the process of evaluating inter-coder reliability along with the result.

With a possible cycle of iterations, like Olson, from Step 4 to Step 2, both works coincide in excluding IRR/IRA calculation from the first phase of open-coding phase. Open coding of Step 2 is performed by only one coder and the codebook is said to be constructed both inductively (from raw data) or derived from previous literature. Therefore, in this methodology, IRR/IRA seems to play a secondary role at first (or even no role), and it is used as a posteriori checking/verification.

As we can observe from this method, there is an open coding phase performed by all coders to create a first version of the codebook which is not subject to inter-coder reliability calculation. Then, the same data is re-codified and IRR/IRA is calculated. No agreement threshold is sought, but it is used as a tool to unveil disagreement areas and possible coder behavior patterns (Step 5). These five steps are repeated for all data samples, which means that before entering into the next phase of coding (selective or thematic), the data is passed over and codified several times.

8. Conclusion

Qualitative research, and GT as one instance, is often tarnished by epistemological debates like the validity and reliability of obtained knowledge. When applied to computer science (CS), the epistemological position is usually not clear. However, it is not uncommon to apply quantitative instruments in qualitative research as a possible way to confer validity and methodological strength to the researches carried out in the qualitative paradigm. We have focused on the use of quantitative instruments such as IRR/IRA techniques in GT studies. As shown in Section 3, GT-driven research in CS usually presents some deficiencies when dealing with epistemological and methodological issues that support the validity and reliability of their outcomes: self-allegedly GT studies do not clarify which GT school/trend they adhere to, thus using GT terminology confusingly and reducing GT methodology to the mere use of coding procedures. In addition, the IRR/IRA instruments are sometimes poorly used, like confusing the concepts of reliability and agreement, and above all, using these statistical instruments with no further purpose in the study.

Convinced that we are of the utility and essential role in the science of qualitative research and aware of the validity and quality issues of the obtained results, we have formalized a process to integrate IRR/IRA into qualitative GT-based research that allows researchers to rigorously use these statistical techniques for measuring reliability and agreement during the coding process, thus fostering consensus and reflection. We do not consider the notion of reliability as trying to establish a single reality, but rather as an approach for developing a shared understanding that can also establish consistency among coders. Our method is independent of, and should fit, different families of GT theory. It is targeted at those who decide to validate consensus and shared understanding in teamwork during coding processes. The process was validated with a case of study (limited in scope and extension) to prove its feasibility. It is a limited case study in extension but not in depth, focusing on open and selective coding phases. Finally, there is no definitive and correct way to handle validity and reliability in qualitative research. Our intention is to define the first steps towards the definition of a *de facto* standard to be applied to those GT studies that would benefit from the use of IRR/IRA instruments. We are working in a continuous improvement of the process, such as a pre-meetup among collaborative coders to establish a common understanding of the concepts that are the primary focus of the investigation/study under investigation.

CRediT authorship contribution statement

Jessica Díaz: Conceptualization, Investigation, Methodology, Writing – original draft, Supervision, Visualization, Validation. **Jorge Pérez:** Conceptualization, Investigation, Methodology, Writing – original draft, Supervision, Formal analysis, Validation. **Carolina Gallardo:** Investigation, Writing – original draft, Supervision, Data curation, Investigation, Writing – review & editing. **Ángel González-Prieto:** Investigation, Supervision, Data curation, Formal analysis, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Link to supplementary materials in a long-term archive: https://doi.org/10.5281/zenodo.5034244. The data of the GT study on EdgeOps were collected from an open-ended questionnaire available in https://es.surveymonkey.com/r/PMWD7ZM.

Acknowledgments

The authors would like to thank Paul Ralph for his valuable review and suggestions. The fourth named author acknowledges the hospitality of the Department of Mathematics at Universidad Autónoma de Madrid, where part of this work was completed.

Appendix. Krippendorff's α for ICA

This appendix describes the following two versions of Krippendorff's α coefficients:

- The coefficient *cu*-α: This coefficient is computed on a specific semantic domain *S*. It indicates the degree of agreement with which coders identify codes within *S*.
- The coefficient *Cu*-α: This coefficient measures the degree of agreement in the decision to apply different semantic domains, independent of the chosen code.

