CHAPMAN & HALL/CRC INNOVATIONS IN
SOFTWARE ENGINEERING AND SOFTWARE DEVELOPMENT

EVIDENCE-BASED SOFTWARE ENGINEERING AND SYSTEMATIC REVIEWS

Barbara Ann Kitchenham
David Budgen
Pearl Brereton

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

EVIDENCE-BASED SOFTWARE ENGINEERING AND SYSTEMATIC REVIEWS

Chapman & Hall/CRC Innovations in Software Engineering and Software Development

Series Editor
Richard LeBlanc
Chair, Department of Computer Science and Software Engineering, Seattle University

AIMS AND SCOPE

This series covers all aspects of software engineering and software development. Books in the series will be innovative reference books, research monographs, and textbooks at the undergraduate and graduate level. Coverage will include traditional subject matter, cutting-edge research, and current industry practice, such as agile software development methods and service-oriented architectures. We also welcome proposals for books that capture the latest results on the domains and conditions in which practices are most effective.

PUBLISHED TITLES

Computer Games and Software Engineering
Kendra M. L. Cooper and Walt Scacchi

Software Essentials: Design and Construction
Adair Dingle

Software Metrics: A Rigorous and Practical Approach, Third Edition
Norman Fenton and James Bieman

Software Test Attacks to Break Mobile and Embedded Devices
Jon Duncan Hagar

Software Designers in Action: A Human-Centric Look at Design Work
André van der Hoek and Marian Petre

Evidence-Based Software Engineering and Systematic Reviews
Barbara Ann Kitchenham, David Budgen, and Pearl Brereton

Fundamentals of Dependable Computing for Software Engineers
John Knight

Introduction to Combinatorial Testing
D. Richard Kuhn, Raghu N. Kacker, and Yu Lei

Building Enterprise Systems with ODP: An Introduction to Open Distributed Processing
Peter F. Linington, Zoran Milosevic, Akira Tanaka, and Antonio Vallecillo

Software Engineering: The Current Practice
Václav Rajlich

Software Development: An Open Source Approach
Allen Tucker, Ralph Morelli, and Chamindra de Silva

CHAPMAN & HALL/CRC INNOVATIONS IN
SOFTWARE ENGINEERING AND SOFTWARE DEVELOPMENT

EVIDENCE-BASED SOFTWARE ENGINEERING AND SYSTEMATIC REVIEWS

Barbara Ann Kitchenham
Keele University, Staffordshire, UK

David Budgen
Durham University, UK

Pearl Brereton
Keele University, Staffordshire, UK

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20151022

International Standard Book Number-13: 978-1-4822-2866-3 (eBook - PDF)

Contents

This page intentionally left blank

List of Figures

List of Tables

Preface

As a relatively young (and as we will later argue, still somewhat immature) discipline, *software engineering* is in an emergent[1] state for many purposes. Its foundations as a distinct sub-discipline of computing are widely considered to have been laid down at the 1968 NATO conference, although the term was probably in fairly regular use before that. Since then, ideas have ebbed and flowed, along with the incredibly rapid expansion and evolution of computing from an activity largely concerned with 'crunching numbers' in support of scientific research, to something that forms a pervasive element of everyday life. While this has helped to drive the development of software engineering as a discipline, the headlong pace has also meant that there has often been little opportunity to appraise and reflect upon our experiences of how software systems can be developed, how well the different approaches work, and under what conditions they are likely to be most effective.

The emergence of the concept of *evidence-based software engineering* (EBSE) can certainly be assigned a clear starting point, with the seminal paper being presented at the 2004 International Conference on Software Engineering (ICSE). In the decade that has followed, ideas about EBSE, and about its key tool, the systematic review, have evolved and matured; it has taken its place in the empirical software engineer's toolbox; and has helped to categorise and consolidate our knowledge about many aspects of software engineering research and practice. While few commercial software development activities can as yet even be described as 'evidence-informed', the philosophy of EBSE is beginning to be widely recognised and appreciated. As such then, this seems to be a suitable time to bring this knowledge together in a single volume, not least to help focus thinking about what we as a community might usefully do with that knowledge in the future.

Like Gaul, our book is divided into three parts[2]. In the first part we discuss the nature of evidence and the evidence-based practices centred around the systematic review, both in general and also as applying to software engineering. The second part examines the different elements that provide inputs to a systematic review (usually considered as forming a *secondary* study), especially the main forms of primary empirical study currently used in software

[1] An emergent process is one that is 'in a state of continual process change, never arriving, always in transition' (Truex, Baskerville & Klein 1999).

[2] Those with a classical education will remember that this was the first observation in Julius Caesar's *The Conquest of Gaul*, and quite possibly, that is the only thing that many of us remember from that work!

engineering. Lastly, the third part provides a practical guide to conducting systematic reviews (the *guidelines*), drawing together accumulated experiences to guide researchers and students when they are planning and conducting their own studies. In support of these we also include an extensive *glossary*, and an appendix that provides a *catalogue* of reviews that may be useful for practice and teaching.

This raises the question of who we perceive to be the audience for this book. We would like to think that almost anyone with any involvement in software engineering (in the broadest sense) can find something of use within it, given that our focus is upon seeking to identify what works in software engineering, when it works, and why. For the researcher, it provides guidance on how to make his or her own contribution to the corpus of knowledge, and how to determine where the research efforts might be directed to best effect. For practitioners, the book both explains the foundations of evidence-based knowledge related to software engineering practices, and also identifies useful examples of this. Finally, for teachers and students, it provides an introduction to the nature and role of empirical software engineering and explains what empirical studies can tell us about our subject.

So, how should the aspiring empiricist, or even the merely curious, approach all of this material, assuming that he or she might be reluctant to attempt to devour each chapter in turn, in the way that they would read a novel? We would suggest that the first few chapters provide a background to EBSE that should be relevant to anyone. These chapters explain the basic thinking about evidence-based studies and concepts, and show how they can be applied within a software engineering context.

The researcher, including of course, all PhD students, should additionally read the rest of Part I, so as to understand how to plan a secondary study. Armed with this understanding they can then turn to Part III, which provides essential practical guidance on the conduct of such a study, and which can then lead them through the steps of putting their plan into action. And, should any researcher determine that the ground is not yet solid enough for a secondary study, they can turn to Part II to learn something about how to conduct and report on a primary study in such a way as to make it a useful input to a future secondary study. Indeed, even when undertaking a secondary study, Part II should also be useful to the systematic reviewer when he or she is facing the tasks of data extraction and synthesis, by explaining something of the context behind the different forms of empirical study that provide the inputs to their analysis.

Practitioners and others who want to know more about EBSE and the use of secondary studies may find that Part I provides much of what they need in order to understand (and use) the outcomes from secondary studies. Likewise, teachers will, we hope, find much useful material in both Part I and Part II, in the latter case because an understanding of secondary studies is best founded upon a solid appreciation of the roles and forms of primary studies. Both

of these groups should also find material that is of direct usefulness in the catalogue of reviews provided in the appendix.

We are teachers as well as researchers, and should observe here that teaching the practices used in performing secondary studies to advanced undergraduates can be beneficial too. Students usually need to undertake a literature review as part of their individual 'capstone' projects, and adopting a systematic and objective approach to this can add valuable rigour to the outcomes.

In writing this book, we have drawn upon our own experiences with conducting systematic reviews and primary studies, and so our material and its organisation build upon the lessons that we have learned through these. These experiences have included both designing our own studies and reviewing the designs of others, and with conducting both methodological studies as well as ones that examine some established software engineering practices. Wherever possible we have tried to illustrate our points by drawing upon these experiences, as well as learning from those of many others, whose contribution to EBSE and its development we gratefully acknowledge.

This leads to an issue that always presents something of a challenge for evidence-based researchers such as ourselves, namely that of how to handle *citation*. As evidence-based software engineering researchers we usually feel it necessary to justify everything we possibly can by pointing to relevant evidence—but equally as authors, we are aware that this risks present the reader with a solid wall of reference material, which itself can form a distraction from gaining an understanding of key concepts. We have therefore tried to find a balance, providing citations whenever we think that the reader may possibly wish to confirm or clarify ideas for themselves. At the same time we have tried to avoid a compulsive need for justification at every opportunity, and especially when this is not really essential to enjoying the text—and of course, a sense of interest and enjoyment is exactly what we sincerely hope others will be able to experience from learning about EBSE and how the use of systematic reviews can help to inform software engineering as a discipline.

Finally, as a related point, since all the chapters of Part I relate to different aspects of secondary studies, we have provided a single set of suggestions for *further reading* at the end of this part, in order to avoid undue repetition. In Part II, where we address different forms of primary study in each chapter, we have reverted to the more conventional approach of providing recommendations for further reading at the end of each chapter.

This page intentionally left blank

Glossary

The vocabulary used in this book has been derived from a variety of sources and disciplines, which is not unreasonable, as that is how the ideas of empirical software engineering have themselves been derived. Our glossary does not purport to be definitive, the aim is to convey the relevant concepts quickly, so that when consulting it, the reader does not have to stray far from the flow of what they are reading.

absolute (measurement scale): This is the most restrictive of the measurement scales and simply uses counts of the elements in a set of entities. The only operation that can be performed is a test for equality. (See also *measurement scales*.)

accuracy: The accuracy of a measurement is an assessment of the degree of conformity of a measured or calculated value to its actual or specified value.

accuracy range: The accuracy range tells us how close a sample is to the true population of interest, and is usually expressed as a plus/minus margin. (See also *confidence interval*.)

aggregation: The process of gathering together knowledge of a particular type and form (for example, in a table).

attribute: An attribute is a measurable (or at least, identifiable) characteristic of an entity, and as such provides a mapping between the abstract idea of a *property* of the entity and something that we can actually measure in some way.

between-subject: (Also known as *between-groups* or *parallel experiment*.) In this form of study, participants are assigned to different treatment (intervention) groups and each participant only receives one treatment.

bias: A tendency to produce results that depart systematically from the true results.

blinding: A process of concealing some aspect of an experiment from researchers and participants. In single-blind experiments, participants do not know which treatment they have been assigned to. In double-blind

experiments, neither participants nor experimenters know which treatment the participants have been assigned to. In software engineering we sometimes use blind-marking, where the marker does not know which treatment the participants adopted to arrive at their answers or responses.

