Sampling in Software Engineering Research: A Critical Review and Guidelines

Sebastian Baltes · Paul Ralph

Abstract Representative sampling appears rare in empirical software engineering research. Not all studies need representative samples, but a general lack of representative sampling undermines a scientific field. This article therefore reports a systematic review of the state of sampling in recent, high-quality software engineering research. The key findings are: (1) random sampling is rare; (2) sophisticated sampling strategies are very rare; (3) sampling, representativeness and randomness often appear misunderstood. These findings suggest that *software engineering research has a generalizability crisis.* To address these problems, this paper synthesizes existing knowledge of sampling into a succinct primer and proposes extensive guidelines for improving the conduct, presentation and evaluation of sampling in software engineering research. It is further recommended that while researchers should strive for more representative samples, disparaging non-probability sampling is generally capricious and particularly misguided for predominately qualitative research.

Keywords Sampling · Representative sampling · Random sampling · Purposive sampling · Probability sampling · Research methodology

1 Introduction

Most research involves selecting some of many possible items to study, i.e. *sampling.* Sampling is crucial not only for positivist research (e.g. questionnaires, experiments, positivist case studies) but also for interpretivist and con-

S. Baltes
University of Adelaide, Adelaide, Australia
E-mail: sebastian.baltes@adelaide.edu.au

P. Ralph
Dalhousie University, Halifax, Canada
E-mail: paulralph@dal.ca

structivist research (e.g. interview studies, grounded theory, interpretivist case studies).

In present-day epistemology, positivism refers to various ideological and philosophical positions usually associated with quantitative research, such as belief in objective reality and attempting to support a priori theories with empirical (usually quantitative) empirical evidence. Sampling is crucial for positivists because it determines external validity.

Meanwhile, social constructivism is the idea that much of our reality is socially or culturally constructed. This leads to interpretivism: the view that the methods of natural science are insufficient or inappropriate for understanding social reality *because* it is socially and culturally constructed. Interpretivists believe that each situation is unique and concepts from one context do not necessarily generalize to others. Both schools are associated with building (moreso than testing) context-dependent, middle-range theories. The goal of interpretivists and constructivists is not statistically generalizing from samples to populations—they sometimes do not even use the term "sampling". However, sampling is still crucial for them because selecting poor research sites hinders data collection and focusing on the wrong topics undermines theory building.

Sampling for software engineering (SE) research is particularly interesting and troublesome in at least three ways:

1. For many SE phenomena, there is no suitable *sampling frame*; that is, a list from which to draw a sample.
2. Some SE studies adopt poorly understood sampling strategies such as random sampling from a *non-representative surrogate population.*
3. Many SE articles evince deep misunderstandings of *representativeness*—the key criteria for assessing sampling in positivist research (see Section 2.6).

While some previous articles and books discuss sampling in the SE context (e.g. Shull et al., 2007; Foster, 2014; Baltes and Diehl, 2016; Nagappan et al., 2013) and sampling specifically for questionnaire studies (Kitchenham and Pfleeger, 2002; de Mello and Travassos, 2016), this paper goes into significantly more depth and discusses more SE-specific sampling problems. More specifically, the purpose of this paper is as follows:

> **Purpose:** *(1) Providing a detailed, SE-specific primer on sampling concerns and techniques; (2) investigating the state of sampling in SE research; (3) providing guidelines for improving sampling in SE research.*

2 Sampling: A Primer

Sampling is the process of selecting a smaller group of items to study (a *sample*) from a larger group of items of interest. The group of items of interest is called the *population*, and the (usually imperfect) population list is called the *sampling frame*. Researchers have developed diverse approaches for selecting

samples from populations. There is no single best technique or hierarchy of better and worse techniques. Different sampling approaches are appropriate for different purposes in different circumstances. While there are dozens of sampling strategies, we focuses on those most applicable to SE research.

The following exposition on sampling is primarily synthesized from several textbooks (Cochran, 2007; Henry, 1990; Trochim and Donnelly, 2001) and these concepts are widely accepted across the research methods literature. A good grasp of the sampling strategies and issues dissected in this section is necessary to understand the study that follows.

2.1 Non-probability sampling

Non-probability sampling includes all of the sampling techniques that do not employ randomness. This section discusses four kinds of non-probability sampling: convenience, purposive, referral-chain (or snowball) sampling, and respondent-driven sampling.

While some (positivist) methodologists (e.g. Hoeven et al., 2015) present non-probability sampling as intrinsically inferior to probability sampling, it really depends on the aims and context of the research (see Section 2.6). Each sampling strategy is appropriate in some circumstances and inappropriate in others.

2.1.1 Convenience sampling

In convenience sampling, items are selected based on *availability or expedience.* When we select people or artifacts to study arbitrarily, or based on them being nearby, available or otherwise easy to study, we adopt convenience sampling. Convenience sampling is controversial because it is very popular, especially in laboratory experiments, despite threatening generalizability (cf. Arnett, 2008; Henrich et al., 2010). The key advantages of convenience sampling are: (1) speed, (2) low cost and (3) no need for a sampling frame. These advantages make convenience sampling ideal for pilot studies and studying universal phenomena such as cognitive biases (Mohanani et al., in press).

2.1.2 Purposive sampling

In purposive sampling, items are selected based on specific characteristics, other than availability, that make them relevant to the study's objective. When we select our objects of study according to some logic or strategy, carefully but not randomly, we employ purposive or purposeful sampling (Patton, 2014).

Guidelines for purposive sampling are often provided in the context of selecting sites or data sources for predominately qualitative, interpretivist research (e.g. Miles et al., 2014). While non-probability samples can be representative of broader populations, the goal of non-probability sampling is often

to find accessible, information rich cases, sites, organizations or contexts from which researchers can learn about their topic of study (Patton, 2014).

Purposive sampling includes a whole range of approaches, for example:

1. Studying projects hosted on GitHub[1] because it is popular and has good tool support.
2. Recruiting a panel of experts on a particular topic for a focus group ("expert sampling").
3. Selecting projects that are as diverse[2] as possible ("heterogeneity sampling").
4. Extracting the sample from a repository using a specific search query, as in a systematic literature review (Kitchenham and Charters, 2007) ("query-based sampling").[3]

The key advantages of purposive sampling are: (1) the researcher can exercise expert judgment; (2) the researcher can ensure representativeness on a specific dimension (see Section 2.6); (3) no sampling frame is needed. The main challenge with purposive sampling is that it is intrinsically subjective and opportunistic.

2.1.3 Referral-chain (snowball) sampling

In referral-chain sampling (also called snowball sampling) sampling, items are selected based on their *relationship to previously selected items*. Snowball sampling is useful when there is no good sampling frame for the population of interest (Faugier and Sargeant, 1997). For example, there is no comprehensive list of black-hat hackers or software developers who have experienced sexual harassment. However, members of such "hidden populations" often know each other. Snowball sampling with human participants therefore works by finding a few individuals in the population, studying them, and then asking them to refer other members of the population whom they know.

In SE, snowball sampling is commonly used in systematic literature reviews to supplement query-based sampling. When we begin with an article A, searching the papers A cites is sometimes called *backward snowballing* while searching the papers that cite A is sometimes called *forward snowballing*. We can study software libraries, methods and services (as in service-oriented architectures), in much the same way.

The advantage of snowball sampling is that it helps us to identify items that are not in our sampling frame. However, snowball sampling has two major limitations: 1) it biases results toward more connected people (or artifacts); 2) it can lead to sampling a small, highly-interconnected subset of a larger population (see Figure 1).

---

[1] `https://github.com`

[2] Diversity can be defined along many different axes, gender being one of them (Vasilescu et al., 2015).

[3] Query-based sampling is limited by the scope and features of the search engine and the researcher's ability to construct a good query.
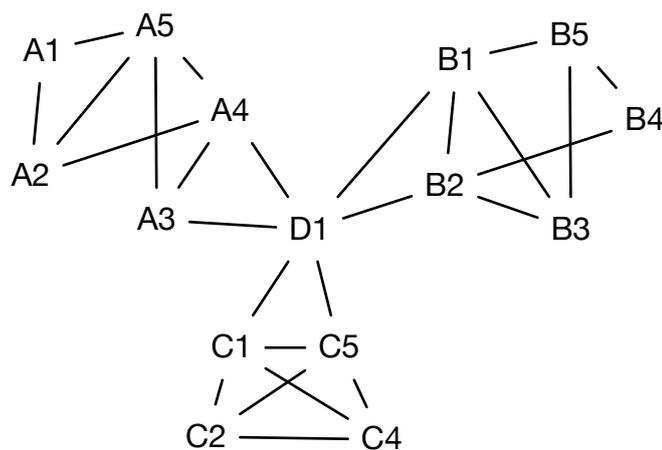
Fig. 1 Snowball sampling limitations: (i) item D1 is more likely to be selected because it has more connections and has higher network centrality; (ii) beginning with item B3, for example, the sample may be dominated by items from the B-cluster.

2.1.4 Respondent-driven sampling

Respondent-driven sampling is an advanced form of referral-chain sampling designed to mitigate sampling bias. Heckathorn (Heckathorn, 1997) proposed respondent-driven sampling to mitigate bias in referral chain sampling. Briefly, it works as follows.

1. Begin with diverse initial participants (seeds) who (i) have large social networks, (ii) represent different sub-populations, (iii) do not know each other, and (iv) can influence peers to participate.
2. Have participants recruit, rather than identify, peers. This reduces selection bias by the researcher.
3. Limit recruitment such that each participant can only recruit a small number of peers (typically three). This prevents highly-connected participants from biasing the sample.
4. Require many (e.g. 20) recruitment waves. This generates longer referral chains, decreasing the risk of oversampling from a highly connected subset of the population.
5. Prevent individuals from participating more than once.
6. Continue recruitment until the sample reaches *equilibrium*, the point where the distribution of variables of interest is stable.
7. Apply a mathematical model to account for sampling bias (Johnston and Sabin, 2010; Heckathorn, 1997).