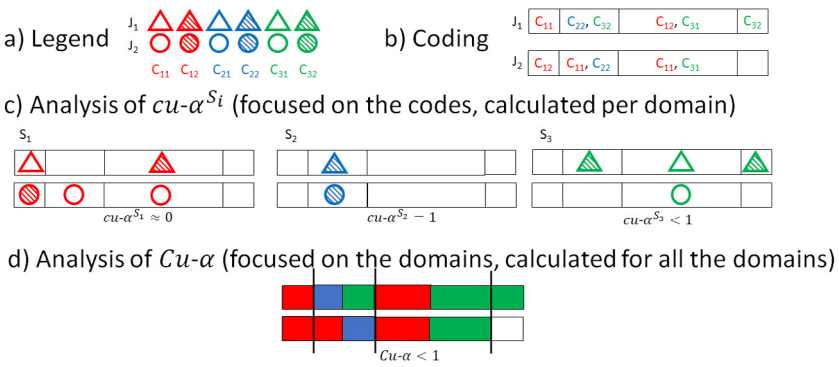


Fig. A.5. Illustrative example of the Krippendorff's α coefficients. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For the convenience of the reader, we provide a running example of the use of these coefficients. This case of use has been extracted from Díaz et al. (2021) (see also Perez et al. (2021)). Fig. A.5 shows an illustrative example of the use of these coefficients. Let three semantic domains and their respective codes be as follows:

$$S_1 = \{C_{11}, C_{12}\}, \quad S_2 = \{C_{21}, C_{22}\}, \quad S_3 = \{C_{31}, C_{32}\}.$$

Coder 1 and Coder 2 assign codes to four quotations, as shown in Fig. A.5(a), such that the first quotation is assigned C_{11} by Coder 1 and C_{12} is assigned by Coder 2. We create a graphical metaphor so that each coder, each semantic domain, and each code are represented as shown in Fig. A.5(b). Each coder is represented by a shape, such that Coder 1 is represented by triangles and Coder 2 is represented by circles. Each domain is represented by a color: S_1 is red, S_2 is blue, and S_3 is green. Each code within the same semantic domain is represented as a fill, where C_{i1} codes are represented by a solid fill and C_{i2} codes are represented by dashed fills.

The coefficient $cu\text{-}\alpha$ is calculated per domain (i.e., S_1 red, S_2 blue, S_3 green), but it measures the agreement attained when applying the codes of that domain. In other words, given a domain S_i, this coefficient analyzes whether the coders assigned the same codes of S_i (i.e., the same type of fill) to the quotations or not. In this way, Fig. A.5(c) only focuses on the fills applied to each quotation. In particular, it is shown that $cu\text{-}\alpha = 1$ for S_2, since both coders assigned the same code to the second quotation, but no code from this domain was assigned to the rest of the quotations, i.e., total agreement. Additionally, it is shown that $cu\text{-}\alpha < 1$ for S_3, as the coders assigned the same code of S_3 to the third quotation 3, but they did not assign the same codes of S_3 to the rest of the quotations. Finally, it is shown that the $cu\text{-}\alpha$ coefficient for S_1 is very small (near zero) since the coders achieved no agreement on the chosen codes (the exact value of $cu\text{-}\alpha$ will depend on the expected disagreement, which depends on the marginal frequencies of each code).

On the other hand, the coefficient $Cu\text{-}\alpha$ analyzes all domains as a whole, but it does not take into account the codes within each domain. In this way, in Fig. A.5(d), we color each segment with the colors corresponding to the applied semantic domain (regardless of the particular code used). From these chromatic representations, $Cu\text{-}\alpha$ measures the agreement in applying these colors globally among the coders. In particular, note that $Cu\text{-}\alpha < 1$, as both coders assigned the same domain S_1 to the first quotations, and they assigned domains S_1 and S_3 to the third quotation, but they did not assign the same domains in the second and fourth quotations.