case study: A form of *primary study*, which is an investigation of some phenomenon in a real-life setting. Case studies are typically used for *explanatory*, *exploratory* and *descriptive* purposes. The main two forms are *single-case* studies which may be appropriate when studying a representative case or a special case, but will be less trustworthy than *multiple-case* forms, where replication is employed to see how far different cases predict the same outcomes. (Note that the term *case study* is sometimes used in other disciplines to mean a narrative describing an example of interest.) Case study research is covered in detail in Yin (2014) and for software engineering, in Runeson, Höst, Rainer & Regnell (2012).

causality: The link between a stimulus and a response, in that one *causes* the other to occur (also termed cause and effect). The notion of some form of causality usually underpins *hypotheses*.

central tendency: The 'typical value' for a probability distribution. The three most common measures used for this are the *mean*, the *median* and the *mode*. (See the separate definitions of these.)

closed question: (As used in a questionnaire.) Such a question constrains respondents by requiring them to select from a pre-determined list of answers. This list may optionally include 'other' or 'don't know' options. (See also *open question*.)

conclusion validity: (See *validity*.)

confidence interval: This is an assessment of how sure we are that the region within the stated interval around our measured mean does contain the true mean. This is expressed as a percentage, for example, a confidence interval of 95% (which corresponds to two standard deviations either side of the mean) means that there is a 95% likelihood that the true population mean lies within two standard deviations of our sample mean. So, for this value of the confidence interval, if we did many independent experiments and calculated confidence intervals for each of these, the true mean of the population being studied would be within the confidence limits in 95% of these.

confounding factor: An undesirable element in an experimental study that produces an effect that is indistinguishable from that of one of the treatments.

construct validity: (See *validity*.)

content validity: (As used in a survey.) Concerned with whether the questions are a well-balanced sample for the domain we are addressing.

control group: For laboratory experiments we can divide the participants into two groups—with the *treatment group* receiving the experimental treatment being investigated, and the experimental context of the *control group* involving no manipulation of the independent variable(s). It is then possible to attribute any differences between the outcomes for the two groups as arising from the treatment.

controlled experiment: (See *laboratory experiment*, *field experiment* and *quasi-experiment*.)

convenience (sample): A form of *non-probabilistic sampling* in which participants are selected simply because it is easy to get access to them or they are willing to help. (See *sampling technique*.)

cross-over: (See *within-subject*.)

dependent variable: (Also termed *response variable* or *outcome variable*.) This changes as a result of changes to the independent variable(s) and is associated with an *effect*. The outcomes of a study are based upon measurement of the dependent variable.

descriptive (survey): (See *survey*.)

direct measurement: Assignment of values to an attribute of an entity by some form of counting.

divergence: A divergence occurs when a study is not performed as specified in the *experimental protocol*, and all divergences should be both recorded during the study and reported at the end.

double blinding: (See *blinding*.)

dry run: For an experiment, this involves applying the experimental treatment to (usually) a single recipient, in order to test the experimental procedures (which may include training, study tasks, data collection and analysis). May sometimes be termed a *pilot experiment*. A similar activity may be performed for a survey instrument.

effect size: The effect size provides a measure of the strength of a phenomenon. There are many measures of effect size to cater to different types of treatment outcome measures, including the standardized mean differences, the log odds ratio, and the Pearson correlation coefficient.

empirical: Relying on observation and experiment rather than theory (*Collins English Dictionary*).

ethics: The study of standards of conduct and moral judgement (*Collins English Dictionary*). Codes of ethics for software engineering are published by the British Computer Society and the ACM/IEEE. Any empirical study that involves human participants should be vetted by the researcher's local *ethics committee* to ensure that it does not disadvantage any of the participants in any way.

ethnography: A form of observational study that is purely observational, and hence without any form of intervention or participation by the observer.

evidence-based: An approach to empirical studies by which the researcher seeks to identify and integrate the best available research evidence with domain expertise in order to inform practice and policy-making. The normal mechanism for identifying and aggregating research evidence is the *systematic review*.

exclusion criteria: After performing a search for papers (primary studies) when performing a systematic review, the exclusion criteria are used to help determine which ones will not be used in the study. (See also *inclusion criteria*.)

experiment: A study in which an intervention (i.e. a treatment) is deliberately controlled to observe its effects (Shadish, Cook & Campbell 2002).

external attribute: An external attribute is one that can be measured only with respect to how an element relates to other elements (such as reliability, productivity, etc.).

field experiment: An experiment or quasi-experiment performed in a natural setting. A field experiment usually has a more realistic setting than a laboratory experiment, and so has greater external validity.

field study: A generic term for an empirical study undertaken in real-life conditions.

hypothesis: A testable *prediction* of a cause–effect link. Associated with a hypothesis is a *null hypothesis* which states that there are no underlying trends or dependencies and that any differences observed are coincidental. A statistical test is normally used to determine the probability that the null hypothesis can or cannot be rejected.

inclusion criteria: After performing a search for papers (primary studies) when performing a systematic review, the inclusion criteria are used to help determine which ones contain relevant data and hence will be used in the study. (See also *exclusion criteria*.)

independent variable: An independent variable (also known as a *stimulus* variable or an *input* variable) is associated with *cause* and is changed as a

result of the activities of the investigator and not of changes in any other variables.

indirect measurement: Assigning values to an attribute of an entity by measuring other attributes and using these with some form of 'measurement model' to obtain a value for the attribute of interest.

input variable: (See *independent variable*.)

instrument: The 'vehicle' or mechanism used in an empirical study as the means of data collection (for the example of a survey, the instrument might be a questionnaire).

internal attribute: A term used in software metrics to refer to a measurable attribute that can be extracted directly from a software document or program without reference to other software project or process attributes.

interpretivism: In information systems research and computing in general, interpretive research is 'concerned with understanding the social context of an information system: the social processes by which it is developed and construed by people and through which it influences, and is influenced by, its social setting' (Oates 2006). (See also *positivism*.)

interval scale: An interval scale is one whereby we have a well-defined ratio of intervals, but have no absolute zero point on the scale, so that we cannot speak of something being 'twice as large'. Operations on interval values include testing for equivalence, greater and less than, and for a known ratio. (See also *measurement scales*.)

interview: A mechanism used for collecting data from participants for surveys and other forms of empirical study. The forms usually encountered are *structured*, *semi-structured* and *unstructured*. The data collected are primarily subjective in form.

laboratory experiment: Sometimes referred to as a *controlled laboratory experiment*, this involves the identification of precise relationships between experimental variables by means of a study that takes place in a controlled environment (the 'laboratory') involving human participants and supported by quantitative techniques for data collection and analysis.

longitudinal: Refers to a form of study that involves repeated observations of the same items over long periods of time.

mapping study: A form of secondary study intended to identify and classify the set of publications on a topic. May be used to identify 'evidence gaps' where more primary studies are needed as well as 'evidence clusters' where it may be practical to perform a systematic review.

mean: Often referred to as the *average*, and one of the three most common measures of the *central tendency*. Computed by adding the data values and dividing by the number of elements in the dataset. It is only meaningful for data forms that have genuinely numerical values (as opposed to codes).

measurement: The process by which numbers or symbols are assigned to attributes of real-world entities using a well-defined set of rules. Measurement may be direct (for example, length) or indirect, whereby we measure one or more other attributes in order to obtain the value (such as measuring the length of a column of mercury on a thermometer in order to measure temperature).

measurement scales: The set of scales usually used by statisticians are absolute, nominal, ordinal, interval and ratio. (See the separate definitions of these for details). A good discussion of the scales and their applicability is provided in Fenton & Pfleeger 1997.

median: (Also known as the 50th percentile.) One of the three most common measures of the *central tendency*. This is the value that separates the upper half of a set of values from the lower half, and is computed by ordering the values and taking the middle one (or the average of two middle ones if there is an even number of elements). Then half of the elements have values above the median and half have values below.

meta-analysis: The process of statistical pooling of similar quantitative studies.

mode: One of the three most common measures of the *central tendency*. This is the value that occurs most frequently in a dataset.

nominal measurement scale: A nominal scale consists of a number of categories, with no sense of ordering. So the only operation that is meaningful is a test for equality (or inequality). An example of a nominal scale might be programming languages. (See also *measurement scales*.)

null hypothesis: (See *hypothesis*.)

objective: Objective measures are those that are independent of the observer's own views or opinions, and so are repeatable by others. Hence they tend to be quantitative in form.

observational scale: An observational scale seeks simply to record the actions and outcomes of a study, usually in terms of a pre-defined set of factors, and there is no attempt to use this to confirm or refute any form of hypothesis. Observational scales are commonly used for diagnosis or making comparison between subjects or between subjects and a benchmark. For research, they may be used to explore an issue and to determine whether more rigorous forms might then be employed.

open question: (As used in a questionnaire.) An open question is one that leaves the respondent free to provide whatever answer they wish, without any constraint on the number of possible answers. See also *closed question.*

ordinal scale: An ordinal scale is one that *ranks* the elements, but without there being any sense of a well-defined interval between the different elements. An example of such a scale might be *cohesion*, where we have the idea that particular forms are better than others, but no measure of how much. Operations are equality (inequality) and greater than/less than. (See also *measurement scales.*)

outcome variable: (See *dependent* variable.)

participant: Someone who takes part (participates) in a study, sometimes termed a *subject.* Participant is the better term in a software engineering context because involvement nearly always has an active element, whereas subject implies a passive recipient.

population: A group of individuals or items that share one or more characteristics from which data can be extracted and analysed. (See *sampling frame.*)

positivism: The philosophical paradigm that underlies what is usually termed the 'scientific method'. It assumes that the 'world' we are investigating is ordered and regular, rather than random, and that we can investigate it in an objective manner. It therefore forms the basis for hypothesis-driven research. For a fuller discussion, see (Oates 2006).

power: (See *statistical power.*)

precision: (See also *recall.*) In the context of information retrieval, the *precision* of the outcomes of a search is a measure of the proportion of studies found that are *relevant.* (Note that this makes no assumptions about whether or not all possible relevant documents were found.) If the number of relevant documents N_{rel} is defined as

$$N_{rel} = N_{retr} - \overline{N_{rel}}$$

where N_{retr} is the number retrieved and $\overline{N_{rel}}$ is the number that is classified as not relevant, then

$$precision = \frac{N_{rel}}{N_{retr}}$$

Hence if we retrieve 20 documents, of which 8 are not relevant, the value for precision will be $(20 - 8)/20$ or 0.6. So a value of 1.0 for precision indicates that all of the documents found were relevant, but says nothing about whether every relevant document was found.

primary study: This is an empirical study in which we directly make measurements about the objects of interest, whether by surveys, experiments, case studies, etc. (See also *secondary study*.)

proposition: (In the context of a case study.) This is a more detailed element derived from a *research question* and performs a role broadly similar to that of a *hypothesis* (and like a hypothesis can be derived from a theory). Propositions usually form the basis of a case study and help to guide the organisation of data collection (Yin 2014). However, an *exploratory* case study would not be expected to involve the use of any propositions.

protocol: In the context of empirical studies, this term is used in two similar (but different) ways.