While the details of the mathematical model used are beyond the scope of this paper, more information and tools are available.[4] Respondent-driven

---

[4] http://respondentdrivensampling.org/

sampling has, however, been criticized for producing optimistic confidence intervals, and generally being "substantially less accurate than generally acknowledged" (Goel and Salganik, 2010).

2.2 Probability sampling

Probability sampling includes all of the sampling techniques that employ randomness. In everyday language, *random* is often used to mean arbitrary or without logic. In sampling and statistics more generally, however, *random* means that each item in the population has an equal probability of selection (Daniel, 2011). Standing on a street corner interviewing "random" pedestrians is not random in the statistical sense. Recruiting participants using email or advertising on social networks is not random. Assigning participants to experimental conditions in the order in which they arrive at a laboratory is not random. Practically speaking, any selection without using a random number generator,[5] probably is not random.

Research on and guidelines for probability sampling are often written from the perspective of positivist questionnaire studies. These studies are typically descriptive (e.g. political polling) or explanatory (i.e. testing a theory based on perceptions of respondents). Since examining the entire population is usually impractical, the researcher selects a subset of the population (a sample) and attempts to estimate a property of the population by statistically analyzing the sample. Probability sampling ostensibly facilitates such statistical generalization (cf. Mullinix et al., 2015).

The overwhelming challenge for applying any kind of probability sampling in SE is the absence of comprehensive sampling frames for common units of analysis/observation (see Section 5). This section describes some probability sampling approaches that are relevant to SE research.

2.2.1 Whole frame

"Whole frame" simply means that *all* items in the sampling frame are selected. Suppose a researcher wants to assesses morale at a specific software development company. The company provides a complete list of developers and their contact information. The researcher creates a survey with questions about job satisfaction, views of the company, employees' future plans, etc. They send the questionnaire to all of the developers—the entire sampling frame. Whether this is technically "sampling" is debatable, but it is an important option to consider, especially when data collection and analysis are largely automated.

---

[5] True random number generation is available from numerous sources, including `https://www.random.org/`

2.2.2 Simple Random Sampling

Simple random sampling means that items are selected *entirely by chance, such that each item has equal chance of inclusion*. Now suppose the results of the above morale survey are less than spectacular. The researcher decides to follow up with some in-depth interviews. However, interviewing all 10,000 developers is clearly impractical, so the researcher assigns each developer a number between 1 and 10,000, uses a random number generator to select 20 numbers in the same range, and interviews those 20 developers. This is simple random sampling because the researcher simply chooses n random elements from the population.

2.2.3 Systematic Random Sampling

In systematic random sampling given an interval, x, *every x^{th} item is selected*, from a starting point chosen entirely by chance. To complement the interviews, the researcher decides to review developers' posts on the company's messaging system (e.g. Slack). Suppose there is no easy way to jump to a random message and there are too many messages to read them all. So the researcher generates a random number between 1 and 100 (say, 47) and then reads message 47, 147, 247, etc. until reaching the end of the messages. This is systematic random sampling. Each post still has an equal probability of inclusion; however, the consistent interval could bias the sample if there is a recurring pattern that coincides with the interval (e.g. taking annual weather data in the middle of summer vs. the middle of winter).

2.2.4 Panel sampling

Panel sampling is when the same sample is studied two or more times. Now suppose the researcher implements a program for improving morale, and a year later, re-interviews the same 20 developers to see if their attitudes have changed. This is called *panel sampling* because the same *panel* of developers is sampled multiple times. Panel sampling is probability sampling if the panel is selected randomly, and non-probability sampling otherwise.

2.2.5 A repository mining example of probability sampling

All four of these probability sampling strategies could also be applied in, for example, repository mining. We could (in principle) study every public project on GitHub using GHTorrent (Gousios, 2013) as our sampling frame (whole frame), randomly select 50 projects (simple random sampling), sort projects by starting date and study every 100th project (systematic random sampling), or take repeated measurements from the same 100 projects over time (panel sampling).

2.3 Multistage sampling

Methodologists often present multistage sampling as a special case where two or more sampling strategies are intentionally combined (e.g. Valliant et al., 2018). Two common approaches are stratified/quota sampling and cluster sampling.

2.3.1 Stratified/Quota sampling

In stratified and quota sampling, the sampling frame is divided into sub-frames with proportional representation. Suppose that the developer morale survey discussed above reveals significant differences between developers who identify as white and those who do not. However, further suppose that 90% of the developers are white. To get more insight into these differences, the researcher might divide developers into two *strata*—white and non-white—and select 10 developers from each strata. If the developers are selected randomly, this is called *stratified random sampling*. If the developers are selected purposively, it is called *quota sampling*. This sampling strategy is interesting because it is *intentionally* non-representative (Trost, 1986).

   We conceptualize these strategies as multistage because the researcher purposively chooses the strata (stage 1) before selecting the people or artifacts to study (stage 2). The systematic review reported in this paper (see Section 3) uses stratified random sampling.

2.3.2 Cluster sampling

Cluster sampling is when the sampling frame is divided into groups and items are drawn from a subset of groups. Suppose that the company from our morale survey example has 20 offices spread around the world. If the researcher wants to conduct face-to-face interviews, traveling to all 20 offices could be prohibitively expensive. Instead, the researcher selects three offices (stage 1) and then selects 7 participants in each of these offices (stage 2). This is called *cluster sampling*. If and only if both selections are random, it is *cluster random sampling*. Cluster sampling works best when the groups (clusters) are similar to each other but internally diverse on the dimensions of interest.

   Suppose that the researcher finds that the seven developers at one office seem much happier than developers in the rest of the company. If the researcher decides to conduct extra interviews at that office, in hopes of unraveling the sources of improved morale, this is called *adaptive cluster sampling* (Thompson, 1990; Turk and Borkowski, 2005).

2.4 Sampling in qualitative research

Qualitative researchers have to select both sites (e.g. teams, organizations, projects) and data sources (e.g. whom to interview, which documents to read,

which events to observe). Different qualitative research traditions (e.g. grounded theory, phenomenology, ethnography) talk about this "selection" in significantly different ways (Gentles et al., 2015). Some qualitative researchers use the term "sampling" (e.g. Glaser and Strauss, 2017). Others argue that "sampling" implies statistical generalization whereas qualitative research involves *analytical generalization* (Yin, 2018; van Manen, 2016). Others argue that there are many kinds of generalization, and qualitative researchers generalize from data to descriptions to concepts to theories, rather than from samples to populations (e.g. Lee and Baskerville, 2003).

This paper tries to clarify that sampling is about more than just statistical generalization. Predominately qualitative approaches including case studies, interview studies, grounded theory and action research typically use *non-probability sampling* to support *non-statistical generalization* from data to theory (cf. Stol et al., 2016; Checkland and Holwell, 1998; Ralph, 2019). Predominately quantitative studies, especially questionnaire surveys, *sometimes* use *probability sampling* to support *statistical generalization* from samples to populations. As we shall see below, however, many quantitative studies also adopt non-probability sampling.

Selecting sites and data sources is a kind of sampling. When a researcher selects a site because they consider it to possess an interesting property, that is purposive sampling. When a researcher interviews whoever will speak on a subject of interest, that is convenience sampling.

As fledgling concepts or theories begin to emerge, however, the researcher may use them to decide what to focus on next. In the grounded theory literature, *theoretical sampling* refers to selecting items to study based on an emerging theory (Draucker et al., 2007). For example, suppose the researcher from our running example begins generating a theory of developer morale, which includes a preliminary category "interpersonal conflict." The researcher might purposively sample peer code reviews with many back-and-forth interactions, because these reviews might contain evidence of interpersonal conflict.

2.5 Sampling Frameworks and Algorithms

Historically, sampling strategies were described using a three-tier framework: population, sampling frame, sample. For example:

 - Population: Households in City X
 - Sampling Frame: City X's phone book
 - Sample: 100 phone numbers randomly selected from City X's phone book

Many sampling approaches used in software engineering do not map into the population-frame-sample model. For instance, suppose that we are interested in non-code documents in software projects (e.g. specifications, lists of contributors, diagrams, budgets). Furthermore, we are especially interested in documents for open source systems and we have had good experiences mining GitHub, so we will limit ourselves to open source projects on GitHub. We only

speak English, so we exclude all non-English documents. Now we randomly select 50 English-language open source projects on GitHub and then use simple random sampling to choose up to ten documents from each selected project. We end up with 500 documents. Now suppose we also contact the owners of each selected project to ask if they object to the research, and suppose two of them do, so we delete the corresponding 20 documents.

It is not clear what *sample*, *sampling frame* and *population* refer to in the example above. Is the sample the 500 documents we collected or the 480 we retained? What if we asked permission before collecting the data and never collected the 20 stories from the two objecting projects? Is the sampling frame all GitHub projects or just all GitHub projects that have non-code documents, that are open source, that are in English or some combination thereof? Is the population all non-code documents in the world or is it limited to GitHub, English, open source projects or some combination thereof? Is there a "study population" (e.g. English software-related documents) that is narrower than a "theoretical population" (e.g. all software-related documents). What do we present as part of the population definition and what do we present as selection criteria? Which part is sampling and which is applying inclusion/exclusion criteria?

Furthermore, while simple random sampling was *involved*, it came after eliminating most software projects. Does claiming that a sample is good because we used probability sampling make any sense if we have previously excluded 99% of the population?

We suggest bypassing all this confusion by thinking of most quantitative studies as having a multistage sampling algorithm. Precisely defining what is the sampling frame and what is the sample is unnecessary; what matters is that the sampling procedures are clear enough to replicate and to understand how the sample might be biased. If the sample is retrieved using a script, simply provide the script in an appropriately archived replication package. Alternatively, if the sampling is performed manually, provide the precise, step-by-step algorithm used to determine the sample either in the main text (for shorter algorithms) or the replication package (for longer algorithms). In studies with many steps (e.g. systematic literature reviews) it may help to include a diagram that shows how many items were found or eliminated in each step (cf. Moher et al., 2009, Fig 1).