The larger the α coefficients are, the better the observed agreement. Typically, the α coefficients lie in the range of $0 \leq \alpha \leq 1$. A common rule-of-thumb in the literature (Krippendorff, 2018) is that $\alpha \geq 0.667$ is the minimal threshold required for drawing conclusions from the data. For $\alpha \geq 0.80$, we can consider that there exists statistical evidence of reliability in the coding. A thorough explanation of the use of these coefficients and their interpretation can be found in González-Prieto et al. (2020).

References

Armstrong, D., Gosling, A., Weinman, J., Marteau, T., 1997. The place of inter-rater reliability in qualitative research: An empirical study. Sociology 31 (3), 597–606. http://dx.doi.org/10.1177/0038038597031003015.

Auerbach, C.F., Silverstein, L.B., 2003. Qualitative Data: An Introduction to Coding and Analysis. New York University Press, New York.

Braun, V., Clarke, V., 2013. Successful Qualitative Research. Sage.

Campbell, J.L., Quincy, C., Osserman, J., Pedersen, O.K., 2013. Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement. Sociol. Methods Res. 42 (3), 294–320. http://dx.doi.org/10.1177/0049124113500475.

Charmaz, K., 2014. Constructing Grounded Theory. Sage 2nd Ed..

Charmaz, K., Thornberg, R., 2020. The pursuit of quality in grounded theory. Qual. Res. Psychol. 1–23. http://dx.doi.org/10.1080/14780887.2020.1780357.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20 (1), 37–46.

Cornish, F., Gillespie, A., Zittoun, T., Collaborative Analysis of Qualitative Data. SAGE Publications Ltd, pp. 79–93. http://dx.doi.org/10.4135/9781446282243.

Creswell, J.W., Creswell, J.D., 2017. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Sage publications.

Cruzes, D.S., Dyba, T., 2011. Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement. IEEE, pp. 275–284.

Díaz, J., López-Fernández, D., Pérez, J., González-Prieto, Á., 2021. Why are many businesses instilling a DevOps culture into their organization? Empir. Softw. Eng. 26 (2), 1–50.

Dubé, L., Paré, G., 2003. Rigor in information systems positivist case research: current practices, trends, and recommendations. MIS Quart. 597–636.

Erickson, K., Stull, D., 1998. Doing Team Ethnography: Warnings and Advice.. Sage, Thousand Oaks, CA.

Gibbs, G.R., 2007. Analyzing Qualitative Data. Qualitative Research Kit. Sage publications.

Gisev, N., Bell, J.S., Chen, T.F., 2013. Interrater agreement and interrater reliability: key concepts, approaches, and applications. Res. Soc. Admin. Pharmacy 9 (3), 330–338.

Glaser, B., Strauss, A.L., 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine de Gryter, New York.

González-Prieto, Á., Perez, J., Díaz, J., López-Fernández, D., 2020. Inter-coder agreement for improving reliability in software engineering qualitative research. arXiv:2008.00977.

Guest, G., MacQueen, K.M., 2008. Handbook for Team-Based Qualitative Research. AltaMira Press, Lanham, MD.

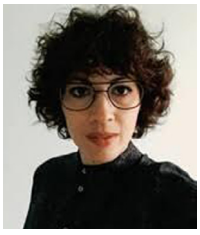
Hall, W.A., Long, B., Bermbach, N., Jordan, S., Patterson, K., 2005. Qualitative teamwork issues and strategies: Coordination through mutual adjustment. Qual. Health Res. 15 (3), 394–410. http://dx.doi.org/10.1177/1049732304272015.

Hammer, D., Berland, L.K., 2014. Confusing claims for data: A critique of common practices for presenting qualitative research on learning. J. Learn. Sci. 23 (1), 37–46. http://dx.doi.org/10.1080/10508406.2013.802652.