- For empirical studies in general, the *experimental protocol* is a document that describes the way that a study is to be performed. It should be written before the study begins and evaluated and tested through a 'dry run'. During the actual study, any *divergences* from the protocol should be recorded. It is this interpretation that is used throughout this book.
- The practice of *protocol analysis* can be used for qualitative studies, forming a data analysis technique that is based upon the use of *think-aloud*. In this, the protocol provides a categorisation of possible utterances that can be used to analyse the particular sequence of words produced by a participant while performing a task, as well as to strip out irrelevant material (Ericsson & Simon 1993).

qualitative: A measurement form that (typically) involves some form of human judgement or assessment in assigning values to an attribute, and hence which may use an ordinal scale or a nominal scale. Qualitative data is also referred to as *subjective data*, but such data can be quantitative, such as responses to questions in survey instruments.

quantitative: A measurement form that involves assigning values to an attribute using an interval scale or (more typically) a ratio scale. Quantitative data is also referred to as *objective data*, however this is incorrect, since is it possible to have quantitative subjective data.

quasi-experiment: An experiment in which units are not assigned at random to the interventions (Shadish et al. 2002).

questionnaire: A data collection mechanism commonly used for surveys (but also in other forms of empirical study). It involves participants in answering a series of questions (which may be 'open' or 'closed').

randomised controlled trial (RCT): A form of large-scale controlled experiments performed in the field using a random sample from the population of interest and (ideally) *double blinding*. In clinical medicine this is

regarded as the 'gold standard' in terms of experimental forms, but there is little scope to perform RCTs in disciplines (such as software engineering) where individual participant skill levels are involved in the treatment.

randomised experiment: An experiment in which units are assigned to receive the treatment or alternative condition by a random process such as a coin toss or a table of random numbers.

ratio scale: This is a scale with well-defined intervals and also an absolute zero point. Operations include equality, greater than, less than, and ratio—such as 'twice the size'. (See also *measurement scales.*)

reactivity: This refers to a change in the participant's behaviour arising from being tested as part of the study, or from trying to help the experimenter (hypothesis guessing). It may also arise because of the influence of the experimenter (forming a source of bias).

recall: (See also *precision.*) In the context of information retrieval, the *recall* of the outcomes of a search (also termed *sensitivity*) is a measure of the proportion of all relevant studies found in the search. If the number of relevant documents N_{rel} is defined as

$$N_{rel} = N_{retr} - \overline{N_{rel}}$$

where N_{retr} is the number retrieved and $\overline{N_{rel}}$ is the number that is classified as not relevant, then

$$recall = \frac{N_{rel}}{N_{rel}^{tot}}$$

where N_{rel}^{tot} is the total number of documents that are relevant (if you know it). Hence if we retrieve 20 documents of which 8 are not relevant, and we know that there are no other relevant ones, then the value for recall will be $(20 - 8)/12$ or 1.0. So while a value of 1.0 for recall indicates that all relevant documents were found, it does not indicate how many irrelevant ones were also found.

research question: The research question provides the rationale behind any primary or secondary empirical study, and states in broad terms the issue that the study is intended to investigate. For experiments this will be the basis of the *hypothesis* used, but the idea is equally valid when applied to a more observational form of study.

response rate: For a survey, the response rate is the proportion of surveys completed and returned, compared to those issued.

response variable: An alternative term for the *dependent variable.*

sample: This is the set (usually) of people who act as participants in a study (for example, a survey or a controlled laboratory experiment). Equally, it can be a sample set of documents or other entities as appropriate. An important aspect of a sample is the extent to which this is representative of the larger population of interest.

sample size: This is the size of the sample needed to achieve a particular *confidence interval* (with a 95% confidence interval as a common goal). As a rule of thumb, if any statistical analysis is to be employed, even at the level of calculating means and averages, a sample size of at least 30 is required.

sampling frame: This is the set of entities that could be included in a survey, for example, people who have been on a particular training course, or who live in a particular place.

sampling technique: This is the strategy used to select a sample from a sampling frame and takes two main forms:

non-probabilistic sampling Employed where it is impractical or unnecessary to have a representative sample. Includes purposive, snowball, self-selection and convenience sampling.

probabilistic sampling An approach that aims to obtain a sample that forms a representative cross-section of the sampling frame. Includes random, systematic, stratified and cluster sampling.

secondary study: A secondary study does not generate any data from direct measurements, instead it analyses a set of *primary studies* and usually seeks to aggregate the results from these in order to provide stronger forms of *evidence* about a particular phenomenon.

statistical power: The ability of a statistical test to reveal a true pattern in the data (Wohlin, Runeson, Höst, Ohlsson, Regnell & Wesslen 2012). If the power is low, then there is a high risk of drawing an erroneous conclusion. For a detailed discussion of statistical power in software engineering studies, see (Dybå, Kampenes & Sjøberg 2006).

stimulus variable: (See *independent variable.*)

subjective: Subjective measures are those that depend upon a value judgement made by the observer, such as a ranking ('A is more significant than B). May be expressed as a qualitative value ('better') or in a quantitative form by using an ordinal scale.

survey: A comprehensive research method for collecting information to describe, compare or explain knowledge, attitudes and behaviour. The purpose of a survey is to collect information from a large group of people in a standard and systematic manner and then to seek *patterns* in the

resulting data that can be generalised to the wider population. Surveys can be broadly classified as being

- *experimental* when used to assess the impact of some intervention
- *descriptive* if used to enable assertions to be made about some phenomenon of interest and the distribution of particular attributes—where the concern is not *why* the distribution exists, but *what* form it has

synthesis: The process of systematically combining different sources of data (evidence) in order to create new knowledge.

systematic (literature) review: This is a particular form of *secondary study* and aims to provide an objective and unbiased approach to finding relevant primary studies, and for extracting, aggregating and synthesising the data from these.

tertiary study: This is a secondary study that uses the outputs of secondary studies as its inputs, perhaps by examining the secondary studies performed in a complete discipline or a part of it.

test–retest: Conventionally, this forms a measure of the *reliability* and *stability* of a survey instrument. Respondents are 'tested' at two well-separated points in time, and the responses are compared for consistency by means of a correlation test, with correlation values of 0.7–0.8 usually being considered satisfactory. Use of test–retest is only appropriate in situations where 'learning' effects are unlikely to occur within the intervening time period. In the context where a single researcher is performing a systematic review, the use of test–retest can be interpreted as being for the researcher to perform such tasks as *selection* and *data extraction* twice, with these being separated by a suitable time interval, and to check for consistency between the two sets of outcomes. Where possible, these tasks should be performed using different orderings of the data items, in order to reduce possible bias.

treatment: This is the 'intervention' element of an experiment (the term is really more appropriate to *randomised controlled trials* where the participants are recipients). In software engineering it may take the form of a task (or tasks) that participants are asked to perform such as writing code, testing code, reading documents.

triangulation: Refers to the use of multiple elements that reinforce one another in terms of providing evidence, where no single source would be adequately convincing. The 'sources' may be different forms of data, or the outcomes from different research methods.

validity: This is concerned with the degree to which we can 'trust' the outcomes of an empirical study, usually assessed in terms of four commonly encountered forms of *threat to validity*. The following definitions are based upon those provided in Shadish et al. (2002).

- *internal:* Relating to inferences that the observed relationship between treatment and outcome reflects a cause–effect relationship.
- *external:* Relating to whether a cause–effect relationship holds over other conditions, including persons, settings, treatment variables and measurement variables.
- *construct:* Relating to the way in which concepts are operationalised as experimental measures.
- *conclusion:* Relating inferences about the relationship between treatment and outcome variables.

within-subject: Refers to one of the possible design forms for a quasi-experiment. In this form, participants receive a number of different treatments, with the order in which these are received being randomised. A commonly encountered design (two treatments) is the *A/B–B/A crossover* whereby some participants receive treatment A and then treatment B, while others receive them in reverse order. A weaker version is a *before–after* design, whereby all participants perform a task, are then given some training (the treatment), and are then asked to undertake another task. (Also known as a *sequential* or *repeated-measures* experiment.)

Part I

Evidence-Based Practices in Software Engineering

This page intentionally left blank

Chapter 1

The Evidence-Based Paradigm

Since this is a book that is about the use of evidence-based research practices, we feel that it is only appropriate to begin it by considering what is meant by *evidence* in the general sense. However, because this is also a book that describes how we acquire evidence about software engineering practices, we then need to consider some of the ways in which ideas about evidence are interpreted within the rather narrower confines of science and technology.

Evidence is often associated with *knowledge*. This is because we would usually like to think that our knowledge about the world around us is based upon some form of evidence, and not simply upon wishful thinking. If we go to catch a train, it might be useful to have evidence in the form of a timetable that shows the intention of the railway company to provide a train at the given time that will take us to our destination. Or, rather differently, if we think that some factor might have caused a 'population drift' away from the place where we live, we might look at past census data to see if such a drift really has occurred, and also whether some groups have been affected more than others. Of course the link between evidence and knowledge is rarely well-defined, as in our second example, where any changes in population we observe might arise from many different factors. Indeed, it is not unusual, in the wider world at least, for the same evidence to be interpreted differently (just think about global warming).

In this chapter we examine what is meant by evidence and knowledge, and the processes by which we interpret the first to add to or create the second. We also consider some limitations of these processes, both those that are intrinsic, such as those that arise from the nature of the things being studied, and of data itself, and also those that arise from the inevitable imperfections of research practice. In doing so, we prepare the ground for Chapter 2, where we look at how the discipline of software engineering interprets these concepts, and review the characteristics of software engineering that influence the nature of our evidence—and hence the nature of our knowledge too.