For some studies, sampling algorithms can be expressed in a single sentence such as 'we selected Corporation Y because of our pre-existing relationship' or 'we recruited a convenience sample of students from the authors' university using our faculty's student email list.' There is nothing wrong with having a simple sampling algorithm as long as it is appropriate for the goals of the study.

In contrast, the process by which qualitative researchers choose which data sources to focus on can be complex, intuitive and impossible to describe algorithmically. Rather, qualitative researchers should describe their approach and reasoning. Authors should consult existing work on theoretical sampling for guidance (e.g. Breckenridge and Jones, 2009; Draucker et al., 2007).

2.6 Representativeness

Kruskal and Mosteller (1979a) argue, with extensive examples, that the term "representative" has been (mis)used in at least five ways:

1. as a "seal of approval bestowed by the writer"
2. as the "absence of selective forces in the sampling"
3. as a "miniature or small replica of the population"
4. as a claim that its members are "typical of the population" or "the ideal case"
5. as a claim to heterogeneity or that all subpopulations or classes are included (not necessarily proportionately) in the sample

For our purposes, *representativeness* is the degree to which a sample's properties (of interest) resemble those of a target population. This section discusses common misunderstandings of representativeness and arguments for representativeness.

First, **representativeness is rooted in positivist epistemology**. Postmodernists, interpretivists, and constructivists reject the entire notion of statistical generalization on numerous grounds, including:

– Broadly applicable ("universal") theories of social phenomenon simply do not exist (Duignan, 2019).
– Each context is unique; therefore, findings from one do not translate wholesale into others (Guba and Lincoln, 1982).
– Statistical generalization precludes deep understanding of a particular social, political economic, cultural, technological context (Thomas and Myers, 2015).

Representativeness of a sample (or site) is therefore not a valid evaluation criterion under these epistemologies. Contrastingly, in positivism, falsificationism and Bayesian epistemology, the primary purpose of sampling is to support statistically generalizing findings from a sample to a population. Representativeness is therefore the overwhelming quality criterion for sampling: good samples are representative—bad samples are biased.

Second, **representativeness is widely conflated with randomness.** Suppose that two researchers, Kaladin and Shallan, have a sampling frame of 10,000 software projects, with an average size of 750,000 lines of code: 70% open source and 30% closed source. Kaladin randomly selects 10 projects. Suppose 4 of them are open source and 6 are closed source, with an average size of 747,823 lines of code. Meanwhile, Shallan inspects a histogram of project size and discovers a bi-modal distribution, with clear clusters of large and small projects. Shallan purposively selects 7 large open source projects, 7 small open source project, 3 large closed source projects and 3 small closed source projects.

This example illustrates several key points about representativeness:

1. Representativeness is dimension-specific. A sample can be representative on one parameter but not another.

2. Probability sampling does not guarantee representativeness on the dimensions of interest.
3. Non-random samples can be representative on the dimensions of interest.
4. Non-probability sampling can lead to more representative samples than probability sampling.
5. Random sampling depends on sample size.

If an unbiased, random sample is large enough, the law of large numbers dictates that its parameters will converge on the parameters of the sampling frame. What constitutes a "large" sample depends on many factors including the strength of the effects under investigation, the type of statistical tests being used and the number of variables involved. The parameters of small samples, however, may differ greatly from the sampling frame. While Bayesian approaches handle small samples more systematically and transparently, establishing representativeness with small samples remains problematic (McElreath, 2020).

Moreover, no sample size can overcome bias in the sampling frame. Suppose that Kaladin and Shallan want to draw inferences about all of the software projects conducted in Brazil. However, suppose the sampling frame is a list of all *public sector* projects in Brazil. Further suppose that public sector projects are more often large and open source than private sector projects, so Kaladin's sample is biased not only toward open source projects but also toward larger projects, and Shallan's sample is less representative than it first appeared.

Clearly then, *random* is not equal to *representative*. **Rather than defining representativeness, randomness is one of several possible arguments that a sample *should be* representative.** Extending Kruskal and Mosteller (1979a), we see seven main arguments regarding representativeness (see Table 1). None of these approaches guarantee a representative sample, but each has some merit.

As discussed above, we can argue that the sample is representative because individuals were selected randomly. However, random selection is only reliable if the sample size is large and the sampling frame is unbiased, and even then only produces a representative sample most of the time.

Alternatively, suppose we are surveying developers about their perceptions of the relationship between agile practices and morale. We can argue that the larger and broader our sample, the less likely it is to have missed an important subpopulation. The breath argument supports generalization of correlations (e.g. between agility and morale). Breath is the argument of heterogeneity sampling, and can apply to convenience and snowball sampling where oversampling some subpopulations is a key threat.

However, the breath argument does not support point estimates. Suppose only 1% of our sample reports abandoning agile practices because of morale. While the point estimate of 1% is not reliable, abandoning agile practices over morale issues is probably rare. It seems highly unlikely that a survey of 10,000 developers from 100 countries, including thousands of companies and dozens of industries, would miss a large subpopulation of low-morale agile-abandoners.

Table 1 Reasonable Arguments regarding Representativeness

Name	GPRS*	Argument	Threats
Random Selection	No	Each item in sampling frame has an equal probability of inclusion.	Assumes large sample size, good sampling frame and no coincidences.
Size	No	Sample is so large that missing a subpopulation is unlikely.	Bias toward one or more subpopulations or on important parameters.
Breadth	No	Sample captures a large variance on important parameters.	Point estimates unreliable; groups may not be proportional.
Parameter Matching	No	Sample and population parameters have similar distributions.	Possible bias outside of considered parameters; requires known population parameters.
Universality	No	All possible samples are representative because the phenomenon affects the entire population equally.	Phenomenon is not actually universal.
Postmodern Critique	No	The entire logic of statistically generalizing from a sample to a population is flawed.	Statistical generalization not supported.
Practical Critique	No	Generalizing to a population is not the purpose of this kind of study (e.g. case study, experiment).	Statistical generalization not supported.

*Guaranteed to Produce a Representative Sample?

Another reasonable argument for representativeness is that sample parameters mirror known distributions of population parameters. If we know the approximate distributions for a population of projects' size, age, number of contributors, etc., we can compare the sample parameters (e.g. using the chi-square goodness-of-fit test) to see if they differ significantly. If the sample parameters are close to known population parameters, the sample is representative on those dimensions. If the sample and population match on known dimensions, it seems more likely (but not guaranteed) that they will also match on unknown dimensions. This is the argument of quota sampling. Unfortunately, concerted efforts will be needed to establish population parameters for common units of analysis in software engineering research.

A quite different argument is the appeal to universality. Suppose we have good reasons to believe that the phenomenon of interest affects the entire population equally. For example, Fitt's law predicts the time required to point at a target based on the target's size and distance (Fitts, 1954). Insofar as Fitt's law is universal, researchers can argue that sampling is irrelevant—all samples are representative. The appeal to universality could apply to many laboratory studies and is related to debates about generalizing from student participants to professionals (cf. Sjøberg et al., 2002).

Contrastingly, some studies address sampling concerns by simply dismissing them on philosophical grounds (as described above). Others argue that,

practically speaking, statistical generalization is not the purpose of the present study (cf. Stol and Fitzgerald, 2018). There is nothing intrinsically wrong with this as long as the manuscript does not imply representativeness elsewhere.

Many studies, especially laboratory studies, simply ignore sampling concerns.[6] Sometimes a single sentence acknowledges the limitation that the results may not generalize (see also Section 4.2). Other studies give dubious arguments for representativeness (Kruskal and Mosteller, 1979b).

An ongoing discussion is centered around students vs. professionals as study participants (Feldt et al., 2018). Suppose researchers conduct an experiment based on a convenience sample of six American white, male, professional developers, who have bachelor's degrees in software engineering and are between the ages of 30 and 40. This sample is patently not representative of professional developers. Implying that is is representative because the participants are not students is plainly incorrect. Small convenience samples do not support statistical generalizability.

3 Systematic Review

We conducted a systematic review to investigate the state of sampling in software engineering. We manually retrieved and analyzed a collection of software engineering papers. This section describes the study's research questions, data collection and data analysis. The study design is based on common guidelines for systematic literature reviews and mapping studies (Kitchenham and Charters, 2007; Petersen et al., 2008). For this study, we adopt a positivist epistemology.

3.1 Objective and research questions

The objective of this study is to investigate the sampling techniques used in software engineering research, and their relationship to research methods and units of analysis/observation. This objective motivates the following research questions. In recent, high-quality software engineering research . . .

RQ1: . . . what *sampling approaches* are most common?
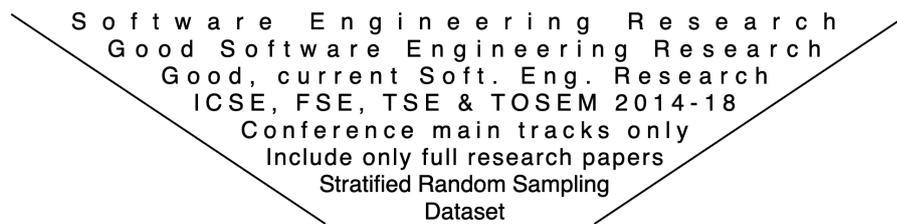RQ2: . . . how do authors *justify* their sampling approaches?
RQ3: . . . what *empirical research methodologies* are most common?
RQ4: . . . what *units of observation* are most common?

RQ1 is the main focus of the study. We include RQ2 to highlight the diversity of arguments for representativeness seen in the literature. Meanwhile RQs 3 and 4 are essential for informing guidance: which sampling approaches are appropriate strongly depends on the research method and unit of analysis/observation.

---

[6] Here, we do not cite specific examples because we do not wish to cast aspersions on individual researchers. While the statement is perhaps less credible this way, our credibility is not worth our colleagues' animosity.