Hayes, A.F., Krippendorff, K., 2007. Answering the call for a standard reliability measure for coding data. Commun. Methods Measures 1 (1), 77–89. http://dx.doi.org/10.1080/19312450709336664.
Hoda, R., 2021. Decoding grounded theory for software engineering. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). pp. 326–327. http://dx.doi.org/10.1109/ICSE-Companion52605.2021.00139.
Kenny, M., Fourie, R., 2015. Contrasting classic, straussian, and constructivist grounded theory: Methodological and philosophical conflicts. Qual. Rep. 20 (8), 1270–1289. http://dx.doi.org/10.46743/2160-3715/2015.2251.
Kitchenham, B.A., Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report, URL https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf.
Krippendorff, K., 2004. Reliability in content analysis: Some common misconceptions and recommendations. Hum. Commun. Res. 30 (3), 411–433. http://dx.doi.org/10.1111/j.1468-2958.2004.tb00738.
Krippendorff, K., 2011. Computing krippendorff's alpha-reliability. Retrieved from http://repository.upenn.edu/asc_papers/43.
Krippendorff, K., 2018. Content Analysis: An Introduction to Its Methodology, fourth ed. Sage publications.
Krippendorff, K., Mathet, Y., Bouvry, S., Widlöcher, A., 2016. On the reliability of unitizing textual continua: Further developments. Qual. Quant. Int. J. Methodol. 50 (6), 2347–2364. http://dx.doi.org/10.1007/s11135-015-0266-1, URL https://ideas.repec.org/a/spr/qualqt/v50y2016i6d10.1007_s11135-015-0266-1.html.
Leite, L., Pinto, G., Kon, F., Meirelles, P., 2021. The organization of software teams in the quest for continuous delivery: A grounded theory approach. Inf. Softw. Technol. 139, 106672.
Lincoln, Y., Guba, E., 1985. Naturalistic Inquiry. Sage, Beverly Hills, CA.
López-Fernández, D., Diaz, J., Garcia-Martin, J., Pérez, J., Gonzalez-Prieto, A., 2021. Devops team structures: Characterization and implications. IEEE Trans. Softw. Eng..
Luz, W.P., Pinto, G., Bonifácio, R., 2019. Adopting DevOps in the real world: A theory, a model, and a case study. J. Syst. Softw. 157, 110384.
MacPhail, C., Khoza, N., Abler, L., Ranganathan, M., 2016. Process guidelines for establishing Intercoder Reliability in qualitative studies. Qual. Res. 16 (2), 198–212. http://dx.doi.org/10.1177/1468794115577012.
Marques, J., Mccall, C., 2005. The application of interrater reliability as a solidification instrument in a phenomenological study. Qual. Rep. 10, http://dx.doi.org/10.46743/2160-3715/2005.1837.
McDonald, N., Schoenebeck, S., Forte, A., 2019. Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. Proc. ACM Hum.-Comput. Interact. 3 (CSCW), http://dx.doi.org/10.1145/3359174.
Nili, A., Tate, M., Barros, A., 2017. A critical analysis of inter-coder reliability methods in information systems research. In: Proceedings of the 28th Australasian Conference on Information Systems. University of Tasmania, pp. 1–11.
Nili, A., Tate, M., Barros, A., Johnstone, D., 2020. An approach for selecting and using a method of inter-coder reliability in information management research. Int. J. Inf. Manage. 54, 102154. http://dx.doi.org/10.1016/j.ijinfomgt.2020.102154.
O'Connor, C., Joffe, H., 2020. Intercoder reliability in qualitative research: Debates and practical guidelines. Int. J. Qual. Methods 19, 1609406919899220. http://dx.doi.org/10.1177/1609406919899220.
Olson, J., McAllister, C., Grinnell, L., Walters, K., Appunn, F., 2016. Applying constant comparative method with multiple investigators and inter-coder reliability. Qual. Rep. 21, 26–42. http://dx.doi.org/10.46743/2160-3715/2016.2447.
Patton, M., 1999. Enhancing the quality and credibility of qualitative analysis. Health Serv. Res. 34 (5 Pt 2), 1189–1208.
Pérez, J., Díaz, J., Garcia-Martin, J., Tabuenca, B., 2020. Systematic literature reviews in software engineering—enhancement of the study selection process using cohen's kappa statistic. J. Syst. Softw. 168, 110657, URL https://www.sciencedirect.com/science/article/pii/S0164121220301217.
Perez, J., Gonzalez-Prieto, A., Diaz, J., Lopez-Fernandez, D., Garcia-Martin, J., Yague, A., 2021. DevOps research-based teaching using qualitative research and inter-coder agreement. IEEE Trans. Softw. Eng..
Ralph, P.e., 2021. Empirical standards for software engineering research. URL arXiv:2010.03525v2 [cs.SE].
Richards, K.A.R., Hemphill, M.A., 2018. A practical guide to collaborative qualitative data analysis. J. Teach. Phys. Educ. 37 (2), 225–231. http://dx.doi.org/10.1123/jtpe.2017-0084, URL https://journals.humankinetics.com/view/journals/jtpe/37/2/article-p225.xml.
Saldaña, J., 2012. The Coding Manual for Qualitative Researchers. Sage publications.
Stol, K.-J., Ralph, P., Fitzgerald, B., 2016. Grounded theory in software engineering research: A critical review and guidelines. In: Proceedings of the 38th International Conference on Software Engineering. ICSE '16, Association for Computing Machinery, New York, NY, USA, pp. 120–131. http://dx.doi.org/10.1145/2884781.2884833.
Strauss, A., Corbin, J., 1990. Basics of Qualitative Research: Grounded Theory Procedures and Techniques. SAGE Publication, London.
Venkatesh, V., Brown, S.A., Bala, H., 2013. Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. MIS Quart. 37 (1), 21–54.
Weston, C., Gandell, T., Beauchamp, J., McAlpine, L., Wiseman, C., Beauchamp, C., 2001. Analyzing interview data: The development and evolution of a coding system. Qual. Sociol. 24 (3), 381–400. http://dx.doi.org/10.1023/10.1023%2FA%3A10106909082000.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering. Springer Science & Business Media.
Wu S., W., C., D., Fraser, M.W., 2016. Author guidelines for manuscripts reporting on qualitative research. J. Soc. Soc. Work Res. 7, 405–425. http://dx.doi.org/10.1086/685816.