1.1 What do we mean by evidence?

As noted above, evidence can be considered as being something that underpins *knowledge*, and we usually expect that knowledge will be derived from evidence through some process of *interpretation*. The nature of that interpretation can take many forms. For example, it might draw upon other forms of knowledge, as when the fictional detective Sherlock Holmes draws upon his knowledge about different varieties of tobacco ash, or about the types of earth to be found in different parts of London, in order to turn a clue into evidence. Interpretation might also be based upon mathematical or statistical procedures, such as when a scientist gathers together different forms of experimental and observational data—for example, using past medical records to demonstrate that smoking is a cause of lung cancer. Yet another, less scientific, illustration of the concept is when the jury at a criminal trial has to consider the evidence of a set of witnesses in order to derive reasonable knowledge about what actually happened. Clearly these differ in terms of when they arise, the form of knowledge derived, and the rigour of the process used for its derivation (and hence the *quality* of the resulting knowledge). What they do have in common though, is that our confidence about the knowledge will be increased if there is more than one source (and possibly form) of evidence. For the fictional detective, this may be multiple clues; for the clinical analysis it might involve using records made in many places and on patients who have different medical histories; for the jury, it may be that there are several independent witnesses whose statements corroborate each other. This process of *triangulation* between sources (a term derived from navigation techniques) is also an important means of testing the *validity* of the knowledge acquired.

Science in its many forms makes extensive use of these concepts, although not always expressed using this vocabulary. Over the years, particular scientific disciplines have evolved their own accepted set of empirical practices that are intended to give confidence in the validity and quality of the knowledge created from the forms of evidence considered to be appropriate to that discipline, and also to assess how strong that confidence is. Since this book is extensively concerned with different forms of *empirical* study, this is a good point to note that such studies are ones that are based upon *observation* and *measurement*. Indeed, this is a reminder that, strictly speaking, scientific processes never 'prove' anything (mathematics apart), they only 'demonstrate' that some relationship exists between two or more factors of interest. Even physicists, who are generally in the best position to isolate factors, and to exclude the effect of the observation process, are confronted with this issue. The charge on an electron, or the universal gravitational constant, may well be known to a very high level of precision, and with high confidence, but even so, some residual uncertainty always remains. For disciplines where it can be harder to separate out the key experimental characteristics and where (hor-

rors), humans are involved in roles other than as observers, so the element of variability will inevitably increase. This is of course the situation that occurs for many software engineering research studies, and we will look at some of the consequences in the next chapter.

When faced with evidence for which the values and quality may vary, the approach generally adopted is to use repeated observations, as indicated above, and even better, to gather observations made by different people in different locations. By pooling these, it becomes easier to identify where we can recognise repeated occurrences of patterns in the evidence that can be used to provide knowledge. This repetition also helps to give us confidence that we are not just seeing something that has happened by chance.

The assumption that it is meaningful to aggregate the observations from different studies and to seek patterns in these is termed a *positivist* philosophy. Positivism is the philosophy that underpins the 'scientific method' in general, as well as almost all of the different forms of empirical study that are described in this book.

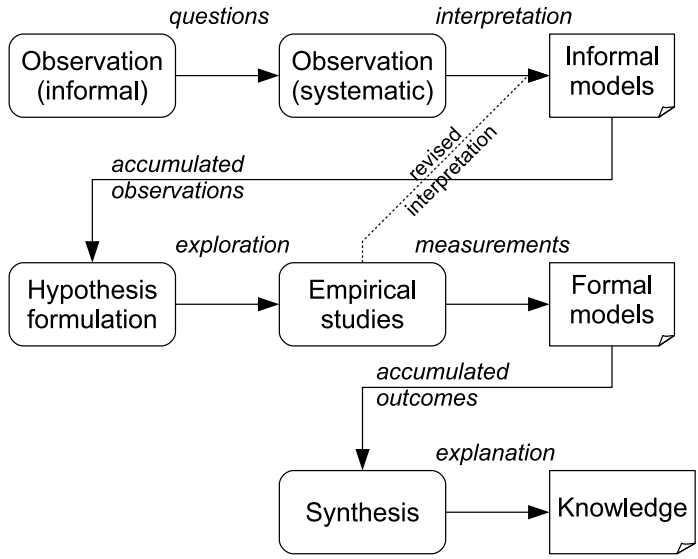

FIGURE 1.1: A simple model of knowledge acquisition.

Figure 1.1 shows a simple model that describes how these concepts relate to one another in a rather general sense. The top row represents how, having noticed the possible presence of some effect, we might begin gathering observations to create a rather informal model to describe some phenomenon. This model might well identify more than one possible cause. If this looks promising, then we might formulate a hypothesis (along the lines that "factor X causes outcome Y to occur") and perform some more systematically

organised studies to explore and test this model, during which process, we may discard or revise our ideas about some of the possible causes. Finally, to confirm that our experimental findings are reliable, we encourage others to repeat them, so that our knowledge is now accumulated from many sources and gathered together by a process that we refer to as *synthesis*, so that the risk of bias is reduced. Many well-known scientific discoveries have followed this path in some way, such as the discovery of X-rays and that of penicillin.

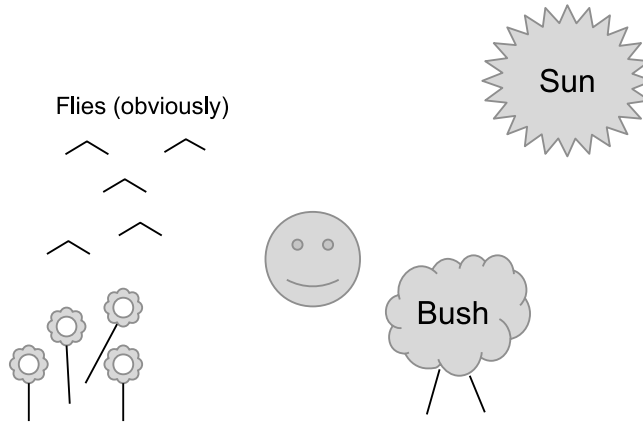

FIGURE 1.2: Does the bush keep the flies off?

Since this is rather abstract, let's consider a simple (slightly contrived but not unrealistic) example. This is illustrated (very crudely) in Figure 1.2. If we imagine that, while sitting out in a garden one day in order to enjoy the summer sunshine, we notice that we are far less bothered by flies when sitting near a particular bush, then this provides an example of informal observation. If we get enough good weather (we did say this example was contrived), we might try repeating the observation, perhaps by sitting near other bushes of that variety. If we continue to notice the effect, then this now constitutes an informal model. Encouraged by visions of the royalties that could arise from discovering a natural insecticide, we might then go on to pursue this rather more systematically, and of course, in so doing we will probably find all sorts of other possible explanations, or indeed, that it is not really an effect at all. But of course, we might also just end up with some systematically gathered knowledge about the insect-repellent nature of this plant (or perhaps, of this plant in conjunction with other factors).

This book is mainly concerned with the bottom two layers of the model shown in Figure 1.1. In Part I and Part III we are concerned with how knowledge from different sources can be 'pooled', while in Part II we provide a subject-specific interpretation of what is meant by the activities in the middle

layer. In particular, we will be looking at ways of gathering evidence that go beyond just the use of formal experiments.

In the next section we examine how the concepts of *evidence-based* knowledge and of *evidence-informed* decision-making, have been interpreted in the 20th and 21st centuries. In particular, we will discuss the procedures that have been adopted to produce evidence that is of the best possible quality.

1.2 Emergence of the evidence-based movement

It is difficult to discuss the idea of evidence-based thinking without first providing a description of how it emerged in clinical medicine. And in turn, it is difficult to categorise this as other than a movement that has influenced the practice and teaching of medicine (and beyond). At the heart of this lies the *Cochrane Collaboration*[1], named after one of the major figures in its development. This is a not-for-profit body that provides both independent guardianship of evidence-based practices for clinical medicine, and also custodianship of the resulting knowledge.

So, who was Cochrane? Well, Archie Cochrane was a leading clinician, who became increasingly concerned throughout his career about how to know what was the best treatment for his patients. His resulting challenge to the medical profession was to find the most effective and fairest way to evaluate available medical evidence, and he was particularly keen to put value upon evidence that was obtained from randomised controlled trials (RCTs). Cochrane's highly influential 1971 monograph "Effectiveness and Efficiency: Random Reflections on Health Services" (Cochrane 1971) particularly championed the extensive use of randomisation in RCTs, in order to minimise the influence of different sources of potential bias (such as trial design, experimenter conduct, allocation of subjects to groups, etc.). Indeed, he is quoted as saying that "you should randomise until it hurts", in order to emphasise the critical importance of conducting fair and unbiased trials.

Cochrane also realised that even when performed well, individual RCTs could not be relied upon to provide unequivocal results, and indeed, that where RCTs on a given topic were conducted by different groups and in different places, they might well produce apparently conflicting outcomes. From this, he concluded in 1979 that "it is surely a great criticism of our profession that we have not organised a critical summary by speciality or subspeciality, adapted periodically, of all relevant randomised controlled trials".

Conceptually, this statement was at complete variance with accepted scientific practice (not just that in clinical medicine). In particular, the role of the *review paper* has long been well established across much of academia, with

[1] www.cochrane.org

specialist journals dedicated to publishing reviews, and with an invitation to write a review on a given topic often being regarded as a prestigious acknowledgement of the author's academic standing. However, a problem with this practice was (and still is) that two people who are both experts on a given topic might well write reviews that draw contrasting conclusions—and with each of them selecting a quite different set of sources in support of their conclusion.

While this does not mean that an expert review is necessarily of little value, it does raise the question of how far the reviewer's own opinions may have influenced the conclusions. In particular, where the subject-matter of the review requires interpretation of empirical data, then how this is selected is obviously a critical parameter. A widely-quoted example of this is the review by Linus Pauling in his 1970 publication on the benefits of Vitamin C for combatting the common cold. His 'cherry-picking' of those studies that supported his theory, and dismissal of those that did not as being flawed, produced what is now regarded as an invalid conclusion. (This is discussed in rather more depth in Ben Goldacre's book, *Bad Science* (2009), although Goldacre does observe that in fairness, cherry-picking of studies was the norm for such reviews at the time when Pauling was writing—and he also observes that this remains the approach that is still apt to be favoured by the purveyors of 'alternative' therapies.)

Finding the most relevant sources of data is, however, only one element in producing reviews that are objective and unbiased. The process by which the outcomes (findings) from those studies are *synthesised* is also a key parameter to be considered. Ideas about synthesis have quite deep roots—in their book on literature reviews, Booth, Papaioannou and Sutton (2012) trace many of the ideas back to the work of the surgeon James Lind and his studies of how to treat scurvy on ships—including his recognition of the need to discard 'weaker evidence', and to do so by using an objective procedure. However, the widespread synthesis of data from RCTs only really became commonplace in the 1970s, when the term *meta-analysis* also came into common use[2].