S o f t w a r e   E n g i n e e r i n g   R e s e a r c h
G o o d   S o f t w a r e   E n g i n e e r i n g   R e s e a r c h
G o o d ,   c u r r e n t   S o f t .   E n g .   R e s e a r c h
I C S E ,   F S E ,   T S E   &   T O S E M   2 0 1 4 - 1 8
C o n f e r e n c e   m a i n   t r a c k s   o n l y
Include only full research papers
Stratified Random Sampling
Dataset

Fig. 2 Sampling strategy

3.2 Sampling strategy and inclusion/exclusion criteria

Figure 2 summarizes our sampling strategy. Systematic reviews typically use query-based sampling, followed by various techniques for addressing sampling bias such as reference snowballing (Kitchenham and Charters, 2007). Since most SE articles involve sampling, query-based sampling will retrieve all empirical SE studies, which is not very helpful. Obviously, we cannot manually analyze all empirical software engineering research papers (it took 15-30 minutes to code each paper with longer discussions of difficult cases; and there are hundreds of thousands of papers). Because of the time-intensive nature of the analysis, we considered a sample of 100 articles to be feasible to analyze. The question was, how best to select these 100 articles?

We wanted to know where the field is headed. This suggests focusing on recent papers in the most influential outlets. Consequently, we limit our sampling frame to articles published between 2014 and 2018 inclusive (we started discussing the systematic review in December 2018), in one of four outlets (flagship venues having an A* ranking according to CORE[7]):

1. The International Conference on Software Engineering (ICSE)
2. Foundations of Software Engineering (FSE), which was held jointly with European Software Engineering Conference in 2015, 2017 and 2018
3. IEEE Transactions on Software Engineering (TSE)
4. ACM Transactions on Software Engineering and Methodology (TOSEM)

Moreover, we applied several inclusion and exclusion criteria:

1. Include: Only full papers.
2. Exclude: Front matter, letter from editor, etc.
3. For FSE and ICSE: Include only papers in the main technical track (for symmetry).
4. For TSE and TOSEM: Exclude journal extensions of papers already in the sample.

We did not evaluate the quality of the articles because they were all accepted at top venues and we were interested in their sampling technique, not

---

[7] `https://www.core.edu.au/conference-portal`

their results. Applying these criteria produced a sampling frame of 1,565 full papers.

Next, we applied stratified random sampling, i.e. we randomly selected five papers from from each outlet-year (e.g. five papers published by TOSEM in 2016). This means that each outlet and each year have equal representation in our sample. We used a true-random number generator[8] to select papers from each outlet-year randomly, and selected them from the sampling frame we had previously built. The tool that we implemented to retrieve the papers and the R script implementing our sampling approach are available in our replication package (see Section 7).

Like many SE studies, we adopt a poorly understood sampling approach. First, we purposively selected "full research papers published in four good outlets over five years" and then randomly selected items to study from this more manageable list. We can make the randomness argument to representativeness; however, the four venues in the sampling frame are obviously not representative of all SE research because they are among the most competitive and are all in English. Other researchers might have chosen a different sampling frame. *There is no objective basis on which to select outlets or to study five years of four outlets vs. four years of five outlets vs. one year of 20 outlets.* To proceed, we must simply make reasonable choices and explain their implications.

3.3 Data extraction and analysis

Papers were randomly ordered and assigned a unique identifier from 1 to 100. Below, we use *PXX* to denote the *XX*th paper (e.g. P35 is the 35th paper in our sample).

The first author reviewed each paper and recorded the following data points in a spreadsheet: venue, year, title, authors, length (pages), relevant quotations (paper summary, sampling description, quality attributes, sampling limitations, population), number of studies reported in the paper, empirical method, number of samples, sampling stages, sample origin, units of observation, properties analyzed, study population, sampling frame, sampling frame size, sample size, number of items studied. The complete list of primary studies and dataset are included in the supplementary material (see Section 7).

Some papers clearly stated the sampling technique, for example:

1. "We invited 739 developers, via e-mail using convenience sampling" (P35).
2. "From the top 500 projects, we sampled 30 projects uniformly at random" (P43).
3. "We used stratified sampling to identify potential interviewees" (P56).

However, many papers did not clearly explain their sampling technique (specifics below). We therefore had to infer the sampling technique from the text. For example, we inferred purposive sampling from the statement: "We prepared

---

[8] https://www.random.org/

Table 2 Properties of studied papers ($n = 100$)

Property	Count					
	0	1	2	3	4	≥ 5
Studies per paper	4	73	17	5	0	1
Samples per study	0	91	24	5	2	1
Sampling stages per sample	0	112	11	0	0	0

... buggy code ... and asked workers to describe how to fix it. Only those who passed this qualifying test could proceed to our debugging tasks" (P92).

The first author marked ambiguous cases for review by the second author, and these were classified by consensus. For each ambiguous case, we developed a decision rule to guide future classifications (decision rules are included in our replication pack). The most important decision rule was to code studies where authors applied certain filtering criteria such as popularity of a project or experience of a developer as *purposive samples*. Another important rule was to use the code *sub-sample* to indicate when authors derived a new sample based on a previously introduced one (e.g. selecting developers who were active in projects that had been sampled before). A third rule was to classify studies based on dominant methodology. A study that was predominately quantitative with a small qualitative component, for instance, was coded as quantitative. For articles reporting multiple studies where some methods were primarily quantitative and others primarily qualitative, we captured both using a separate row for each study.

4 Results and Discussion

We analyzed 100 articles, of which 96 contained an empirical study. Some articles reported multiple studies; some studies had multiple stages of sampling (Table 2). We examined the results by *article* (100), *study* (127), and *sampling stage* (179). Note that multiple studies can use the same sample and multiple samples can be used by the same study.

The sample sizes ranged from 1 to 700,000 with a median of 20. In two cases it was not possible to derive the sample size from the descriptions in the paper. Only for 54 sampling stages it was possible to derive the size of the corresponding sampling frames. The sizes of the reported sampling frames ranged from 3 to 2,000,000 with a median of 447.5. This section addresses each research question and then comments on observable trends.

4.1 RQ1: Sampling techniques used

Table 3 shows the different sampling techniques used in the studies. The frequencies in Table 3 total more than 100 because some papers report multiple studies and some studies use multiple samples or sampling stages.

Table 3 Frequency of sampling techniques ($n = 179$ sampling stages)

Type	Strategy	Frequency
Non-probability	Purposive	125 (69.8%)
Non-probability	Convenience	20 (11.2%)
Probability	Simple random	12 (6.7%)
Other	Whole sampling frame	9 (5.0%)
No sampling	No empirical study or generated data	9 (5.0%)
Probability	Stratified random	3 (1.7%)
Non-probability	Snowballing	1 (0.6%)

Table 4 Origins of samples in SE research ($n = 179$ sampling stages)

Strategy	Definition	Frequency
Unclear	The strategy was not explained	51
Existing-sample(s)	Study used previously collected data, e.g., reported in related work	49
Sub-sample	Study used a subset of another sample presented in the same paper	21
Online-resources(s)	Sample retrieved from the internet (e.g., websites, mailing lists, LinkedIn)	20
Personal-network	Sample comprises artifacts or people that researchers had access to (e.g., students, industry contacts)	15
Other	For example public or corporate datasets, snowballing.	23
Gen-data	Study used generated data	5
No-study	Paper did not report an empirical study	4

The most common strategies were purposive (125) and convenience sampling (20). Only 15 stages utilized probability sampling—12 used simple random sampling and 3 used stratified random sampling. Nine stages analyzed their entire sampling frame. Nine stages did not analyze empirical data.

Half of the sampling stages that involved simple random sampling were sub-samples of previously derived non-random samples. The same applies to all three sampling stages utilizing stratified random sampling. This is an important observation because random sampling from a non-random frame undermines the argument that the sample should be representative because it is random. See the coding of papers P16, P27, P35, P43, P47, P48 and P82 in the supplementary material for more details.

Samples are derived from a variety of sources (see Table 4). In 49 cases (27%), the sample was based on an existing dataset, typically from related work. In 21 sampling stages, a subset of a larger sample presented in the corresponding paper was used. Twenty stages involved sampling from online resources, which was usually done in a purposive manner (17 out of 20 stages). Sources for such samples included the Android developer documentation (P16), online Scala tutorials (P45), LinkedIn groups (P49), app stores (P80, P89, P98), or the online Common Vulnerabilities and Exposures database (P91). Fifteen stages involved sampling from the researchers' per-

Table 5 Empirical method vs. sampling strategy ($n = 175$ sampling stages)

	None	C*	P*	S*	Other
Experimental tool evaluation (quant.)	11	7	90	5	2
Mining software repositories (quant.)	0	1	8	1	0
Task- & questionnaire user study (quant.)	2	4	3	0	0
Questionnaire survey (quant.)	0	2	5	0	1
Taxonomy building (qual.)	0	0	2	5	0
Coding of artifacts (qual.)	1	0	4	1	0
Other	0	6	13	0	1

*Note: C = Convenience, P = Purposive, S = Simple Random

sonal networks (e.g. their students or colleagues) and five stages used generated data. In 51 cases (29%), the origin of the samples was unclear.

4.2 RQ2: Authors' justifications

Of the 100 articles in our sample, 74 provided some justification as to whether their sample exhibits certain quality criteria, often despite a questionable— or unexplained—sampling approach. The justifications were often mentioned in "Threats to Validity" or "Limitations" sections. The most common justification (22 articles) was that the studied artifacts were "real" (as in "real-world"). P17, for instance, described the sample as containing "representative real-world ORM applications", but did not go into details about the actual strategy followed to select those applications. Further popular adjectives were "representative" (13), "large" (13), and "diverse" (7).