Jessica Díaz is an Associate Professor at Universidad Politécnica de Madrid (UPM, E.T.S. Ingeniería de Sistemas Informáticos), Spain. She received her PhD degree in Computer Science from UPM in 2012 (best thesis Award). Since April 2003, she is a researcher in System and Software Technology Research Group (SYST) and is participating in several European and national projects related to Software Engineering on Internet of Things and Smart Systems. Her research interests are focused on DevOps and Rapid Application Development, Cloud Computing, Software Architectures, Software Product Lines and Model-driven Development.

Jorge E. Pérez received the B.S. degree in computer science from the Universidad Carlos III de Madrid, Spain, in 1999, and the Ph.D. degree from the UPM, in 2004. From 2007 to 2014, he was the Headmaster of the Department of Applied Computer Science, UPM, where he has been an Associate Professor, since 1985. He is currently the coordinator of the DMAE-DIA Educational Innovation Group. He has taken part in the creation of several syllabuses for both grades and Postgraduates and has supervised many research projects. His research interests include Software Architectures and meta-modeling, and qualitative research in Software Engineering. Dr. Pérez is a recipient of several awards from the Rector of the UPM for educational innovation at the university.



Carolina Gallardo holds a PhD from Universidad Politécnica de Madrid in the Computer Science and Artificial Intelligence program; Master's degree in Information Systems Management from Universidad de Cádiz (2016), a degree in English Philology (1998) and a degree in Linguistics (2000) from Universidad Autónoma de Madrid. From 2006, she is Associate Professor at Escuela Técnica Superior de Sistemas Informáticos. Since 2000, she belongs to the Validation and Industrial Applications research group. She is a member of the UNESCO Chair in Language Technologies. She is co-author of 13 software registrations, all in the field of NLP in Spanish (applications for analysis and generation of Spanish, mono and multilingual dictionary applications, document classification tools).

Ángel González-Prieto received a Double B.Sc. in Computer Sciences and Mathematics from Universidad Autónoma de Madrid in 2014, a M.Sc. in Mathematics from the same university in 2015 and a Ph.D. in Mathematics from Universidad Complutense de Madrid in 2018 (Extraordinary Thesis Award). As recognition of his research works, he received the 2021 Vicent Caselles Award of Mathematical Research. Since 2021, he is Assistant Professor at Universidad Complutense de Madrid, and currently he also serves as Vicedean for International and Institutional Affairs of the Facultad de Ciencias Matemáticas at the same university. His research interests include statistics, machine learning, software engineering and algebraic geometry.