Meta-analysis is a statistical procedure used to pool the results from a number of studies, usually RCTs or controlled experiments (we discuss this later in Chapters 9–11). By identifying where individual studies show consistent outcomes, a meta-analysis can provide much greater statistical authority for its outcomes than is possible for individual studies.

Meta-analysis provided one of the key elements in persuading the medical profession to pay attention. In particular, what Goldacre describes as a "landmark meta-analysis" looking at the effectiveness of an intervention given to mothers-to-be who risked premature birth, attracted serious attention. Seven

[2]One of us (DB) can claim to have had relatively early experience of the benefits of synthesis, when analysing scattering data in the field of elementary particle physics (Budgen 1971). Some experiments had suggested the possible presence of a very short-lived Σ particle, but this was conclusively rejected by the analysis based upon the composite dataset from multiple experiments.

trials of this treatment were conducted between 1972 and 1981, two finding positive effects, while the other five were inconclusive. However, in 1989 (a decade later) a meta-analysis that pooled the data from these trials demonstrated very strong evidence in favour of the treatment, and it is a "Forest Plot" of these results that now forms a central part of the logo of the *Cochrane Collaboration*, as shown in Figure 1.3[3]. With analyses such as this, supported by the strong advocacy of evidence-based decision making from David Sackett and his colleagues (Sackett, Straus, Richardson, Rosenberg & Haynes 2000), clinicians became more widely persuaded that such pooling of data could provide significant benefits. And linking all this back with the ideas about evidence, Sackett et al. (2000) defined *Evidence-Based Medicine* (EBM) as "the conscientious, explicit and judicious use of the current best evidence in making decisions about the care of individual patients".

FIGURE 1.3: The logo of the Cochrane Collaboration featuring a forest plot (reproduced by permission of the Cochrane Collaboration).

The concept has subsequently been taken up widely within healthcare, although, as we note in Section 1.4, not always without some opposing arguments being raised. It has also been adopted in other disciplines where empirical data is valued and important, with education providing a good example of a discipline where the outcomes have been used to help determine policy as well as practice. A mirror organisation to that of the Cochrane Collaboration is the Campbell Collaboration[4], that "produces systematic reviews of the effects of social interventions in Crime & Justice, Education, International Development, and Social Welfare". And of course, in the following chapters, we will explore how evidence-based ideas have been adopted in software engineering.

So, having identified two key parameters for producing sound evidence from an objective review process as being:

- objective selection of relevant studies
- systematic synthesis of the outcomes from those studies

[3]We provide a fuller explanation of the form of Forest Plots in Chapter 11. The horizontal bars represent the results from individual trials, with any that are to the left of the centre line favouring the experimental treatment, although only being statistically significant if they do not touch the line. The results of the meta-analysis is shown by the diamond at the bottom.

[4]www.campbellcollaboration.org

we can now move on to discuss the way that this is commonly organised through the procedures of a *systematic review*.

1.3 The systematic review

At this point, we need to clarify a point about the terminology we use in this book. What this section describes is something that is commonly described as a process of *systematic review* (SR). However, in software engineering, a commonly-adopted convention has been to use the term *systematic literature review* (SLR). This was because when secondary studies were first introduced into software engineering, there was concern that they would be confused with code inspection practices (also termed reviews) and so the use of 'literature' was inserted to emphasise that it was published studies that were being reviewed, not code.

Now that secondary studies as a key element of evidence-based software engineering (EBSE) are part of the empirical software engineer's toolbox, the likelihood of confusion seems much less. So we feel that it is more appropriate to use the more conventional term 'systematic review' throughout this book. However, we do mention it here just to emphasise that when reading software engineering papers, including many of our own, a systematic literature review is the same thing as a systematic review.

The goal of a *systematic review* is to search for and identify all relevant material related to a given topic (with the nature of this material being determined by the underlying question and the nature of the stakeholders who have an interest in it). Knowledge about that topic is then used to assist with drawing together the material in order to produce a collective result. The aim is for the procedures followed in performing the review to be as objective, analytical, and repeatable as possible—and that this process should, in the ideal, be such that if the review were repeated by others, it would select the same input studies and come to the same conclusions. We often refer to a systematic review as being a *secondary study*, because it generates its outcomes by aggregating the material from a set of *primary studies*.

Not surprisingly, conducting such a review is quite a large task, not least because the 'contextual knowledge' required means that much of it needs to be done by people with some knowledge of the topic being reviewed. We will encounter a number of factors that limit the extent to which we can meet these goals for a review as we progress through the rest of this part of the book. However, the procedures followed in a systematic review are intended to minimise the effects of these factors and so even when we don't quite meet the aim as fully as we would like, the result should still be a good quality review. (This is not to say that expert reviews are not necessarily of good quality, but

they are apt to lack the means of demonstrating that this is so, in contrast to a systematic review.)

So, a key characteristic of a systematic review is that it is just that, *systematic*, and that it is conducted by following a set of well-defined procedures. These are usually specified as part of the *Review Protocol*, which we will be discussing in more detail later, in Chapter 4. For this section, we are concerned simply with identifying what it is that these procedures need to address. Figure 1.4 illustrates how the main elements of a systematic review are related once a sensible question has been chosen. Each of the ovals represents one of the processes that needs to be performed by following a pre-defined procedure. Each process also involves making a number of decisions, as outlined below.

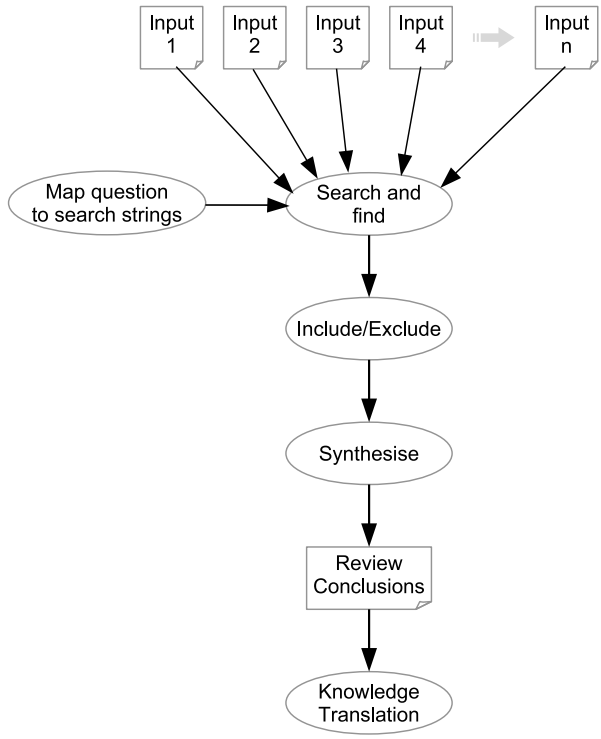

FIGURE 1.4: The systematic review process.

What searching strategy will be used? An important element of the review is to make clear *where* we will search, and *how* we will search for appropriate review material. In addition, we need to ensure that we have included all the different keywords and concepts that might be relevant. We address this in detail in Chapter 5.

What material is eligible for inclusion? This relates to both the different *forms* in which material (usually in the form of the outcomes of empirical studies) might occur, and also any characteristics that might affect its *quality*. Indeed, we often have more detailed specifications for what is to be *excluded* than for what is to be included, since we want to ensure that we don't miss anything that could be in a form that we didn't anticipate, or expect to encounter. Again, these issues will be considered more fully in Chapters 6 and 7.

How is the material to be synthesised? This addresses the analytical procedures that are to be followed. These may be fairly simple, as we explain below, or quite complex. Chapters 9, 10 and 11 consider the relevant issues for a software engineering context.

How to interpret the outcomes of the review? This is not necessarily a single process, since the outcomes might need to be interpreted differently when used in specific contexts. The processes involved are termed *Knowledge Translation* (KT), and are still the topic of extensive discussion in domains where evidence-based practices are much more established than they are in software engineering. However, in Chapter 14, we do examine how KT can be applied in a software engineering context.

The point to emphasise though, is that all of these activities involve *procedures* that need to be applied and interpreted by human beings, with many of them also needing knowledge about the topic of the review. While tools can help with managing the process, the individual decisions still need to be made by a human analyst. In particular, because there will almost certainly be a wide variation of potential inputs to a review, it is possible that some of these will be interpreted differently by different people. To minimise the effects of this, systematic reviews are often conducted by two (or even more) people, who compare results at each stage, and then seek to resolve any differences (again in a systematic manner).

As indicated, because systematic reviews have different forms, the process of synthesis can also take many forms. (A very good categorisation of the wide range of forms of synthesis used across those disciplines that employ systematic reviews is provided in the book by Booth et al. (2012).) At its most simple, synthesis can consist mainly of classification of the material found, identifying where there are groups of studies addressing a particular issue, or equally, where there is a lack of studies. We term this a *mapping study*, and software engineering research has made quite extensive use of this form. A value of a mapping study lies partly in identifying where there is scope to perform a fuller review (the groups of related studies), and also where there is a need for more primary studies (the gaps). At the other extreme, where the material consists mostly of RCTs, or good quality experiments, synthesis may be organised in the form of a statistical meta-analysis. Meta-analyses do exist in the software engineering literature, but only in small numbers. Most software engineering

studies use less rigorous forms (and sometimes forms that are less rigorous than could actually be used), and again, we will examine this in much more detail in Chapters 9, 10 and 11.

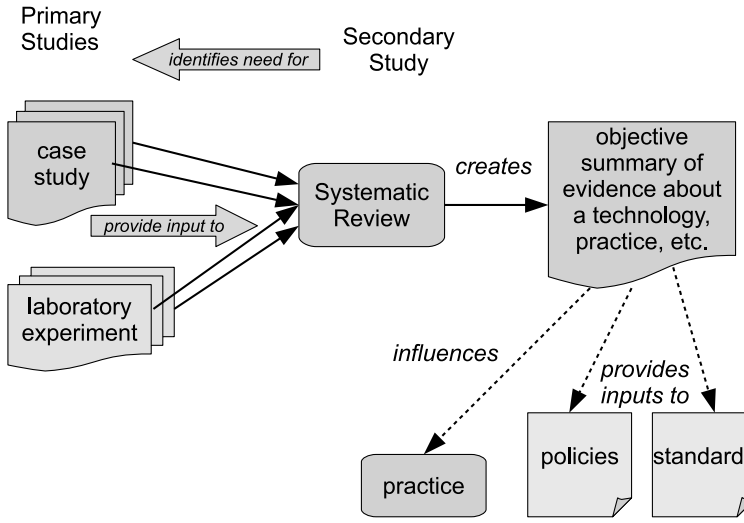

FIGURE 1.5: The context for a systematic review.

Figure 1.5 illustrates the wider context for a systematic review. So far we have mainly described the things that affect a review, but as we can see, the review itself also has some quite important roles. One of these is in providing a context for primary studies. Until the adoption of the evidence-based paradigm, these were mostly viewed as essentially being isolated studies that formed 'islands' of knowledge. When primary studies are viewed in terms of their role as inputs to a systematic review, there are two new factors that may influence the way that they are organised. One is the choice of topic—perhaps because a review has identified the need for further studies. The other is the way that primary studies report their results—one of the frequent complaints from analysts who conduct a systematic review is that important information is apt to be omitted from papers and reports. So designing and reporting of primary studies now needs to be more influenced by this role as an input to a secondary study than was the case in the past. Reviews also influence policies, standards and decisions about practice—and while this is still less likely to be the case in software engineering than in disciplines such as education and clinical medicine, consideration of these aspects should increasingly be a goal when performing systematic reviews.

The systematic review is the main instrument used for evidence-based studies and so will be discussed in depth through most of this book, and certainly in the rest of Part I. So, to conclude this introductory chapter, we

need to consider some of its limitations too. This is because an appreciation of these is really needed when designing and conducting reviews as well as when seeking to understand what the outcomes of a review might mean to us.

1.4 Some limitations of an evidence-based view of the world

Not surprisingly, there has been a growing tendency for researchers, at least, to consider that knowledge that has been derived from an evidence-based process must inevitably be better than 'expert' knowledge that has been derived, albeit less systematically, from experience. And as the preceding sections indicate, we would to some degree support such a view, although replacing "inevitably" with the caveat "depending upon circumstances".

In clinical medicine and in wider healthcare, it has been argued that evidence-based research practices have become the "new orthodoxy", and that there are dangers in blind acceptance of the outcomes from this. Some of the arguments for this position are set out in a paper by Hammersley (2005). In particular, he questions whether professional practice can be wholly based on research evidence, as opposed to informed by it, noting that research findings do themselves rely upon judgement and interpretation. While many of the arguments focus upon how to interpret outcomes for practice, rather than upon the research method itself, the appropriateness of this form of research for specific topics does need to be considered. Even for systematic reviews, the two well known adages of "to a person with a hammer everything looks like a nail" and "garbage in–garbage out" may sometimes be apt.

So here we suggest some factors that need to be kept in mind when reading the following chapters. They are in every sense 'limitations', in that they do not necessarily invalidate specific evidence-based studies, but they might well limit the extent to which we can place full confidence in the outcomes of a systematic review.

A systematic review is conducted by people. There is inevitably an element of *interpretation* in the main activities of a systematic review: performing searches; deciding about inclusion and exclusion; and making various decisions during synthesis. All of these contain some potential for introducing *bias* into the outcomes. The practice of using more than one analyst can help with constraining the degree of variability that might arise when performing these tasks, but even then, two analysts who have the same sort of background might arrive at a set of joint decisions about which primary studies to include that would be different from those that would be made by two analysts who come from different

backgrounds. Both the selection of studies, and also the decisions made in synthesis, can affect the outcomes of a review.

The outcomes depend upon the primary studies. The *quality* of the primary studies that underpin a systematic review can vary quite considerably. A review based upon a few relatively weak primary studies is hardly likely to be definitive.

Not all topics lend themselves well to empirical studies. To be more specific, the type of empirical study that is appropriate to some topics may well offer poorer scope for using strong forms of synthesis than occur (say) when using randomised controlled experiments. We will examine this more fully in Part II.

All of these are factors that we also need to consider when planning to perform a systematic review. And in the same way that a report of a primary study will usually make an assessment of the limitations upon its conclusions imposed by the relevant "threats to validity" (we discuss this concept further later), so a report of the outcomes from a systematic review needs to do the same. Such an assessment can then help the reader to determine how fully they can depend upon the outcomes and also how limited or otherwise the scope of these is likely to be.

In the next chapter we go on to look at the way that systematic reviews are performed in software engineering, and so we also look at some of these issues in rather more detail and within a computing context.

This page intentionally left blank

Bibliography

AGREE (2009), 'Appraisal of Guidelines for Research and Evaluation II (AGREE II)', AGREE Next Steps Consortium Report.

Ali, M. S., Babar, M. A., Chen, L. & Stol, K.-J. (2010), 'A systematic review of comparative evidence of aspect-oriented programming', *Information and Software Technology* **52**(9), 871–887.

Ali, N. B., Peterson, K. & Wohlin, C. (2014), 'A systematic literature review on the industrial use of software process simulation', *Journal of Systems & Software* **97**, 65–85.

Alves, V., Niu, N., Alves, C. & Valença, G. (2010), 'Requirements engineering for software product lines: A systematic literature review', *Information and Software Technology* **52**(8), 806–820.

Ampatzoglou, A. & Stamelos, I. (2010), 'Software engineering research for computer games: A systematic review', *Information and Software Technology* **52**(9), 888–901.

Anjum, M. & Budgen, D. (2012), A mapping study of the definitions used for Service Oriented Architecture, *in* 'Proceedings of 16th EASE Conference', IET Press, pp. 1–5.

Arias, T. B. C., van der Spek, P. & Avgeriou, P. (2011), 'A practice-driven systematic review of dependency analysis solutions', *Empirical Software Engineering* **16**, 544–586.

Atkins, S., Lewin, S., Smith, H., Engel, M., Fretheim, A. & Volmink, J. (2008), 'Conducting a meta-ethnography of qualitative literature: Lessons learnt', *BMC Medical Research Methodology* **8**(21).

Azfal, W., Torkar, R. & Feldt, R. (2009), 'A systematic review of search-based testing for non-functional system properties', *Information and Software Technology* **51**, 957–976.

Babar, M. A. & Zhang, H. (2009), Systematic literature reviews in software engineering: Preliminary results from interviews with researchers, *in* 'Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement', ESEM '09, IEEE Computer Society, Washington, DC, USA, pp. 346–355.

Barends, E. G. R. & Briner, R. B. (2014), 'Teaching evidence-based practice: Lessons from the pioneers—An interview with Amanda Burls and Gordon Guyatt', *Academy of Management Learning & Education* **13**(3), 476–483.

Barnett-Page, E. & Thomas, J. (2009), 'Methods for the synthesis of qualitative research: a critical review', *BMC Medical Research Methodology* **9**(59).

Basili, V., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sorumgard, S. & Zelkowitz, M. (1996), 'The empirical investigation of perspective-based reading', *Empirical Software Engineering* **1**(2), 133–164.

Basili, V. R., Shull, F. & Lanubile, F. (1999), 'Building Knowledge through Families of Experiments', *IEEE Transactions on Software Engineering* **25**(4), 456–473.

Beecham, S., Baddoo, N., Hall, T., Robinson, H. & Sharp, H. (2006), *Protocol for a Systematic Literature Review of Motivation in Software Engineering*, University of Hertfordshire.

Beecham, S., Baddoo, N., Hall, T., Robinson, H. & Sharp, H. (2008), 'Motivation in software engineering: A systematic literature review', *Information and Software Technology* **50**(9–10), 860–878.

Benbasat, I., Goldstein, D. K. & Mead, M. (1987), 'The case research strategy in studies of information systems', *MIS Quarterly* **11**(3), 369–386.

Boehm, B. W. (1981), *Software Engineering Economics*, Prentice-Hall.

Booth, A., Papaioannou, D. & Sutton, A. (2012), *Systematic Approaches to a Successful Literature Review*, Sage Publications Ltd.

Borenstein, M., Hedges, L. V., Higgins, J. P. T. & Rothstein, H. T. (2009), *Introduction to Meta-Analysis*, John Wiley and Sons Ltd.

Bowes, D., Hall, T. & Beecham, S. (2012), Slurp: A tool to help large complex systematic reviews deliver valid and rigorous results, in '*Proceedings 2nd International Workshop on Evidential Assessment of Software Technologies (EAST'12)*', ACM Press, pp. 33–36.

Bratthall, L. & Jørgensen, M. (2002), 'Can you trust a single data source exploratory software engineering case study?', *Empirical Software Engineering* **7**(1), 9–26.

Briand, L. C., Melo, W. L. & Wust, J. (2002), 'Assessing the applicability of object-oriented software projects fault-proneness models across', *IEEE Transactions on Software Engineering* **28**, 706–720.

Brooks Jr., F. P. (1987), 'No silver bullet: essences and accidents of software engineering', *IEEE Computer* **20**(4), 10–19.

Budgen, D. (1971), 'A $KN \to \Lambda\pi$ partial-wave analysis in the region of the $\Sigma(1670)$', *Lettere al Nuovo Cimento* **2**(3), 85–89.

Budgen, D., Burn, A., Brereton, P., Kitchenham, B. & Pretorius, R. (2011), 'Empirical evidence about the UML: A systematic literature review', *Software — Practice and Experience* **41**(4), 363–392.

Budgen, D., Burn, A. & Kitchenham, B. (2011), 'Reporting student projects through structured abstracts: A quasi-experiment', *Empirical Software Engineering* **16**(2), 244–277.

Budgen, D., Drummond, S., Brereton, P. & Holland, N. (2012), What scope is there for adopting evidence-informed teaching in software engineering?, in 'Proceedings of 34th International Conference on Software Engineering (ICSE 2012)', IEEE Computer Society Press, pp. 1205–1214.

Budgen, D., Kitchenham, B. A., Charters, S., Turner, M., Brereton, P. & Linkman, S. (2008), 'Presenting software engineering results using structured abstracts: A randomised experiment', *Empirical Software Engineering* **13**(4), 435–468.

Budgen, D., Kitchenham, B. & Brereton, P. (2013), The Case for Knowledge Translation, *in* 'Proceedings of 2013 International Symposium on Empirical Software Engineering & Measurement', IEEE Computer Society Press, pp. 263–266.

Budgen, D., Kitchenham, B., Charters, S., Gibbs, S., Pohthong, A., Keung, J. & Brereton, P. (2013), Lessons from conducting a distributed quasi-experiment, *in* 'Proceedings of 2013 International Symposium on Empirical Software Engineering & Measurement', IEEE Computer Society Press, pp. 143–152.

Burgers, J. S., Grol, R., Klazinga, N. S., Mäkelä, M. & Zaat, J. (2003), 'Towards evidence-based clinical practice: an international survey of 18 clinical guidelines programs', *International Journal for Quality in Health Care* **15**(1), 31–45.

Burrows, R., Garcia, A. & Taïani, F. (2009), Coupling metrics for aspect-oriented programming: A systematic review of maintainability studies, *in* 'ENASE 2009 - Proceedings of the 4th International Conference on Evaluation of Novel Approaches to Software Engineering, Milan, Italy, May 2009', pp. 191–202.

Canfora, G., Cimitile, A., Garcia, F., Piattini, M. & Visaggio, C. A. (2007), 'Evaluating performances of pair designing in industry', *Journal of Systems & Software* **80**, 1317–1327.

Carifio, J. & Perla, R. J. (2007), 'Ten common misunderstandings, misconceptions, persistent myths and urban legends about likert scales and

likert response formats and their antidotes', *Journal of Social Science* **3**(3), 106–116.

Carver, J. C. (2010), Towards reporting guidelines for experimental replications: A proposal, *in* 'Proceedings of the 1st International Workshop on Replication in Empirical Software Engineering Research (RESER 2010)', ACM Press, pp. 1–4.

Casey, V. & Richardson, I. (2008), The impact of fear on the operation of virtual teams, *in* 'Proceedings of IEEE International Conference on Global Software Engineering', IEEE Computer Society Press.

Chen, L. & Babar, M. A. (2011), 'A systematic review of evaluation of variability management approaches in software product lines', *Information and Software Technology* **53**(4), 344–362. Special section: Software Engineering track of the 24th Annual Symposium on Applied Computing Software Engineering track of the 24th Annual Symposium on Applied Computing.

Ciolkowski, M. (2009), What do we know about perspective-based reading? an approach for quantitative aggregation in software engineering, *in* 'Proceedings 3rd International Symposium on Empirical Software Engineering & Measurement (ESEM)', pp. 133–144.

Cochran, W. (1954), 'The combination of estimates from different experiments.', *Biometrics* **10**(101-129).

Cochrane, A. L. (1971), *Effectiveness and Efficiency: Random Reflections on Health Services*, The Nuffield Provincial Hospitals Trust.

Cohen, J. (1960), 'A coefficient of agreement for nominal scales', *Educational and Psychological Measurement* **20**(1), 37–46.

CRD (2009), 'Systematic reviews crd's guidance for undertaking reviews in health care', Centre for Review and Dissemination.

Cruzes, D. S. & Dybå, T. (2011*a*), Recommended steps for thematic synthesis in software engineering, *in* 'Proceedings ESEM 2011'.

Cruzes, D. S. & Dybå, T. (2011*b*), 'Research synthesis in software engineering: A tertiary study', *Information and Software Technology* **53**(5), 440–455.

Cruzes, D. S., Dybå, T., Runeson, P. & Höst, M. (2014), 'Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example', *Empirical Software Engineering* .

Cruzes, D. S., Mendonca, M., Basili, V., Shull, F. & Jino, M. (2007*a*), Automated information extraction from empirical software engineering literature, in 'Proceedings of First International Symposium on Empirical Software Engineering & Measurement (ESEM 2007)', pp. 491–493.

Cruzes, D. S., Mendonca, M., Basili, V., Shull, F. & Jino, M. (2007*b*), Using context distance measurement to analyze results across studies, *in* 'Proceedings of First International Symposium on Empirical Software Engineering & Measurement (ESEM 2007)', pp. 235–244.

Cumming, G. (2012), *Understanding the New Statistics. Effect Sizes, Confidence Intervals, and Meta-Analysis*, Routledge Taylor & Francis Group, New York, London.

Da Silva, F. Q. B.; Cruz, S. S. J. O.; Gouveia, T. B.; & Capretz, L. F. (2013), 'Using meta-ethnography to synthesize research: A worked example of the relations between personality and software team process', *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement,* pp. 153–162 .

da Silva, F. Q., Santos, A. L., Soares, S., França, A. C. C., Monteiro, C. V. & Maciel, F. F. (2011), 'Six years of systematic literature reviews in software engineering: An updated tertiary study', *Information and Software Technology* **53**(9), 899–913.

da Silva, F. Q., Suassuna, M., França, A. C. C., Grubb, A. M., Gouveia, T. B., Monteiro, C. V. & dos Santos, I. E. (2014), 'Replication of empirical studies in software engineering research: A systematic mapping study', *Empirical Software Engineering* **19**, 501–557.

Davis, D., Evans, M., Jadad, A., Perrier, L., Rath, D., Ryan, D., Sibbald, G., Straus, S., Rappolt, S., Wowk, M. & Zwarenstein, M. (2003), 'The case for knowledge translation: shortening the journey from evidence to effect', *BMJ* **327**, 33–35.

Díaz, J., Pérez, J., Alarcón, P. P. & Garbajosa, J. (2011), 'Agile product line engineering—A systematic literature review', *Software: Practice and Experience* **41**, 921–941.

Dickinson, T. L. & McIntyre, R. M. (1997), A conceptual framework of teamwork measurement, *in* M. T. Brannick, E. Salas & C. Prince, eds, *Team Performance Assessment and Measurement: Theory, Methods and Applications*, Psychology Press, NJ, USA, pp. 19–43.

Dieste, O., Grimán, A. & Juristo, N. (2009), 'Developing search strategies for detecting relevant experiments', *Empirical Software Engineering* **14**(5), 513–539.

Dieste, O., Griman, A., Juristo, N. & Saxena, H. (2011), Quantitative determination of the relationship between internal validity and bias in software engineering experiments: Consequences for systematic literature reviews, *in* 'Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on', pp. 285–294.

Dieste, O. & Juristo, N. (2011), 'Systematic review and aggregation of empirical studies on elicitation techniques', *IEEE Transactions on Software Engineering* **37**(2), 283–304.

Dieste, O., Juristo, N. & Martinez, M. D. (2014), Software industry experiments: A systematic literature review, in *'Proceedings of the 1st International Workshop on Conducting Empirical Studies in Industry (CSEI'14)'*, IEEE Computer Society Press, pp. 2–8.

Dieste, O. & Padua, O. (2007), Developing search strategies for detecting relevant experiments for systematic reviews, in *'Empirical Software Engineering and Measurement, 2007. ESEM 2007.* First International Symposium on', pp. 215–224.

Dixon-Woods, M., Sutton, A., Shaw, R., Miller, T., Smith, J., Young, B., Bonas, S., Booth, A. & Jones, D. (2007), 'Appraising qualitative research for inclusion in systematic reviews: a quantitative and qualitative comparison of three methods', *Journal of Health Services Research and Policy* **12**(1), 42–47.

Dybå, T. & Dingsøyr, T. (2008*a*), 'Empirical studies of agile software development: A systematic review', *Information & Software Technology* **50**, 833–859.

Dybå, T. & Dingsøyr, T. (2008*b*), Strength of evidence in systematic reviews in software engineering, in *'Proceedings of International Symposium on Empirical Software Engineering and Metrics (ESEM)'*, pp. 178–187.

Dybå, T., Kampenes, V. & Sjøberg, D. (2006), 'A systematic review of statistical power in software engineering experiments', *Information & Software Technology* **48**(8), 745–755.

Eaves, Y. D. (2001), 'A synthesis technique for grounded theory data analysis', *Journal of Advanced Nursing* **35**(5), 654–663.

Eisenhardt, K. M. (1989), 'Building theories from case study research', *Academy of Management Review* **14**, 532–550.

Elamin, M. B., Flynn, D. N., Bassler, D., Briel, M., Alonso-Coello, P., Karanicolas, P. J., Guyatt, G., Malaga, G., Furukawa, T. A., Kunz, R., Schnemann, H., Murad, M. H., Barbui, C., Cipriani, A. & Montori, V. M. (2009), 'Choice of data extraction tools for systematic reviews depends on resources and review complexity', *Journal of Clinical Epidemiology* **62**(5), 506–510.

Elberzhager, F., Rosbach, A., Münch, J. & Eschbach, R. (2012), 'Reducing test effort: A systematic mapping study on existing approaches', *Information and Software Technology* **54**, 1092–1106.

Engström, E., Runeson, P. & Skoglund, M. (2010), 'A systematic review on regression test selection techniques', *Information and Software Technology* **52**, 14–30.

Ericsson, K. & Simon, H. (1993), *Protocol Analysis: Verbal Reports as Data*, MIT Press.

Felizardo, K. R., Andery, G. F., Paulovich, F. V., Minghim, R. & Maldonado, J. C. (2012), 'A visual analysis approach to validate the selection review of primary studies in systematic reviews', *Information and Software Technology* **54**(10), 1079 – 1091.

Felizardo, K. R., Nakagawa, E. Y., Feitosa, D., Minghim, R. & Maldonado, J. C. (2010), An approach based on visual text mining to support categorization and classification in the systematic mapping, *in* 'Proceedings EASE '10', British Computer Society.

Fenton, N. E. & Pfleeger, S. L. (1997), *Software Metrics: A Rigorous and Practical Approach*, 2nd edn, PWS Publishing.

Fernández-Sáez, A. M., Bocco, M. G. & Romero, F. P. (2010), SLR-Tool—a tool for performing systematic literature reviews, *in* J. A. M. Cordeiro, M. Virvou & B. Shishkov, eds, 'Proceedings of ICSOFT (2)', SciTePress, pp. 157–166.

Fichman, R. G. & Kemerer, C. F. (1997), 'Object technology and reuse: Lessons from early adopters', *IEEE Computer* **30**, 47–59. (Reports on four Case Studies).

Fink, A. (2003), *The Survey Handbook*, 2 edn, Sage Books. Volume 1 of the Survey Kit.

Foss, T., Stensrud, E., Myrtveit, I. & Kitchenham, B. (2003), 'A simulation study of the model evaluation criterion MMRE', *IEEE Transactions on Software Engineering* **29**(11), 985–995.

Galin, D. & Avrahami, M. (2006), 'Are CMM program investments beneficial? Analysing past studies', *IEEE Software* pp. 81–87.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Ghapanchi, A. H. & Aurum, A. (2011), 'Antecedents to IT personnel's intentions to leave: A systematic literature review', *Journal of Systems & Software* **84**, 238–249.

Giuffrida, R. & Dittrich, Y. (2013), 'Empirical studies on the use of social software in global software development–A systematic mapping study', *Information & Software Technology* **55**, 1143–1164.

Glaser, B. & Strauss, A. (1967), *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Publishing Company.

Glass, R., Ramesh, V. & Vessey, I. (2004), 'An Analysis of Research in Computing Disciplines', *Communications of the ACM* **47**, 89–94.

Goldacre, B. (2009), *Bad Science*, Harper Perennial.

Gómez, O. S., Juristo, N. & Vegas, S. (2010), Replications types in experimental disciplines, *in* 'Proceedings of Empirical Software Engineering & Measurement (ESEM)', pp. 1–10.

Gorschek, T., Svahnberg, M., Borg, A., Loconsole, A., Börstler, J., Sandahl, K. & Eriksson, M. (2007), 'A controlled empirical evaluation of a requirements abstraction model', *Information & Software Technology* **49**(7), 790–805.

Gotterbarn, D. (1999), 'How the new Software Engineering Code of Ethics affects you', *IEEE Software* **16**(6), 58–64.

GRADE Working Group (2004), 'Grading quality of evidence and strength of recommendations', *British Medical Journal (BMJ)* **328**(1490).

Graham, I. D., Logan, J., Harrison, M. B., Straus, S. E., Tetroe, J., Caswell, W. & Robinson, N. (2006), 'Lost in knowledge translation: Time for a map?', *Journal of Continuing Education in the Health Professions* **26**(1), 13–24.

Greenhalgh, T. (2010), *How to read a paper The basics of evidence-based medicine*, 4th edn, Wiley-Blackwell BMJIBooks.

Greenhalgh, T., Robert, G., MacFarlane, F., Bate, P. & Kyriakidou, O. (2004), 'Diffusion of Innovations in Service Organisations: Systematic Review and Recommendations', *The Milbank Quarterly* **82**(4), 581–629.

Greenhalgh, T. & Wieringa, S. (2013), 'Is it time to drop the "knowledge translation" metaphor? a critical literature review', *Journal of the Royal Society of Medicine* **104**, 501–509.

Grimstad, S. & Jørgensen, M. (2007), 'Inconsistency of expert judgment-based estimates of software development effort', *Journal of Systems & Software* **80**, 1770–1777.

Grimstad, S., Jørgensen, M. & Moløkken-Østvold, K. (2005), The clients' impact on effort estimation accuracy in software development projects, in *'Proceedings of 11th IEEE International Software Metrics Symposium (METRICS 2005)'*, pp. 1–10.

Gu, Q. & Lago, P. (2009), 'Exploring service-oriented system engineering challenges: a systematic literature review', *Service Oriented Computing and Applications* **3**(3), 171–188.

Guyatt, G. H., Oxman, A. D., Vist, G. E., Kunz, R., Falck-Ytter, Y., Alonso-Coello, P. & Schünemann, H. J. (2008), 'Grade: an emerging consensus on rating quality of evidence and strength of recommendations', *British Medical Journal* **336**, 924–926.

Guzmán, L., Lampasona, C., Seaman, C. & Rombach, D. (2014), Survey on research synthesis in software engineering, in *'18th International Conference on Eavluation and Assessment in Software Engineering'*, ACM, New York, USA.

Hall, T., Beecham, S., Bowes, D., Gray, D. & Counsell, S. (2012), 'A systematic literature review on fault prediction performance in software engineering', *IEEE Transactions on Software Engineering* **38**(6), 1276–1304.

Hammersley, M. (2005), 'Is the evidence-based practice movement doing more good than harm? Reflections on Iain Chalmers' case for research-based policy making and practice', *Evidence & Policy* **1**(1), 85–100.

Hammersley, M. & Atkinson, P. (1983), *Ethnography, Principles in Practice*, Tavistock.

Hannay, J., Dybå, T., Arisholm, E. & Sjøberg, D. (2009), 'The effectiveness of pair programming. A meta analysis', *Information & Software Technology* **51**(7), 1110–1122.

Hannes, K., Lockwood, C. & Pearson, A. (2010), 'A comparative analysis of three online appraisal instruments' ability to assess validity in qualitative research', *Qualitative Health Research* **20**(12), 1736– 1743.

Hanssen, G. K., Bjørnson, F. O. & Westerheim, H. (2007), Tailoring and introduction of the rational unified process, in *'Software Process Improvement (EuroSPI 2007)'*, Vol. LNCS 4764/2007, Springer, pp. 7–18.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning Data Mining, Inference, Prediction*, 2nd edn, Springer.

Haugset, B. & Hanssen, G. K. (2008), Automated acceptance testing: A literature review and an industral case study, in *'Proceedings of Agile 2008'*, IEEE Computer Society Press, pp. 27–38.

Hedges, L. V. & Olkin, I. (1985), *Statistical Methods for Meta-Analysis*, Academic Press.

Hernandes, E., Zamboni, A., Fabbri, S. & Thommazo, A. A. D. (2012), 'Using GQM and TAM to evaluate StArt–a tool that supports systematic review', *CLEI Electronic Journal* **15**, 3.

Higgins, J. P. T. & Thompson, S. G. (2002), 'Quantifying heterogeneity in a meta-analysis.', *Statistics in Medicine* **21**(11), 1539–1558.

Higgins, J. P. T., Thompson, S. G., Deeks, J. J. & Altman, D. G. (2003), 'Measuring inconsistency in meta-analyses', *BMJ* **327**(7414), 557–560.

Hordijk, W., Ponisio, M. L. & Wieringa, R. (2009), Harmfulness of code duplication–a structured review of the evidence, in *'Proceedings of 13th International Conference on Evaluation and Assessment in Software Engineering (EASE 2009)'*, pp. 1–10.

Hossain, E., Babar, M. A. & Paik, H. (2009), Using scrum in global software development: A systematic literature review, in *'Proceedings of 4th International Conference on Global Software Engineering'*, IEEE Computer Society Press, pp. 175–184.

IEEE-CS/ACM (1999), ACM/IEEE-CS software Engineering Code of Ethics and Professional Practice. (Version 5.2).
URL: *http://www.acm.org/about/se-code/*

Jalali, S. & Wohlin, C. (2012), Systematic literature studies: Database searches vs. backward snowballing, in *'Empirical Software Engineering and Measurement (ESEM)*, 2012 ACM-IEEE International Symposium on', pp. 29–38.

Jedlitschka, A. & Pfahl, D. (2005), Reporting guidelines for controlled experiments in software engineering, in *'Proc. ACM/IEEE International Symposium on Empirical Software Engineering (ISESE) 2005'*, IEEE Computer Society Press, pp. 95–195.

Jiménez, M., Piattini, M. & Vizcaíno, A. (2009), Challenges and improvements in distributed software development: A systematic review, in *'Advances in Software Engineering'*, pp. 1–14.

Jørgensen, M. (2004), 'A review of studies on expert estimation of software development effort', *Journal of Systems & Software* **70**(1–2), 37–60.

Jørgensen, M. (2005), 'Evidence-based guidelines for assessment of software development cost uncertainty', *IEEE Transactions on Software Engineering* **31**(11), 942–954.

Jørgensen, M. (2007), 'Forecasting of software development work effort: Evidence on expert judgement and formal models', *Int. Journal of Forecasting* **23**(3), 449–462.

Jørgensen, M. (2014*a*), 'Failure factors of small software projects at a global outsourcingmarketplace', *The Journal of Systems and Software* **92**, 157–169.

Jørgensen, M. (2014*b*), 'What we do and don't know about software development effort estimation', *IEEE Software* pp. 37–40.

Jorgensen, M. & Shepperd, M. (2007), 'A systematic review of software development cost estimation studies', *Software Engineering, IEEE Transactions on* **33**(1), 33–53.

Juristo, N., Moreno, A. M., Vegas, S. & Solari, M. (2006), 'In search of what we experimentally know about unit testing', *IEEE Software* pp. 72–80.

Juristo, N. & Vegas, S. (2011), 'The role of non-exact replications in software engineering experiments', *Empirical Software Engineering* **16**, 295–324.

Kabacoff, R. I. (2011), *R in Action*, Manning.

Kakarla, S., Momotaz, S. & Namim, A. (2011), An evaluation of mutation and data-flow testing: A meta analysis, in *'Fourth International Conference on Software Testing, Verification and Validation Workshops'*, ICSTW, IEEE Computer Society, Washington, DC, USA, pp. 366–375.

Kalinowski, M., Travassos, G. H. & Card, D. N. (2008), Towards a defect prevention based process improvement approach, in *'Proceedings of the 34th Euromicro Conference on Software Engineering and Advanced Applications'*, IEEE Computer Society Press, pp. 199–206.

Kampenes, V. B., Dybå, T., Hannay, J. E. & K. Sjøberg, D. I. (2009), 'A systematic review of quasi-experiments in software engineering', *Inf. Softw. Technol.* **51**(1), 71–82.

Kampenes, V. B., Dybå, T., Hannay, J. E. & Sjøberg, D. I. K. (2007), 'Systematic review: A systematic review of effect size in software engineering experiments', *Information and Software Technology* **49**(11-12), 1073–1086. **URL:** *http://dx.doi.org/10.1016/j.infsof.2007.02.015*

Kasoju, A., Peterson, K. & Mäntylä, M. (2013), 'Analyzing an automotive testing process with evidence-based software engineering', *Information & Software Technology* **55**, 1237–1259.

Kearney, M. H. (1998), 'Ready-to-wear: Discovering grounded formal theory', *Research in Nursing and Health* **21**(2), 179–186.

Khan, K. S., Kunz, R., Kleijnen, J. & Antes, G. (2011), *Systematic Reviews to Support Evidence-Based Medicine*, 2nd edn, Hodder Arnold.

Kitchenham, B. (2008), 'The role of replications in empirical software engineering—a word of warning', *Empirical Software Engineering* **13**, 219–221.

Kitchenham, B. A., Budgen, D. & Brereton, O. P. (2011), 'Using mapping studies as the basis for further research—a participant-observer case study', *Information & Software Technology* **53**(6), 638–651. Special section from EASE 2010.

Kitchenham, B. A., Fry, J. & Linkman, S. (2003), The case against cross-over designs in software engineering, in *'Proceedings of Eleventh Annual International Workshop on Software Technology & Engineering (STEP 2003)'*, IEEE Computer Society Press, pp. 65–67.

Kitchenham, B. A., Li, Z. & Burn, A. (2011), