Chapter 10

Qualitative Synthesis

10.1 Qualitative synthesis in software engineering research 112
10.2 Qualitative analysis terminology and concepts 113
10.3 Using qualitative synthesis methods in software engineering systematic reviews 116
10.4 Description of qualitative synthesis methods 117
10.4.1 Meta-ethnography 118
10.4.2 Narrative synthesis 120
10.4.3 Qualitative cross-case analysis 121
10.4.4 Thematic analysis 123
10.4.5 Meta-summary 124
10.4.6 Vote counting 127
10.5 General problems with qualitative meta-synthesis 129
10.5.1 Primary study quality assessment 129
10.5.2 Validation of meta-syntheses 130

This chapter discusses qualitative methods for synthesizing research studies. In most cases, qualitative synthesis methods are used when the individual primary studies used qualitative research methods, or used a variety of different experimental methods. In the context of software engineering, industrial case studies are a particularly important form of primary study because they provide more realistic information about the extent to which new methods and tools scale-up to the complexity of industrial scale software development than laboratory experiments. As discussed in Chapter 18 and Chapter 19, case studies often adopt qualitative methods. They, therefore, require qualitative approaches, such as the ones described in this chapter, to synthesise their results.

Qualitative synthesis methods are also useful for synthesising data from experiments, quasi-experiments, and data mining studies when the differences among outcome metrics, analysis methods, and experimental designs are too great to make statistical meta-analysis feasible. For this situation, we recommend *vote counting*. Vote counting is the practice of counting the number of primary studies that found a significant positive effect and the number that found an insignificant effect (or a significant negative effect) and assuming the effect is real if the majority of the studies are significant. Although vote counting is sometimes assumed to be a form of meta-analysis, many meta-analysts are strongly opposed to its use. The main argument is that although

a significant finding provides evidence that an effect exists, a non-significant finding does not indicate that there is no effect, because lack of significance can be due to low statistical power. In addition, vote counting may give an idea of the direction of an effect but it does not give any indication of the magnitude of the effect, so it is not possible to decide whether an effect is practically important as well as statistically significant. However, in practice, many software engineering researchers, ourselves included, adopt vote counting when it is not possible to undertake a proper meta-analysis. We agree with Popay et al. (2006) that vote counting can be used constructively as part of a narrative synthesis, particularly if it can be associated with some form of qualitative moderator analysis.

10.1 Qualitative synthesis in software engineering research

Before discussing qualitative methods for synthesis, we discuss the extent to which qualitative synthesis is important for software engineering research. Cruzes & Dybå (2011*b*) reviewed the state of research synthesis in software engineering systematic reviews. They undertook a *tertiary study* that identified 49 systematic reviews published between the 1st of January 2005 and the 31st of July 2010. They found that the methods authors claimed to have used for synthesis were not always correct. They also reported that:

- 24 studies were mapping studies not systematic reviews
- 22 of the systematic reviews were not explicit about the synthesis method they used.
- Meta-analysis was used in only two systematic reviews (see Kampenes, Dybå, Hannay & Sjøberg (2007) and Dybå et al. (2006)), and, excluding mapping studies, all other systematic reviews used qualitative methods.
- Narrative synthesis was the most common form of synthesis (9 systematic reviews), followed by thematic analysis (8 systematic reviews) and comparative analysis(4 systematic reviews).
- Meta-ethnography and case survey were each used by one systematic review.

Cruzes & Dybå's study confirms the importance of qualitative synthesis for systematic reviews, but also, suggests that software engineering researchers are not good at describing the methods they use to aggregate and synthesise non-numerical findings.

Later studies indicate that the use of qualitative synthesis in software engineering systematic reviews continues to increase. Another tertiary study (da Silva et al. 2011) identified a second systematic review that used meta-ethnography (Gu & Lago 2009), while more recently Da Silva, F. Q. B.; Cruz, S. S. J. O.; Gouveia, T. B.; & Capretz, L. F (2013) reported a meta-ethnography of four primary studies presented as a worked example of the method. In addition, Cruzes, Dybå, Runeson & Höst (2014) present a study based on synthesising two case studies related to trust in outsourcing which used three different methods: thematic synthesis, cross-case analysis and narrative synthesis.

Also, the use of meta-analysis was underestimated with meta-analyses by Hannay et al. (2009), Ciolkowski (2009), and Salleh, Mendes & Grundy (2009) being missed by Cruzes & Dybå's tertiary study. In addition, at least, two more meta-analyses were published after 2010 (see Rafique & Misic (2013), and Kakarla, Momotaz & Namim (2011)).

Before discussing specific qualitative synthesis methods, we discuss some of the terminology used in the context of qualitative analysis. We then discuss the specific methods we believe are of most relevance to software engineering qualitative aggregation and synthesis. In this chapter, some of our methodological references come from the healthcare domain, in particular, nursing and healthcare policy. This is because this domain has a long history of qualitative research and has been grappling with the problems of synthesising qualitative research for many years. Furthermore, methodological studies using health care examples discuss topics that are familiar to most of us, for example, promoting healthy eating practices, or caring for sick children, which makes them easier to understand than examples from other domains.

10.2 Qualitative analysis terminology and concepts

Throughout this chapter, we will use the term *meta-synthesis* to apply to any method of qualitative aggregation or synthesis, that is, every form of aggregation or synthesis except quantitative meta-analysis. It is important to understand that most qualitative analysts view aggregation and synthesis as very different activities.

Aggregation is assumed to be similar to quantitative meta-analysis where information from different primary studies is combined together using counts and averages. For example, *quantitative content analysis* involves counting the number of times some specific words or phrases are mentioned in text. This is a rather quantitative approach to analysis and if it was used to obtain information from a set of primary studies would equate to an aggregation-based synthesis. Novice analysts usually find that aggregation is much easier than synthesis, but is only suitable for use with qualitative primary studies

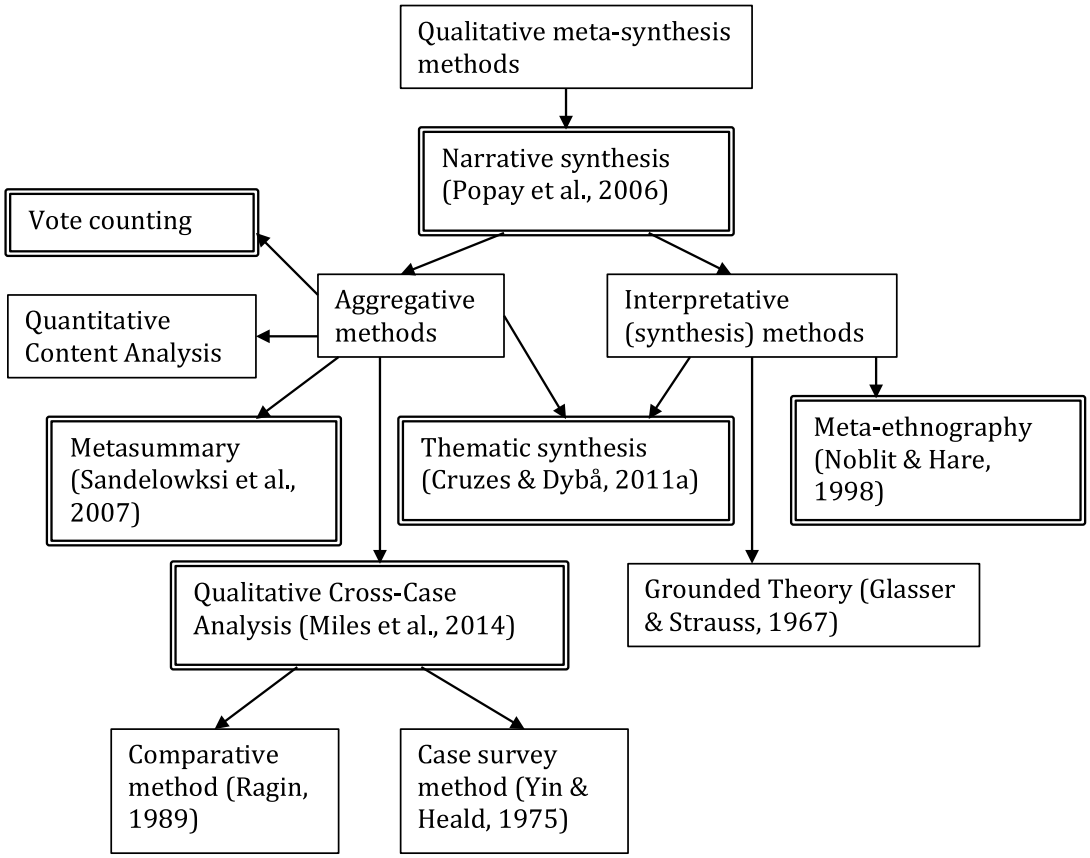

FIGURE 10.1: Methods for qualitative synthesis.

that have used fairly simple approaches to reporting their findings, such as content analysis or simply reporting the topics mentioned by participants.

In contrast, the goal of more purely qualitative studies is synthesis. *Synthesis* is referred to as an *interpretive* process which means using the concepts defined in specific primary studies to construct *higher order* models, that is, models that include concepts not found in any primary study. For example, in studies of globally-distributed software development, primary study authors may report problems observed by individuals working on distributed projects, whereas authors of secondary studies might use the information about reported problems to *infer* or *hypothesise* underlying causes of the problems which were *not mentioned specifically in any of the primary studies*. As a concrete example, Casey & Richardson (2008) re-analysed three case studies undertaken over a period of eight years. Although the only similarity between the cases was the shared aim of finding out what was actually going on and identifying what positive and negative factors influenced the software development strategy, the authors were able to identify the fear of job loss among staff employed by the client company as a factor that explained many of the problems observed between clients and vendors in each of the cases.

In general, the qualitative method used in the primary studies will influence the type of meta-synthesis that can be performed. Two types of qualitative method that are common in disciplines such as health care, psychology and social policy, are *ethnography* and *phenomenology*.

Ethnography is used to undertake longitudinal studies aimed at understanding the social and societal behaviour of human groups. Noblit & Hare (1988) developed *meta-ethnography* as a method for synthesizing different ethnography studies. In the context of software engineering, observational studies of agile teams might be based on an ethnography-based approach, see for example, Sharp & Robinson (2008) and Robinson, Segal & Sharp (2007).

Phenomenology is concerned with the way in which individuals perceive and interpret events. Phenomenology can underpin the use of *Grounded Theory*, which has the main aim of developing theory from the observed data. In the context of software engineering, Oza, Hall, Rainer & Grey (2006) present a Grounded Theory analysis of trust in outsourcing projects. Grounded Theory has had a major influence on qualitative research. Remenyi (2014) says that:

> "Grounded Theory not only offers a method by which social science research may be rigorously conducted but it also provides a more general explanation and understanding of how qualitative research works."

Terminology originating from grounded theory is used by many different qualitative synthesis and meta-synthesis methods and includes:

- *Coding* which involves applying descriptive labels to pieces of textual fragments, such as words, phrases or sentences. Miles, Huberman & Saldaña (2014) point out that words are the basic medium for all qualitative analysis, irrespective of the way in which the raw data was obtained. Initially, analysts look for codes that can be used to identify related textual fragments in different sources.

- *Axial coding* is an additional level of coding used to organise the basic codes derived directly from the text into more comprehensive concepts. This is also referred to as second order or second-level coding.

- *Theoretical sampling* or *purposeful sampling* aims to find data from a wide range of sources to increase understanding of the topic of interest. It assumes that information obtained from one source might raise novel issues leading the researcher to look for new types of data.

- *Theoretical Saturation* is the mechanism used to determine the completion of the theoretical sampling process. It occurs when obtaining additional data does not appear to be adding any new insight to the topic of interest. In secondary studies, the concepts of theoretical sampling and theoretical saturation are contrary to the concept of a search and selection process pre-defined in a study protocol. However, both approaches

can be integrated if theoretical sampling and theoretical saturation are used as the final selection process applied to a set of related studies found by a pre-defined search process.

- *Continuous comparison* involves always comparing data from one situation with data found in another. In primary studies, this comparison is at the data level. In secondary studies, comparisons are based on the interpreted theories produced by primary studies (see the comments on *substantive theory* and *formal theory* below).

- *Memoing* refers to notes that a researcher makes to him/herself. They may be simple comments that one data item seems to resemble an item found in a previous text, or *analytic memos* that record initial ideas about higher level codes or themes.

- *Substantive theory* refers to the outcome of grounded theory. It is a theory that is derived from the data and is bounded by the context in which the data was obtained. It may not be generalisable to other situations.

- *Formal theory* is more generalised than substantive theory. Formal theories are sometimes referred to as *mid-level theories*. The originators of grounded theory suggested substantive theories produced in different studies could be synthesised into more general formal theories (Glaser & Strauss 1967). Kearney (1998) discusses the use of grounded theory to produce formal theories in the context of qualitative meta-synthesis.

A more detailed discussion of the philosophical basis of various qualitative meta-synthesis methods can be found in Barnett-Page & Thomas (2009).

10.3 Using qualitative synthesis methods in software engineering systematic reviews

If we attempt a qualitative meta-synthesis of interpretive qualitative studies, our qualitative meta-synthesis will involve interpreting the interpretations of the primary study authors. This, and the issue that synthesis may remove the contextual details necessary to fully understand qualitative findings, has led some qualitative researchers to suggest that the goal of qualitative meta-synthesis is inherently flawed.

However, many well-respected qualitative researchers believe that qualitative synthesis is essential to inform practice, see for example the discussion in Sandelowski, Docherty & Emden (1997). Nonetheless, we accept the warning of experienced qualitative researchers that undertaking qualitative

meta-synthesis is difficult even for experienced researchers, let alone novices (Thorne, Jensen, Kearney, Noblit & Sandelowski 2004). However, we note pragmatically:

- Organisations such as the Cochrane Collaboration and the University of York Centre for Reviews and Dissemination both recommend incorporating qualitative synthesis with quantitative reviews in their systematic review handbooks, see Noyes & Lewin (2011) and CRD (2009). These reports make it clear that, in the context of health care, qualitative meta-synthesis can and should contribute to quantitative reviews of intervention effectiveness by helping to specify important research questions (that is, ones that matter to patients), providing evidence that explain variations in outcomes (for example, detailed investigations of variations in interventions, participants, and settings), and supplying complementary evidence related to aspects other than effectiveness (such as acceptability to patients). We believe it should be particularly useful in a domain such as software engineering, where relatively few primary studies are suitable for quantitative meta-analysis.
- In our experience, qualitative studies in software engineering report participants' viewpoints with relatively little interpretation being performed by the researchers. Such studies can be aggregated using the metasummary method (Sandelowski & Barroso 2003) and thematic synthesis (Cruzes & Dybå 2011*a*) which make less stringent demands on the expertise of the analysts. These techniques are discussed in more detail in Section 10.4.

10.4 Description of qualitative synthesis methods

This section briefly describes the qualitative synthesis methods we judge to be most relevant to software engineering researchers. These are methods that:

- Are currently being used by software engineering researchers.
- Or, are suitable for synthesising findings from software engineering primary studies.
- Or, are suitable for use by most researchers including relative novices. Note that we do not recommend any complete novice attempting qualitative meta-synthesis without having an expert mentor or supervisor.

Figure 10.1 shows the different qualitative methods discussed in this chapter identifying which are interpretive and which are aggregative. The double-lined boxes show the methods that are discussed in detail in this section. We

have identified thematic analysis as both an aggregative and an interpretive method. This depends whether the synthesis stops at second-level coding or produces a higher-level synthesis.

10.4.1 Meta-ethnography

Importance to Software Engineers: This form of synthesis is well-suited to primary studies based on ethnological research, which are likely to occur when researchers study team behaviour over an extended time period. We have found three examples of software engineering systematic reviews have used meta-ethnography: Dybå & Dingsøyr (2008*a*), Gu & Lago (2009), and Da Silva et al. (2013).

Definition: Meta-ethnography is a method for synthesising ethnographic studies, which Noblit & Hare (1988) define to be "long-term, intensive studies involving observation, interviewing, and document review".

Process: Noblit & Hare (1988) define a seven stage process involving:

1. Getting started, that is, defining what is of interest.
2. Deciding what studies are relevant to the topic of interest.
3. Reading the studies. This means detailed reading and re-reading of the relevant primary studies
4. Determining how the studies are related. This involves listing the key *metaphors*, which may be phrases, ideas and/or concepts, in each study. Then looking at how they relate to one another.
5. Translating the studies into one another. Noblit and Hare describe this as comparing metaphors and concepts in one primary study with those in another. They emphasize that translation maintains the central metaphors and/or concepts in each primary study "in their relation to other key metaphors or concepts" in the same study.
6. Synthesizing the translations. Translations may result in agreement among studies, contradictions among studies, or may form parts of a coherent argument.
7. Expressing the synthesis which means reporting the results of the synthesis to interested parties.

Example: Da Silva et al. (2013) present a detailed report of their use of meta-ethnography to analyse four primary studies that investigated personality and team working. In Step 1, they defined their research question as:

> *How does individual personality of team members relate with team processes in software development teams?*

In Step 2, they used a previous systematic review and its unpublished extension as the basis to identify five relevant primary studies. They applied an initial screening to check that the primary studies formed a "coherent set". They then applied the quality criteria used by Dybå & Dingsøyr (2008*a*) and excluded one low quality study. In Step 3, all team members read the papers. They note that the papers were also read and reread during subsequent phases. During this phase they also extracted:

- *Contextual* data about each primary study. They suggest such information should be defined *a priori* and extraction should be performed by at least two researchers, and disagreements should be identified and addressed. They reported their contextual information in a cross-case matrix with rows identifying the concepts (specifically: Objective, Sample, Research methods, Design, Data collection, Setting, Country) and columns identifying each of the four primary studies. In most cases the cells included appropriate quotes from the primary studies.

- *Relevant concepts* associated with the research question identified in each study. The information was presented in a cross-case matrix with columns identifying the studies and rows identifying the concepts (specifically: Task Characteristics, Personality, Conflict, Cohesions, Team Composition, Performance, Satisfaction, Software Quality).

In Step 3, they considered relationships between the different studies. They first considered which of the six concepts were addressed by at least two primary studies (to make synthesis possible). They then investigated the definition of the six relevant concepts and extracted the operational definitions used in each primary study to check whether the terms were used consistently. Finally, they investigated the relationships between the concepts. They considered pairs of concepts and sought findings from the primary studies that discussed the interaction between a pair of concepts. They reported, for each primary study, the interaction between each pair of concepts reported in the primary study with a specific textual quote if available. This was the main input to Step 5.

In Step 5, they translated the concepts and relations from one study to another. Specifically, they compared each pair of concepts across all studies to produce their first-order synthesis as input to Step 6. In Step 6 they produced a second order synthesis which aimed to produce a synthesis that was more than the sum of its part. This involved creating a diagram that summarized the synthesis and narrative that described the "central story" (like grounded theory). Step 7 was realised by their journal paper.

They comment that in their view:

- Meta-ethnography is not straightforward to use. It requires experience with the methodology and "the philosophical stances that form the cornerstones of interpretative research".

- It is not practical to synthesize too many studies since "it would be easy to forget the meanings of previously synthesized studies as the synthesis proceeds".

They also note that, although two of their four primary studies were ethnographical ones, two were quasi-experiments, so they were able to use meta-ethnography in a mixed methods setting.

10.4.2 Narrative synthesis

Importance for Software Engineers: Cruzes & Dybå (2011*b*) identified narrative synthesis as the most frequently used qualitative synthesis method by software engineering researchers.

Definition: Narrative synthesis reports the results of a systematic review in terms of text and words. Popay et al. (2006) refer to it as "a form of story telling". They point out that any qualitative meta-synthesis involves some narrative synthesis even when more specialised synthesis methods are also used.

Process: Popay et al. propose a narrative synthesis methodology that is targeted at systematic reviews that are concerned either with the effectiveness of some intervention or with factors that influence the implementation of interventions.[1] Their approach involves four main elements:

1. Developing a theory of how, why and for whom the intervention works. This activity is usually done at an early stage in the review and is intended to help formulate review questions and identify the appropriate primary studies. The model is also intended to help both interpreting the review findings and also assessing the generality of the findings.

2. Developing a *preliminary* synthesis of the findings of the primary studies. In the case of effectiveness studies, this involves assessing the direction and size of effects. It may also involve identifying the results of any quality appraisal of the primary studies. For implementation reviews, this is aimed at identifying facilitators and barriers to adoption.

3. Exploring relationships in the data. This aspect of synthesis goes beyond the preliminary synthesis to explore the relationships among studies, both between the characteristics of individual studies and their findings, and between the findings of different studies.

4. Assessing the robustness of the synthesis. Robustness refers to the quality and quantity of the primary studies, the information reported in the primary studies, and the methods used in the synthesis.

The basic process model described above seems appropriate for software

[1] Their report is available on request from j.popay@lancaster.ac.uk.

engineering reviews. In particular, the idea of starting by constructing a model of the innovation is particularly interesting. In our view, mapping studies would be of much more value if they aimed to produce a model of the intervention they discuss, organising the literature to illuminate various aspects of the model.

Popay et al. propose a mix-and-match approach to undertaking the various process steps, some based on general approaches such as "grouping and clustering studies" and "tabulation", others based on ideas from a number of different methodologies. For example, they propose "transforming the data into a common rubric" as a technique for developing a primary synthesis, which in their example involved constructing effect sizes that could equally have been used for quantitative meta-analysis. They also recommend reciprocal and refutational translation based on Noblit & Hare (1988) as a technique for exploring relationships among the data.

10.4.3 Qualitative cross-case analysis

Importance for Software Engineers: Cruzes & Dybå (2011*b*) classify qualitative cross-case analysis as the qualitative analysis methods proposed by Miles et al. (2014). Although many of their analysis methods are aimed at individual primary studies, rather than synthesizing across multiple qualitative studies, the analysis methods documented by Miles et al. can be used for cross-case reporting, analysis, and synthesis. The methods are based on graphical and tabular displays of textual information. The displays are described in great detail and provide an operational description for the tables many researchers use in practice. For example, the table, that Cruzes & Dybå (2011*b*) used to compare the synthesis methods claimed by secondary study authors with the synthesis methods they actually used, could be described as a *two-variable cross-case matrix*.

Definition: Miles et al. (2014) define a variety of tables and graphics to summarise data and report findings from qualitative studies, many of which apply to cross-case analysis, and so can be used for qualitative meta-synthesis.

Process: Miles et al. (2014) propose an analysis method based on four elements:

1. *Data collection* which in this case means finding and reading relevant primary studies.

2. *Data condensation* which is the process of "selecting, focussing, simplifying, abstracting and transforming data". They consider data condensation part of the analysis process since it involves coding and summarising the data.

3. *Data display* which is an "organized, compressed assembly of information that allows conclusion drawing and action". Like data condensation, data display is part of analysis since it involves organising the rows and

columns of matrices in order to reveal patterns in the data, or drawing diagrams that show the relationships among named entities.

4. *Drawing and verifying conclusions.* Drawing conclusions involves identifying patterns, explanations, cause-event relationships and propositions. It starts as soon as data collection begins. Verification means testing conclusions with respect to 'their plausibility, their sturdiness, their confirmability— that is, their validity'.

The individual elements in the model are *flows of activity* and are not meant to be sequential.

The various displays they describe form the basis for organising the data, analysing the data and presenting the final results. Many of the displays are based on matrices which they define to be the intersection of two lists set up as rows and columns and as a 'tabular format that collects and arranges data for easy viewing in one place.' They define *Meta-Matrices* to be master charts for assembling descriptive data from different cases (which would correspond to different primary studies in the context of qualitative meta-synthesis).

They describe a great many different types of matrices and meta-matrices, including:

- Partially-ordered meta-matrices that stack up data from different cases in one table that can be reformatted and re-sorted to look for cross-case trends.

- Predictor-outcome matrices that identify the main variables believed to affect the observed outcome. Such matrices are qualitative versions of the effect size versus moderator tables that might be produced during a quantitative meta-analysis.

Miles et al. also identify numerous methods for graphically displaying qualitative data and findings. These involve named entities often within boxes (or circles) linked by lines indicating the direction of a relationship among entities, such as the order of events in time, or the influence of one entity on others. These are particularly useful, since software engineering researchers are often quite familiar with such graphics from process modelling and software design methods. One less common style of graphic that might be of relevance to software engineering researchers interested in categorizing objects such as faults, code changes, process types is a *folk taxonomy.* Miles et al. describe nine types of semantic relationships that can be used (for example, inclusion—where X is a kind of Y, spatial—where X is a place in Y, cause-effect—where X is a cause of Y) and provide an example of how a taxonomy can be constructed.

Example: As part of a comparison of thematic synthesis, narrative synthesis and cross-case analysis(Cruzes et al. 2014) report an example of synthesizing two primary studies related to trust in outsourcing. The overall goal of the synthesis was to:

"Understand factors of trust in outsourcing relationships."

Runeson and Höst performed the cross-case analysis. They point out that the major part of data reduction was already conducted by the primary studies. Furthermore, they were only synthesizing two relatively homogeneous and condensed papers, so they "tagged data directly in printouts of the papers". They extracted data of two types:

1. Characteristics of the cases studies (specifically, goal, target population and culture, number of companies and interviews, maturity of companies, methodological framework, data collection, data analysis, and the definition of trust).

2. Factors and subfactors reported as being associated with trust together with the frequency with which they were mentioned.

Moving to the data display step, this information was initially displayed in two separate unordered cross-case data tables.

For the trust related information, further data reduction was performed to analyse the semantics of the identified factors. Runeson and Höst identified synonyms and homonyms based on the definitions used in each primary study. Based on those definitions, they rearranged the factors table into an ordered meta-matrix showing the unique and common factors identified in each study for establishing and maintaining trust, ordered by the frequency with which factors were mentioned (with a caveat that this is a doubtful practice, if wrongly interpreted).

Data synthesis involved identifying the relations between factors reported in each of the primary studies and expressing them in a graph showing the primary study that identified the relation, and whether it was related to establishing or maintaining trust.

Conclusions and verification involved preparing condensed summaries of the views found in paper to highlight the main results. They comment that they found no contradictions between the studies, although they put different emphasis on the factors.

10.4.4 Thematic analysis

Importance for Software Engineers: After narrative synthesis, thematic analysis is the next most frequently used method of qualitative synthesis adopted by software engineering researchers. It fits well with analysing software engineering studies that are aimed at assessing the benefits, risks, motivators and barriers to adopting new software engineering methods.

Definition: Thematic analysis involves identification and coding of the major or recurrent themes in the primary studies and summarising the results under these thematic headings.

Process: Cruzes & Dybå (2011*a*) define a five-stage process for thematic analysis involving:

1. Reading all the text related to all the primary studies.

2. Identifying specific segments of text relevant to the research questions or topics that seem common to several studies.

3. Labelling and coding the segments of text.

4. Analysing the codes to reduce overlaps and define themes. Some themes are likely to be defined in advance as a result of the research questions, while others may arise as a result of reading the primary studies.

5. Analysing themes to create higher-order themes or models of the phenomenon being studied. The graphical displays discussed by Miles et al. (2014) can be used to represent such models.

Cruzes & Dybå provide a detailed explanation of the process including examples taken from thematic syntheses produced by software engineering researchers and a checklist identifying good practice for each step.

Examples: Staples & Niazi (2008) provide a reasonably detailed description of their thematic analysis methodology. Their systematic review investigated reasons individuals gave for adopting CMM.

With respect to reading the papers (Step 1) only one of the researchers read all the papers. The same researcher identified quotes (that is, text from each study) related to adopting CMM in each study (Step 2). Both researchers then reviewed every quote independently and identified a list of higher level categories that described a unique reason for adoption. The reason comprised a short name and description (Step 3). They note that agreement was initially poor, but they were able to come to agreement via joint discussion and "in some cases a third researcher". Next (Step 4), they reviewed the reasons and grouped them into five higher-level categories Customers, People, Performance, Process, and Product.

Subsequent analysis was based on analysing the frequency with which reasons were mentioned in the identified studies. Thus, they did not undertake Step 5.

Cruzes et al. (2014) present an example of thematic synthesis to synthesis two papers investigating trust in outsourcing. After initially reading the papers and copying textual extracts into the *NVivo* system (Step 1), they used the *NVivo* tool to help both to identify segments of text containing references to factors related to trust (Step 2) and to label (that is, code) the text segments (Step 3). They reduced overlap between codes and identified seven themes that grouped codes together (Step 4). They, finally created a higher-level model (Step 5) with three higher order themes. The higher-level model was presented as concept maps showing the relationships between higher order themes, second-level themes and the original codes.

10.4.5 Meta-summary

Importance for Software Engineers: Although Cruzes & Dybå (2011*a*) did not find any software engineering systematic review that used

this method, it has properties that make it of relevance to software engineering problems. In particular:

- It is an aggregative method that may be easier for inexperienced researchers to understand than an interpretive method.
- It can be used to aggregate data from some types of qualitative and quantitative studies in the same meta-synthesis.
- It is appropriate for integrating findings from studies investigating barriers, motivators, risks and other factors associated with implementing a process innovation. In the context of software engineering research, there have been a large number of primary studies reporting the various problems found in globally distributed projects, and many secondary studies that have attempted to integrate the results of the primary studies (Verner et al. 2014). In our opinion, using this approach would have made the aggregation of primary studies much easier for the analysts to perform and for the readers to understand.

Definition: Metasummary is a quantitatively oriented aggregation method capable of integrating findings from *topical* surveys and *thematic* surveys (Sandelowski, Barroso & Voils 2007). *Topical surveys* are based on opinion-based questionnaires circulated to a relatively large number of participants. Analysis of topical survey data involves identifying the set of topics mentioned by the participants and counting how many participants mentioned each specific topic. This is usually done using content analysis and is essentially quantitative. *Thematic surveys* are typically based on researchers personally interviewing a relatively small number of participants. Analysis of thematic survey data involves looking for *latent patterns* in the interview data via first-order and second-order coding. Thematic analysis is more interpretative than content analysis but if it stops at identifying first-order codes, its findings still remain fairly closely related to the original data.

Furthermore, there is usually a disconnect between the methods that researchers claim to use and those they actually use. Sandelowski et al. (2007) suggest that the differences among methods are "typically honored more in the breach than in the observance". They point out that although qualitative surveys are meant to be "purposeful" and quantitative surveys are meant to be randomised, in the cases they investigated, most studies were actually convenience samples. Our experience with software engineering studies is consistent with their observations. Similarity between the findings and the actual methodology allows the metasummary method to aggregate results from both types of study.

Process: Metasummary is based on a five-step process:

1. Extract the findings from each study. Sandelowski et al. point out that findings in qualitative reports may be presented in other parts of the report rather than just in a separate results section. It is therefore necessary to separate relevant findings from other issues such as:

- Presentations of data, such as quotations or incidents.
- Reference to findings of other studies.
- Descriptions of analytic procedures, such as coding schemes.
- Discussion of the importance of findings.

2. Group topically similar findings together looking for equivalent findings.

3. Summarise and organise findings. Findings should be summarised using concise but comprehensive descriptions. They should be organised to show topical similarity (specifically, topics addressed by several studies) and thematic diversity (for example, favouring adherence or favouring non-adherence to a regime or process) and referenced to each primary study that mentioned the finding.

4. Calculate "effect sizes". Effect sizes are based on the number of *primary studies* that report a specific finding and *not* the number of participants mentioning the finding. This is consistent with the view that prevalence does not equate to importance. The *frequency effect size* for a specific finding is calculated as the proportion of *independent* studies that report specific finding compared with the total number of independent studies, that is:

$$FindingEffectSize = \frac{NumStudiesMentioningSpecificFinding}{TotalNumStudies}$$

The *intensity effect size* identifies which studies contributed most to findings. One intensity effect size metric is the proportion of findings with an effect size $> 25\%$ found in each study compared with the total number of findings with effect sizes $> 25\%$, that is:

$$StudyIntensityA = \frac{NumStudyLargeEffectSizeFindings}{TotalNumLargeEffectSizeFinding}$$

where NumStudyLargeEffectSizeFindings is the number of findings in a particular study that had an effect size$> 25\%$. A second intensity effect size metric is the proportion of findings found in a study compared with the total number of findings, that is:

$$StudyIntensityB = \frac{NumStudyFindings}{TotalNumFindings}$$

5. Report results. Findings can be displayed in summary matrices. An effects matrix would display each of the findings of each major type that is, favourable and unfavourable, indicating the effect size and the specific studies reporting the finding. A study influence matrix would identify the intensity effect sizes for each study, perhaps incorporating information about the nature of the study, for example, whether the study was

quantitative or qualitative, and summary information about participants such as nationality. An explanatory narrative is needed to describe the results and should discuss the impact of individual studies. In particular, studies that contribute little to the results, studies that contribute a great deal to the results, and studies that contribute many unique findings should be discussed to explain their relative contribution.

10.4.6 Vote counting

Importance for Software Engineers: Vote counting can be used in the context of quantitative systematic reviews when the variation among primary studies is too great for formal meta-analysis to be possible. A number of software engineering systemic reviews have reported results using variants of vote counting, see for example, Turner, Kitchenham, Brereton, Charters & Budgen (2010) and Kitchenham et al. (2007).

Definition: At its simplest, vote counting involves simply counting how many primary studies found a significant effect and how many did not. As discussed previously, simple vote counting has major methodological problems. However, it is more valuable when it is associated with a form of "qualitative" moderator analysis that investigates whether there are contextual or methodological factors that can help to explain differences in the outcomes of the primary studies using meta-matrix displays. We note that there is some difficulty in giving this form of analysis a name. Cruzes & Dybå classified several papers, that we would classify as "Vote counting", as examples of "Comparative Analysis" because they involved an investigation of possible moderating factors. If results are displayed in a tabular format, vote counting combined with moderator analysis is also a form of qualitative cross-case analysis (Miles et al. 2014).

Process: Like meta-analysis, vote counting assumes that a systematic review has identified a set of primary studies that each compare two software engineering interventions and it also requires that values of the outcome of the comparison, such as t-values, effect sizes or p-values, can be obtained from each primary study. Tabular displays are used to present the outcome values for each study which can be sorted or colour-coded according to which intervention was preferred. Popay et al. (2006) suggest a five-point scale to describe the outcome of the primary study:

1. Significantly favours intervention
2. Trends towards intervention
3. No difference
4. Trends towards control
5. Significantly favours control.

Additional moderating factors can be added to the displays to investigate whether there are any that appear to be associated with specific outcomes. Often, it is only possible to provide a narrative discussion of possible moderating factors. However, sometimes it may be possible to perform a more sophisticated synthesis. Cruzes & Dybå (2011*b*) suggest two such possibilities:

1. The *comparative method* (Ragin 1989) which uses Boolean truth tables to assess the combinations of moderators (modelled as boolean variables) that are associated with a successful or unsuccessful outcome of a case (for example, a primary study). The method assumes that there may be different combinations of factors that cause a particular outcome. It is able to cope with situations where some logical combinations of moderators do not exist among the set of cases. However, it appears to require that all important moderator variables are known. The technique is extremely complex but may resonate with researchers from computer science who are used to Boolean algebra and truth tables. The aim is to be able to say that a successful intervention occurs only when certain factors are present and other factors are not, using statements of the form "success occurs if and only if A OR (B AND NOT(C))=TRUE". Such statements imply that an underlying causal relationship is expected, rather than a statistical association exists among factors.

2. The *case survey method* (Yin & Heald 1975) uses standard statistical methods (for example, chi-squared tests, or logistic regression) to associate moderator values with binary or ordinal case outcome variables. The case survey method requires the availability of a large number of cases with the same moderator variables, which limits its applicability. The aim is to assess the frequency with which certain context factors are associated (or not) with a successful intervention and to provide a statistical assessment of whether the frequency is significantly different by chance.

Example: Kitchenham et al. (2007) reported a systematic review that compared the accuracy of cost estimation models built from data collection from a variety of different companies (cross-company models) with the accuracy of cost estimation models built from a specific company (within-company models). They grouped the primary studies into three groups: one for which the within-company models were significantly more accurate than the cross-company models, one for which there was no significant difference between the within-company and cross-company models, and one group of studies that were inconclusive (specifically, did not report any statistical analysis). They also produced a matrix display that identified the values of various study-related factors for each primary study, such as: the number of projects in the within-company dataset and the cross-company dataset, the size metric used, the type of model (linear or non-linear) derived from within and between

company data, and the size of projects in each dataset. They also constructed a summary matrix display of the factors that seemed to be associated with within-company models outperforming cross-company models and those that seemed associated with cross-company models performing as well as within-company models identifying which studies contributed to each conclusion.

10.5 General problems with qualitative meta-synthesis

This section discusses two problems that need to be considered in most qualitative meta-syntheses:

1. What to do about primary study quality.
2. How to validate the final meta-synthesis.

10.5.1 Primary study quality assessment

There appears to be no consensus among qualitative meta-synthesists about how to assess the quality of primary studies or, even whether quality should be assessed at all. For example, see Thomas & Harden (2008) and Spencer, Ritchie, Lewis & Dillon (2003). Even researchers who use quality evaluation, on the basis that they wish to avoid drawing conclusions on unreliable data, are unwilling to use quality criteria to exclude studies. For example, see Thomas & Harden (2008) and Atkins, Lewin, Smith, Engel, Fretheim & Volmink (2008).

Empirical evidence casts some doubts on the value of quality assessment checklists for qualitative primary studies. Both Hannes, Lockwood & Pearson (2010) and Dixon-Woods, Sutton, Shaw, Miller, Smith, Young, Bonas, Booth & Jones (2007) compared different quality checklists. Hannes et al. (2010) compared three different structured methods:

1. The Critical Appraisal Skills Programme (CASP) qualitative checklist[2] which is a very widely-used checklist that was the basis of a checklist used by Dybå & Dingsøyr (2008*a*) for their systematic review of agile methods.
2. A checklist compiled by the Australian Joanna Briggs Institute (2014)[3].
3. The Evaluation Tool for Qualitative Studies (ETQR) which was developed by the Health Care Practice Research and Development Unit from

[2](www.casp-uk.net)
[3]www.joannabriggs.org

the University of Salford, in collaboration with the Nuffield Institute and the University of Leeds.

Based on an analytical evaluation, they concluded that CASP was least able to evaluate certain aspects of validity.

Dixon–Woods et al. (2007) undertook a comparison of two structured checklists and a subjective evaluation. They found only slight agreement among the three methods and that the structured methods used which were CASP and a UK Cabinet Office quality framework (Spencer et al. 2003), did not show better agreement than expert judgement. Qualitative analysis indicated that reviewers found it difficult to decide between the potential impact of findings and the quality of the research or reporting practice. They also reported that structured instruments appeared to make reviewers more explicit about the reasons for their judgements.

In a qualitative study of researchers making decisions about the quality of studies for inclusion in a meta-ethnography, Toye, Seers, Allcock, Briggs, Carr, Andrews & Barker (2013) identified two issues of importance to reviewers: firstly, *conceptual clarity*, which relates to how clearly the author articulated an insightful issue, and secondly *interpretive rigour*, which relates to the extent to which the interpretation could be trusted. These two issues are clearly related to the impact of findings and the quality of research practice mentioned by Dixon–Woods et al. (2007).

It is, however, encouraging that both Thomas & Harden (2008) and Atkins et al. (2008) have commented that poor quality studies contributed less to their synthesis than better quality studies. Overall it seems that evaluating quality is mainly useful for sensitivity analysis, where the contribution of the individual studies can be compared with their quality. This is also consistent with the suggestion, in the context of metasummary, that the analysts should discuss the impact of individual studies on the overall results.

10.5.2 Validation of meta-syntheses

There are two aspects to validation of a meta-synthesis. Firstly the systematic reviewers, themselves, should ensure that they have "done a good job" and secondly, readers of the final systematic review report should find it trustworthy and useful.

Systematic reviewers need to reflect on the process they have used and identify any limitations of the process itself, or the way they used the process. Some of these reflections will be reported in the "Limitations" section of the final report, others may lead to additional synthesis activities such re-reading some excluded papers, or obtaining a second opinion on the plausibility of some of the reported findings

Readers of the final report of a qualitative meta-synthesis also need to be able to understand and to trust the findings. Qualitative systematic reviews should have similar properties to reports of qualitative primary studies mentioned by Toye et al. (2013) and Dixon–Woods et al. (2007), such as:

- Clearly reporting of insightful and valuable findings.
- Using a rigorous synthesis method.

For thematic analysis, Cruzes & Dybå (2011*a*) discuss trustworthiness of qualitative meta-synthesis in general, from the viewpoint of credibility, confirmability, dependability and transferability. They also provide a useful checklist, that researchers can use to assess the validity of their process at each stage in the thematic synthesis including the final stage of assessing the trustworthiness of the synthesis. Their checklist includes four questions about the trustworthiness of a synthesis:

1. 'Have the assumptions about, and the specific approach to, the thematic analysis been clearly explicated?'
2. 'Is there a good fit between what is claimed and what the evidence shows?'
3. 'Are the language and concepts used in the synthesis consistent?'
4. 'Are the research questions answered by the evidence of the thematic synthesis?'

We note that these questions seem applicable to any qualitative meta-synthesis not just thematic synthesis.