4.3 RQ3: Research methodologies

Predominately quantitative studies outnumber predominately qualitative studies 100 to 13. Of those 100 quantitative studies, 77 report experimental evaluations of software tools, models, or machine learning approaches. Eight studies involve mining software repositories; eight studies report on different kinds of user studies, and five are questionnaire-based surveys. Beyond that, the diversity of approaches defies organization into common methodological categories. For instance, one study involves comparing software metrics; others build taxonomies. Table 5 contrasts empirical methods and the utilized sampling strategies.

4.4 RQ4: Units of observation

We organized the primary studies' units of observation into the categories shown in Table 6. Most of the studies investigate *code artifacts* including GitHub projects (e.g. P4, P17, P43), code commits (e.g. P16, P85), and packages (e.g. P56). Examples for *other artifacts* include bug reports (P16), faulty

Table 6 Units of observation ($n = 179$ sampling stages)

Unit	Examples	Frequency
Code artifacts	Projects, source code, defects, fixes, commits, code smells	115
Non-code artifacts	Bug reports, discussions, pull requests, effort estimates, feature models	27
People	Developers, maintainers, students, interview transcripts	26
Articles	Papers published in SE journals and conferences	2
Generated data	Generated automata, simulated concurrent accesses, generated Java-projects	5
No study	Formal descriptions, proofs	4

Table 7 Empirical method vs. unit of observation ($n = 175$ sampling stages)

	Software	Artifacts*	People	Other
Experimental tool evaluation (quant.)	88	20	2	5
Mining software repositories (quant.)	8	2	0	0
Task- & questionnaire user study (quant.)	2	0	7	0
Questionnaire survey (quant.)	3	0	5	0
Taxonomy building (qual.)	3	4	0	0
Coding of artifacts (qual.)	5	1	0	0
Other	6	0	12	2

*non-code artifacts

rules in model transformations (P29), and test logs (P99). Besides students (e.g. P69, P94, P96), the category *people* also includes GitHub users (P77), Microsoft developers (P87), and Amazon Mechanical Turk workers (P94). Table 7 contrasts units of observation and the empirical method of the study.

4.5 Discussion

Perhaps the most salient finding of this study is that **purposive and convenience sampling were the most commonly employed strategies** in both qualitative and quantitative studies. Only ten articles employed probability sampling and 9 out of the 15 corresponding sampling stages were sub-samples of non-probability samples. While non-probability sampling can sometimes produce representative samples, the broad absence of probability sampling suggests that **software engineering research has a generalizability crisis**.

While this kind of study cannot determine why probability sampling is so rare, at least three factors contribute to this phenomenon:

1. Probability sampling is intrinsically difficult for some methodologies (e.g. experiments with human participants). However, non-probability sampling is popular even for questionnaires and quantitative data analysis studies.

2. Some SE research adopts interpretivism or other philosophical positions incommensurate with statistical generalization. Although we did not analyze philosophical positions (not least because most articles do not state one), a few studies may fall into this group.
3. Good sampling frames for most SE phenomena do not exist.

The significance of this third factor cannot be overstated. Without unbiased sampling frames, the randomness argument for representativeness falls apart. We can only claim that the sample represents the sampling frame, but we do not know how the sampling frame differs from the population.

Sampling frames are usually incomplete. Telephone sampling, for example, is incomplete because not everyone has a phone, but *most* households have one or more phones and there are techniques to account for unlisted numbers (Landon Jr. and Banks, 1977).

There is nothing like a phone book of all the software developers, projects, products, companies, test suites, embedded systems, design diagrams, user stories, personas, code faults, or code comments in the world or even in a specific country, language or application domain. Instead, we study samples of GitHub projects (e.g. P10), Microsoft developers (e.g. P87) or Huawei test logs (e.g. P99). If we randomly select enough Microsoft developers, we might get a sample representative of all Microsoft developers, but this is obviously not representative of all developers in the world because even if there were such a thing as an average company, Microsoft would not be it.

The closest we can get to publicly available sampling frames of certain sub-populations of software developers is probably the list of registered Stack Overflow users provided by their official data dump or SOTorrent (Baltes et al., 2018) and the lists of GitHub projects and users provided by the GHTorrent dataset (Gousios, 2013). Those datasets, however, both come with their own challenges and limitations (Baltes and Diehl, 2016).

This raises two questions: how do we get better samples for our research and how should we evaluate sampling when reviewing? We will return to these questions in Section 5.

Meanwhile, there seems to be widespread confusion regarding sampling techniques and terminology. The frequency of articles not explaining where their samples came from is concerning. We cannot evaluate sampling bias without knowing the source of the sample. Beyond that, we see the following archetypal problems in the rhetoric around sampling.[9]

- Incorrectly using the term "random" to mean arbitrary.
- Arguing that a convenience sample of software projects is representative because they are "real-world" projects.
- Assuming that a small random sample is representative because it is random.

---

[9] Here, again, we avoid direct quotations because we do not wish to cast aspersions on individual researchers. While our argument is perhaps less credible this way, our credibility is not worth our colleagues' animosity. Anyone who reviews much software engineering research will likely have seen at least some of these attitudes.

- Assuming that a large random sample is representative despite being selected from a potentially or obviously biased sampling frame.
- Implying that results should generalize to large populations without any claim to having a representative sample.
- Dismissing research (particularly case studies) over representativeness concerns when representativeness is not a goal of the study.
- Implying that qualitative research is inferior to quantitative research because of the prevalence of non-probability sampling, as if all quantitative research used representative samples.

Despite the problems described above, there are studies describing their sampling procedures well. To end our discussion on a positive note, we want to highlight positive examples from recent software engineering papers:

- Some experiments with students present a brief account of how a convenience sample of students was recruited at a certain university, from a certain class or cohort (e.g. Mohanani et al., 2019).
- Some case studies mention purposive sampling and state some characteristics of the sites that led to their selection (e.g. Ingram and Drachen, 2020).
- Some repository mining studies clearly explain that they randomly draw elements from a purposively chosen sampling frame, acknowledging that their results should statistically generalize to the sampling frame but not the theoretical population (e.g. Maalej and Robillard, 2013).
- Other repository mining studies clearly explain sampling was constrained by API limitations, and how the researchers worked around those constraints (e.g. Humbatova et al., 2020).
- Some grounded theory studies effectively explain that their goals do not include either representativeness or generalizing from a sample to a population (e.g. Sedano et al., 2019).
- Some questionnaire surveys thoroughly explain their sampling strategy and its implications (e.g. Russo and Stol, in press).
- Some studies conducting query-based sampling provide a detailed account of how the queries were generated (e.g. Cummaudo et al., 2020).
- Some systematic literature reviews present an extensive step-by-step account of how the primary studies were retrieved and filtered (e.g. Beecham et al., 2008).

4.6 Threats to Validity

The preceding results should be interpreted in light of several limitations:

Reliability. Most of the data we extracted from the articles comprises objective, unambiguous facts such as venue, year, title, authors, length (in pages), sample size, and unit of observation. We therefore did not employ multiple independent judges. However, in some articles, some data points (e.g. the methodology or sampling approach) were ambiguous. We tried to mitigate

this threat to reliability by having the second author audit ambiguous cases; however, different researchers may have reached different conclusions.

Replicability. To enhance replicability, we provide a comprehensive replication pack (see Section 7) including our data collection scripts, sample and analysis documents. However, interpreting the primary studies is somewhat subjective—especially when details are sparse—and different researchers may come to different conclusions.

External validity. Our sample should statistically generalize to our sampling frame, that is, to the past five years of full, empirical papers in ICSE, FSE, TSE and TOSEM. Arguably, our results should also transfer, somewhat less accurately, to other similar venues and to research published before or since our window. It is not safe to assume that our results generalize to substantially different venues, research conducted decades ago, or research that will be conducted years in the future.

Construct and measurement validity. This research does not involve any latent variables, so construct validity does not apply per se. However, the reader could question whether we appropriately ascertain the primary studies' empirical methods, sampling approaches and so on. This kind of study does not lend itself to quantitative analysis of measurement validity. All we can say is that the text of this paper ought to demonstrate that we have sufficient training and knowledge of sampling to be credible.

Internal and conclusion validity. We do not test causal hypotheses so internal validity does not apply. Our analysis is predominately descriptive, so conclusion validity is strong—there are no complicated statistical tests here to go awry. There are no strong assumptions (e.g. homoscedasticity) to be violated.

5 Recommendations

5.1 From problem to recommendations

Section 5 demonstrates that software engineering has a generalizability crisis brought on chiefly by a lack of attention to representative sampling. The following section presents numerous recommendations and guidelines for addressing this crisis. However, before explaining the guidelines, we need to clarify their epistemological status.

The literature review presented above demonstrates that sampling guidance is needed. However, the literature review does not, and cannot, *support* specific guidance. Methodological guidance and other meta-scientific claims typically cannot be justified empirically. Important methodological works in software engineering merely explain how research should be conducted or summarize works from reference disciplines that explained how research should conducted (e.g. Kitchenham and Charters, 2007; Stol et al., 2016; Ralph, 2019; Wohlin et al., 2012; Kitchenham et al., 1995; Runeson and Höst, 2009; Juristo and Moreno, 2013; Staron, 2020). Such guidance is not based on an empirical

study because it is not practical to test guidelines empirically. For example, we recommend that manuscripts explain how the sample(s) were collected. Explaining where a study gets its data is not controversial; however, it is not clear how to design an experiment that would support this recommendation. More to the point, such studies are rarely if ever conducted.

Rather, it is up to first reviewers and then readers to examine methodological guidelines and think critically about their face validity, content validity, comprehensiveness, usefulness, consistency and credibility. Further, we do use the guidelines to assess our own approach (see Section 5.4).

5.2 Guidelines for researchers

To improve the conduct and reporting of sampling for any empirical study, we recommend:

1. **Clarify your philosophical position.** A treatise on twenty-first century epistemology is not necessary, but one sentence on the study's perspective—positivism, falsificationism, interpretivism, constructivism, critical realism, etc.—would help. If the reader (or reviewer) has to guess, they might guess wrong, and apply inappropriate criteria.
2. **Explain the purpose of sampling.** Clearly state whether you are aiming for a representative sample, or have a different goal (e.g. finding interesting examples).
3. **Explain how your sample was selected.** For qualitative studies, the reader should be able to *recover* your reasoning about what to study (Checkland and Holwell, 1998). For quantitative studies, the reader should be able to replicate your sampling approach. The size of the sample should be evident.
4. **Make sure your sampling strategy matches your goal, epistemology, and type of study.** For example, a positivist questionnaire might use respondent-driven sampling; a pilot laboratory experiment might use a convenience sample, and an interpretivist case study might employ purposive sampling.
5. **Avoid defensiveness.** Very few software engineering studies have a strong claim to representative sampling. Overselling the representativeness of your sample is unnecessary and unscientific. For example:
   - Do not misrepresent ad hoc sampling as random.
   - Do not pretend small samples are automatically representative because they are random or because they were purposefully selected.
   - Do not ignore the potential differences between sampling frames and populations.
   - Do not pretend a sample is representative because it is "real" (e.g. professionals instead of students, real projects instead of toy examples).
   - Do not admit to sampling bias in your limitations section only to pretend the results are near-universal in your conclusion.

Moreover, if representativeness is the goal of the study, we further recommend:

1. **State the theoretical population;** that is, in principle, who or what you would like to generalize to (e.g. professional software developers in Brazil, code faults in cyberphysical systems).

2. **Present your sampling algorithm** (see Section 2.5). Try not to worry about the population versus the sampling frame versus the inclusion/exclusion criteria. Just provide a replicable, concise, algorithmic account of how another person can generate the same sample. If you use data collection scripts, provide them. If the sampling strategy has many phases, consider a diagram or flow chart. If your sampling approach was nondeterministic (e.g. theoretical sampling), describe your reasoning.

3. Give an **explicit argument for representativeness** (cf. Table 1). Admit the generalizability threats implied by this argument.

4. **Clearly explain how the sample could be biased.** For complicated sampling strategies, discuss bias for each step in your algorithm or diagram explaining the sampling approach. This could be presented in the sampling section or with limitations.

5. **Publish your sample** as part of a replication package **if and only if** it does not contain sensitive or protected information. Be very careful of the potential for de-identified data to be re-identified in the future.

For example, if we are conducting a randomized controlled experiment with a convenience sample of software developers, we should not claim that the sample is representative because the participants are professionals. We should explain that we are prioritizing internal validity and that this kind of research does not support statistical generalization. Similarly, for a case study, we should explain that the purpose of a case study is to understand one site deeply, not to generalize from a sample to a population. We should not pretend that one or a few sites are representative because they are large companies or real projects or use popular practices. For interpretivist research, meanwhile, we should discuss transferability rather than external validity.

The situation is trickier for questionnaire surveys, qualitative surveys, repository mining and exploratory data science, where external validity is more often a key criterion. Authors of these studies should offer an explicit argument for representativeness and clearly explain attempts to mitigate sampling bias.

5.3 Mitigating sampling bias in different kinds of studies

Several other techniques can help improve sampling in certain situations. For large samples, we can use bootstrapping to assess stability. For instance, if we have a convenience sample of 10,000 Java classes, we can randomly exclude 1,000 classes and check whether it perturbs the results. If a sampling frame is biased, consider replicating the study using different sampling frames. For

example, if we find a pattern in a sample of GitHub projects, replicate the study on a sample of SourceForge projects. The more diverse the repositories, the more likely the results generalize. This could be a full replication or a limited sanity check with a small sample from a different domain. For studies of software projects, consider using sample coverage; that is "the percentage of projects in a population that are similar to the given sample" (Nagappan et al., 2013), to support heterogeneity sampling.

Developers (or developers with certain experiences) can be treated as a hidden population. If respondent-driven sampling (Section 2.1.4) can reach groups such as injecting-drug users and sex workers (Malekinejad et al., 2008), surely it can help reach software developers. Meanwhile, many practices can reduce sampling bias and response bias in **questionnaire surveys** (Dillman et al., 2014). These include starting with important questions that resonate with participants (not demographics), avoiding matrix questions, avoiding mandatory questions, and sending reminders. Offering incentives (e.g. cash, prizes) is also effective. Some research suggests that offering charitable donations does not increase response rates (Toepoel, 2012); however, none of this research was done with software developers, and donating to an open source project might be more effective than small cash incentives for motivating open source contributors to complete a questionnaire (e.g. Ralph et al., 2020). There are also myriad techniques for *assessing* response bias (Sax et al., 2003). Many questionnaire studies in SE use none of these techniques.[10]

Similarly, sampling bias and publication bias in **systematic reviews** can be addressed by (i) forward and backward snowballing on references, (ii) searching multiple databases, (iii) searching pre-print servers and dissertations, and (iv) requesting unpublished work in an area through relevant mailing lists, and (v) checking websites of prolific researchers in the area.

Evaluating sampling bias in **software repository mining** is fraught. Each repository is likely biased in unpredictable, non-obvious ways. Therefore, we cannot safely assume that random samples from one repository are representative of other repositories or software in general. Comparing samples from multiple repositories may help improve representativeness. Research comparing projects stored in public and private repositories is especially needed, but intrinsically difficult. The private code that companies are willing to share might systematically differ from the code they will not share (Paulson et al., 2004). One (ethically challenging) option is to examine private code that has been leaked.

In the long term, SE research needs better sampling frames. One way to achieve this is to develop curated corpora like the Qualitas corpus, "a large curated collection of open source Java systems" (Tempero et al., 2010). Similar corpora could be developed for many kinds of code and non-code artifacts used in software projects, including design specifications, requirements specifications, diverse models and diagrams, user stories, scenarios, personas, test

---

[10] Again, we do not cite specific examples here for fear that it would do more harm than good.

cases, closed-source Java systems, systems in all the other common languages, unit tests, and end-user documentation. Creating any one of these corpora is a major undertaking and should be recognized as a significant research contribution in itself. Even without good demographic information, the representativeness of a curated corpus can be improved in numerous ways:

1. Including artifacts from diverse domains (e.g. aerospace, finance, personal computing, robotics).
2. Including artifacts from diverse software (e.g. embedded systems, enterprise systems, console video games).
3. Making the corpus large enough to support heterogeneity sampling and bootstrapping.
4. Attempting to match the parameters we can discern; for example, we could attempt to include artifacts from different countries according to the size of each country's software industry.

Corpora improve reproducibility because one corpus can support many studies. Furthermore, building corpora helps to separate the difficult task of creating and validating a good sampling frame from any particular study of the items in the corpora. This makes research more manageable.

That said, great care is needed to build unbiased corpora, or at least corpora with well-understood biases. Corpora can quickly go out of date and there exists an innate tension between keeping corpora stable and up-to-date.

5.4 Did we follow our own guidelines?

We clearly stated our philosophical position (positivism, at least for the literature review; the primer is more pragmatic), our purpose (representativeness with some caveats), and our sampling strategy (see Section 3). Between our description and supplementary materials, an independent researcher could replicate exactly what we did. We explained why stratified random sampling was appropriate under these circumstances, and described at length how our approach could bias the sample. We enumerated the study's limitations (in Section 4.6) and explained that our theoretical population is leading, recent SE research. Our argument to representativeness is randomness, but we acknowledge bias in the sampling frame. We published our sample (see Section 7).

We studied four venues, which is similar to having four sampling frames. We did not use bootstrapping (because our sample is not large enough), techniques for hidden populations (because published research is not a hidden population), or sample coverage (because we were not using heterogeneity sampling). We did not attempt to mitigate publication bias because it seems highly unlikely that unpublished work has significantly better sampling strategies than published work.

5.5 Guidelines for reviewers

From our experience, many reviewers struggle to evaluate sampling. Our advice is to **evaluate sampling in the context of a study's philosophy, methodology, goals and practical realities**. In-principle, context-free assessments of external validity or transferability are unhelpful.

For anti-positivist studies, it is sufficient for researchers to justify site selection and explain their data collection. Complaining about low external validity in a case study is typically unreasonable because that is not what a case study is for (Stol and Fitzgerald, 2018).

When reviewing a positivist study that does not aim for generalization (e.g. a laboratory experiment with human participants) only worry about high-level external validity threats such as using student participants instead of professionals (Feldt et al., 2018; Falessi et al., 2018). Complaining about low external validity in a laboratory experiment is typically unreasonable because that is not what a lab study is for (Stol and Fitzgerald, 2018).

However, when reviewing a study that does aim for generalization (e.g. a questionnaire study) insist on reasonable attempts to mitigate sampling bias. The whole point of a large questionnaire survey is to sacrifice internal validity for external validity. If external validity is the main priority of a study, it should have a defensible claim that its sample is representative.

For example, suppose we are evaluating a questionnaire study of 3D animators at AAA game companies. The authors recruited animators by posting on mailing lists, which is basically convenience sampling—end of sampling discussion. This should be rejected not because it uses convenience sampling but because appropriate, practical steps for mitigating sampling bias were not taken. Authors should have used respondent-driven sampling, or found a list of AAA game companies and used stratified random sampling, or advertised on multiple channels and compared them (cf. (Baltes and Diehl, 2016)). They should have reported response rates or bounce rates, compared early responders to late responders and so on.

In contrast, suppose we are evaluating a constructivist grounded theory study of agile practices at a Norwegian software company. We could say "this study has low external validity because we cannot generalize from an n of 1." This is simultaneously true and inappropriate. External validity is not an appropriate quality criterion for this kind of study (Charmaz, 2014) and statistical generalizing is not its aim. Instead, we should be asking why this site was selected, how the researchers went about theoretical sampling, and to what extent the resulting theory seems transferable to other contexts.

Assessing sampling in software repository mining is difficult because it comes from SE, so we are creating the norms. What we can say with confidence is that reviewers should question the assumption that a sample is representative because it was randomly selected from a repository, unless evidence is provided that the software in the repository does not differ from software in general on the dimensions of interest.

The key is to evaluate studies against the norms for that particular kind of study, and to question whether researchers applied practical mitigations available to them (like sending reminders for a questionnaire), not to evaluate abstract quality criteria like external validity. Pilot and proof of concept studies investigate something under ideal—not representative—conditions. For experiments with human participants, representative sampling is often prohibitively expensive. Qualitative research does not seek to generalize to other contexts, so representative sampling is irrelevant, and disparaging "a sample of 1" is merely prejudice against qualitative research. For studies that do not aim for representativeness, reviewers should instead focus on over-generalization. Lab studies and pilot studies under ideal conditions do not show that something works in real life; qualitative field studies do not establish universality.

Reviewers should check whether the sampling strategy is commensurate with the study's implications. Non-representative sampling should be accompanied by acknowledging that external validity is limited. Such acknowledgments should not be followed by a sneaky implication that the results are universal. Misusing the term "random" should not be tolerated.

Finally, reviewers should consider whether the sample is large enough. For studies using frequentist statistics to test causal hypotheses, power analysis should be used to estimate desired sample sizes (Cohen, 1988). In contrast, the natural size of a case study paper is one, in-depth case (Lee, 1989). Like reporting multiple experiments in one article, multiple case studies should be considered exceptional. Reviewers should also consider local norms, for example average sample sizes (Caine, 2016).

The discussion of representativeness above foreshadows the difficulty of assessing a sampling strategy. The representativeness of a sample is often subjective, and representativeness is not always the goal of the sampling strategy. We suggest the following questions for guiding assessment of a sampling strategy:

1. Has the paper specified a philosophical stance?
2. Has the paper specified the goal of the sampling strategy (e.g. representativeness, convenience)?
3. Has the paper described the sample and sampling strategy sufficiently?
4. Is the sampling strategy consistent with the stated goal and philosophical position?
5. If representativeness is the goal, what argument to representativeness is made, and is it reasonable given the type of study and practical constraints?
6. Is the sampling strategy reasonable given the context, constraints and maturity of the research?
7. Are the limitations of the sample acknowledged?
8. Does the sampling strategy match the paper's knowledge claims?

It is very important for a sampling strategy to support the specific knowledge claims of the paper. When an article makes claims about a population, based on sample, or makes generic claims of the form X causes Y, we can

and should question the article's argument for representativeness. Much qualitative research, in contrast, seeks to understand one specific context with no attempt to generalize knowledge to other contexts. In such research, challenges to representativeness are far less important.

This is not to say that a paper should be rejected out of hand because some details are missing or the conclusion overreaches. Reviewers can often simply request clarifications or rewording. Even when representativeness is the goal and reasonable attempts to mitigate sampling bias have not been made, such attempts may be possible in a multi-phase review process.

The key phrases here are "reasonable" and "practical constraints." Any criticism of a paper's sampling approach should include suggesting specific, practical techniques to mitigate sampling bias. Complaining that a systematic review should have addressed sampling bias through reference snowballing is reasonable. In contrast, complaining that a study of unit tests should have used probability sampling when no reasonable sampling frame exists is naïve and unreasonable.

5.6 A Strategy for Addressing the Generalizeability Crisis

Based on the preceding discussion, we suggest a multi-pronged strategy for improving the generalizeability of software engineering research.

1. **Education**. PhD students need to take formal research methods courses. There is simply no excuse for failing to teach PhD students a variety of sampling strategies and the technical meaning of "random."
2. **Larger samples.** Many studies are simply too small for their intended purposes. It is not normally possible to detect modest effect sizes with a 12-participant experiment or to reach theoretical saturation with 6 interviewees.
3. **More probability sampling.** Probability sampling is not always appropriate, but should be used more often where it is appropriate. Many repository mining studies, for example, would benefit from probability sampling.
4. **More sophisticated sampling.** Where possible, researchers should try to replace basic sampling approaches (e.g. snowball) with more sophisticated approaches (e.g. respondent-driven) to mitigate sampling bias.
5. **Multiple sampling frames or approaches.** Drawing samples from multiple sources (e.g. countries, organizations, projects, repositories) not only increases sample heterogeneity but also provides a way to investigate whether results are source-specific.
6. **Develop better sampling frames.** A comprehensive research program is needed to develop better sampling frames and better techniques for accessing elusive populations of common units of analysis.

6 Conclusion

This paper makes several contributions:

1. An introduction to sampling with examples from SE research. This exposition is more grounded in SE research than previous discussions in reference disciplines, and more comprehensive than previous discussions within SE.
2. An analysis of the state of sampling methods in a stratified random sample of recent SE research in leading venues. It shows that probability sampling is rare, and most probability samples are drawn from unknown or non-representative sampling frames.
3. A novel exploration of the arguments for representativeness (see Table 1).
4. Guidelines for conducting, reporting and reviewing sampling.

A sample is representative of a population insofar as the relevant parameters correspond. We can assume that a random sample is representative if and only if it is sufficiently large and is drawn from an unbiased sampling frame. Few SE studies use random sampling. Of those that do, some samples are too small to assume representativeness and others are drawn from biased sampling frames. Researchers make various arguments for why their samples should be representative: the sample is large, includes diverse items, matches known population parameters, etc.

This creates a paradox: the lack of representative sampling is undermining SE research but rejecting a study over its non-representative sample is capricious because virtually none of the other studies have representative samples. We can escape this paradox by working towards more representative sampling in studies where generalizability is desired. For questionnaires especially, researchers should apply known techniques for mitigating and estimating sampling bias. We also need to develop more curated corpora of SE artifacts and better sampling frames for SE professionals.

Furthermore, we need reciprocal willingness of researchers to present their research more honestly and reviewers to stop capriciously rejecting work over unavoidable sampling bias. This means no more mislabeling ad hoc sampling as representative, no more pretending small samples are automatically representative because they are random, and no more ignoring the potential differences between sampling frames and populations. It also means no more accepting convenience sampling for experiments while criticizing convenience sampling for case studies and interviews. No more encouraging snowball sampling for literature reviews (Kitchenham and Charters, 2007) while rejecting it in questionnaires.

The contributions above should be considered in light of several limitations. We operationalized "recent high-quality software engineering research" as articles published in four top venues during the past five years. Our sample is therefore unlikely to represent the broader field. Studies that were published twice (e.g. a paper in FSE followed by an extended version in TOSEM) have a greater chance of being selected. Moreover, the analysis was hindered by

widespread confusion regarding sampling techniques and research methodologies.

Additionally, some of the guidelines suggested in this paper are not directly supported by empirical evidence. The guidelines are meta-science, and like most meta-science, are somewhat polemical. It simply is not practical to conduct experiments to determine whether aligning a study's sampling strategy with its goals, epistemology and methodology improves scientific outcomes.

In conclusion, we hope that this article's sampling primer, empirical results and recommendations raise awareness of and provide at least some basis for improving sampling in SE research.

7 Data Availability

Supplementary materials, which have been archived on Zenodo (Baltes and Ralph, 2020), include:

- The complete list of articles;
- An Excel spreadsheet including all of the data extracted from the primary studies and our analyses and decision rules;
- The scripts we used to retrieve sampling frame and sample (the most recent version of our dblp-retriever tool is available on GitHub[11]).

References

Arnett JJ (2008) The neglected 95%: why american psychology needs to become less american. American Psychologist 63(7):602

Baltes S, Diehl S (2016) Worse Than Spam: Issues In Sampling Software Developers. In: Genero M, Jedlitschka A, Jorgensen M (eds) 10th International Symposium on Empirical Software Engineering and Measurement (ESEM 2016), ACM, Ciudad Real, Spain, pp 52:1–52:6, DOI 10.1145/2961111.2962628

Baltes S, Ralph P (2020) Sampling in software engineering research supplementary material [data set]. DOI 10.5281/zenodo.3666826

Baltes S, Dumani L, Treude C, Diehl S (2018) SOTorrent: Reconstructing and Analyzing the Evolution Stack Overflow Posts. In: Zaidman A, Hill E, Kamei Y (eds) 15th International Conference on Mining Software Repositories (MSR 2018), ACM, Gothenburg, Sweden, pp 319–330

Beecham S, Baddoo N, Hall T, Robinson H, Sharp H (2008) Motivation in software engineering: A systematic literature review. Information and Software Technology 50(9):860 – 878, DOI https://doi.org/10.1016/j.infsof.2007.09.004

---

[11] https://github.com/sbaltes/dblp-retriever

Breckenridge J, Jones D (2009) Demystifying theoretical sampling in grounded theory research. Grounded Theory Review 8(2)

Caine K (2016) Local standards for sample size at chi. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, CHI '16, pp 981–992, DOI 10.1145/2858036.2858498

Charmaz K (2014) Constructing grounded theory. Sage, London

Checkland P, Holwell S (1998) Action Research: Its Nature and Validity. Systemic Practice and Action Research 11(1):9–21, DOI 10.1023/A: 1022908820784

Cochran WG (2007) Sampling techniques. John Wiley & Sons

Cohen J (1988) Statistical power analysis for the behavioral sciences. Lawrence Erlbaum Associates, Hillsdale, NJ, USA

Cummaudo A, Vasa R, Barnett S, Grundy J, Abdelrazek M (2020) Interpreting cloud computer vision pain-points: A mining study of stack overflow. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), IEEE

Daniel J (2011) Sampling essentials: Practical guidelines for making sampling choices. Sage Publications

Dillman DA, Smyth JD, Christian LM (2014) Internet, phone, mail, and mixed-mode surveys: the tailored design method, 4th edn. John Wiley & Sons, Hoboken, NJ, USA

Draucker CB, Martsolf DS, Ross R, Rusk TB (2007) Theoretical sampling and category development in grounded theory. Qualitative health research 17(8):1137–1148

Duignan B (2019) Postmodernism. In: Encyclopedia Britannica, Encyclopedia Britannica, Inc., URL https://www.britannica.com/topic/postmodernism-philosophy

Falessi D, Juristo N, Wohlin C, Turhan B, Münch J, Jedlitschka A, Oivo M (2018) Empirical software engineering experts on the use of students and professionals in experiments. Empirical Software Engineering 23(1):452–489

Faugier J, Sargeant M (1997) Sampling hard to reach populations. Journal of Advanced Nursing 26(4):790–797

Feldt R, Zimmermann T, Bergersen GR, Falessi D, Jedlitschka A, Juristo N, Münch J, Oivo M, Runeson P, Shepperd M, Sjøberg DIK, Turhan B (2018) Four commentaries on the use of students and professionals in empirical software engineering experiments. Empirical Software Engineering 23(6):3801–3820

Fitts PM (1954) The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology 47(6):381–391

Foster E (2014) Software Engineering: A Methodical Approach. Apress, New York, USA

Gentles SJ, Charles C, Ploeg J, McKibbon KA (2015) Sampling in qualitative research: Insights from an overview of the methods literature. The Qualitative Report 20(11):1772–1789

Glaser BG, Strauss AL (2017) Discovery of grounded theory: Strategies for qualitative research. Routledge

Goel S, Salganik MJ (2010) Assessing respondent-driven sampling. Proceedings of the National Academy of Sciences 107(15):6743–6747, DOI 10.1073/pnas.1000261107

Gousios G (2013) The GHTorrent dataset and tool suite. In: Zimmermann T, Di Penta M, Kim S (eds) 10th International Working Conference on Mining Software Repositories (MSR 2013), IEEE, San Francisco, CA, USA, pp 233–236

Guba EG, Lincoln YS (1982) Epistemological and methodological bases of naturalistic inquiry. Educational Communication & Technology Journal 30(4):233–252

Heckathorn DD (1997) Respondent-Driven Sampling: A New Approach to the Study of Hidden Populations. Social Problems 44(2):174–199

Henrich J, Heine SJ, Norenzayan A (2010) The weirdest people in the world? Behavioral and brain sciences 33(2-3):61–83

Henry GT (1990) Practical sampling. Sage

Hoeven LRv, Janssen MP, Roes KC, Koffijberg H (2015) Aiming for a representative sample: Simulating random versus purposive strategies for hospital selection. BMC Medical Research Methodology 15(1):90

Humbatova N, Jahangirova G, Bavota G, Riccio V, Stocco A, Tonella P (2020) Taxonomy of real faults in deep learning systems. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), IEEE

Ingram C, Drachen A (2020) How software practitioners use informal local meetups to share software engineering knowledge. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), IEEE

Johnston LG, Sabin K (2010) Sampling hard-to-reach populations with respondent driven sampling. Methodological Innovations Online 5(2):38–48, DOI 10.4256/mio.2010.0017

Juristo N, Moreno AM (2013) Basics of software engineering experimentation. Springer Science & Business Media

Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Tech. rep., Keele University and University of Durham

Kitchenham B, Pfleeger SL (2002) Principles of survey research: Part 5: Populations and samples. ACM SIGSOFT Software Engineering Notes 27(5):17–20

Kitchenham B, Pickard L, Pfleeger SL (1995) Case studies for method and tool evaluation. IEEE software 12(4):52–62

Kruskal W, Mosteller F (1979a) Representative sampling, i: Non-scientific literature. International Statistical Review 47(1):13–24

Kruskal W, Mosteller F (1979b) Representative sampling, iii: The current statistical literature. International Statistical Review 47(3):245–265, DOI 10.2307/1402647

Landon Jr EL, Banks SK (1977) Relative Efficiency and Bias of Plus-One Telephone Sampling. Journal of Marketing Research 14(3):294, DOI

10.2307/3150766

Lee AS (1989) A scientific methodology for mis case studies. MIS quarterly 13(1):33–50

Lee AS, Baskerville RL (2003) Generalizing generalizability in information systems research. Information Systems Research 14(3):221–243

Maalej W, Robillard MP (2013) Patterns of knowledge in api reference documentation. IEEE Transactions on Software Engineering 39(9):1264–1282

Malekinejad M, Johnston LG, Kendall C, Kerr LRFS, Rifkin MR, Rutherford GW (2008) Using Respondent-Driven Sampling Methodology for HIV Biological and Behavioral Surveillance in International Settings: A Systematic Review. AIDS and Behavior 12(1):105–130, DOI 10.1007/s10461-008-9421-1

van Manen M (2016) Phenomenology of practice: Meaning-giving methods in phenomenological research and writing. Routledge

McElreath R (2020) Statistical rethinking: A Bayesian course with examples in R and Stan. CRC press

de Mello RM, Travassos GH (2016) Surveys in software engineering: Identifying representative samples. In: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, Ciudad Real, Spain, p 55

Miles MB, Huberman AM, Saldaña J (2014) Qualitative data analysis: A methods sourcebook, 4th edn. Sage, Thousand Oaks, California, USA

Mohanani R, Turhan B, Ralph P (2019) Requirements framing affects design creativity. IEEE Transactions on Software Engineering DOI https://doi.org/10.1109/TSE.2019.2909033

Mohanani R, Salman I, Turhan B, Rodríguez P, Ralph P (in press) Cognitive biases in software engineering: a systematic mapping study. IEEE Transactions on Software Engineering DOI 10.1109/TSE.2018.2877759

Moher D, Liberati A, Tetzlaff J, Altman DG, Group P, et al. (2009) Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. PLoS med 6(7):e1000097

Mullinix KJ, Leeper TJ, Druckman JN, Freese J (2015) The generalizability of survey experiments. Journal of Experimental Political Science 2(2):109–138

Nagappan M, Zimmermann T, Bird C (2013) Diversity in software engineering research. In: Proceedings of the 9th Joint Meeting on Foundations of Software Engineering, ACM, pp 466–476

Patton MQ (2014) Qualitative research & evaluation methods: Integrating theory and practice. Sage publications

Paulson JW, Succi G, Eberlein A (2004) An empirical study of open-source and closed-source software products. IEEE Transactions on Software Engineering 30(4):246–256, DOI 10.1109/TSE.2004.1274044

Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: Proceedings of the Evaluation and Assessment in Software Engineering Conference (EASE), ACM, Bari, Italy, pp 68–77

Ralph P (2019) Toward methodological guidelines for process theories and taxonomies in software engineering. IEEE Transactions on Software Engi-

neering 45(7):712–735

Ralph P, Baltes S, Adisaputri G, Torkar R, Kovalenko V, Kalinowski M, Novielli N, Yoo S, Devroey X, Tan X, et al. (2020) Pandemic programming: How covid-19 affects software developers and how their organizations can help. Empirical Software Engineering DOI 10.1007/s10664-020-09875-y

Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering 14(2):131

Russo D, Stol K (in press) Gender differences in personality traits of software engineers. IEEE Transactions on Software Engineering DOI 10.1109/TSE.2020.3003413

Sax LJ, Gilmartin SK, Bryant AN (2003) Assessing response rates and non-response bias in web and paper surveys. Research in Higher Education 44(4):409–432

Sedano T, Ralph P, Péraire C (2019) The product backlog. In: 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), IEEE, pp 200–211

Shull F, Singer J, Sjøberg D (2007) Guide to Advanced Empirical Software Engineering. Springer, London

Sjøberg D, Anda B, Arisholm E, Dyba T, Jørgensen M, Karahasanovic A, Koren EF, Vokac M (2002) Conducting realistic experiments in software engineering. In: 2002 International Symposium on Empirical Software Engineering, IEEE, Nara, Japan, pp 17–26, DOI 10.1109/ISESE.2002.1166921

Staron M (2020) Action Research in Software Engineering: Theory and Applications. Springer

Stol KJ, Fitzgerald B (2018) The abc of software engineering research. ACM Transactions on Software Engineering and Methodology (TOSEM) 27(3):11

Stol KJ, Ralph P, Fitzgerald B (2016) Grounded Theory in Software Engineering Research: A Critical Review and Guidelines. In: Proceedings of the International Conference on Software Engineering, IEEE, Austin, TX, USA, pp 120–131

Tempero E, Anslow C, Dietrich J, Han T, Li J, Lumpe M, Melton H, Noble J (2010) The Qualitas Corpus: A Curated Collection of Java Code for Empirical Studies. In: Proceedings of the 17th Asia Pacific Software Engineering Conference, IEEE, Sydney, Australia, pp 336–345, DOI 10.1109/APSEC.2010.46

Thomas G, Myers K (2015) The anatomy of the case study. Sage

Thompson SK (1990) Adaptive cluster sampling. Journal of the American Statistical Association 85(412):1050–1059

Toepoel V (2012) Effects of incentives in surveys. In: Gideon L (ed) Handbook of survey methodology for the social sciences, Springer, pp 209–223

Trochim WM, Donnelly JP (2001) Research methods knowledge base, vol 2. Atomic Dog Publishing, Cincinnati, OH, USA

Trost JE (1986) Statistically nonrepresentative stratified sampling: A sampling technique for qualitative studies. Qualitative Sociology 9(1):54–57

Turk P, Borkowski JJ (2005) A review of adaptive cluster sampling: 1990–2003. Environmental and Ecological Statistics 12(1):55–94

Valliant R, Dever JA, Kreuter F (2018) Designing multistage samples. In: Practical Tools for Designing and Weighting Survey Samples, Springer, pp 209–264

Vasilescu B, Posnett D, Ray B, van den Brand MG, Serebrenik A, Devanbu P, Filkov V (2015) Gender and Tenure Diversity in GitHub Teams. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15, ACM Press, Seoul, Republic of Korea, pp 3789–3798, DOI 10.1145/2702123.2702549

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer Science & Business Media

Yin RK (2018) Case study research: Design and methods, 6th edn. Sage, Thousand Oaks, California, USA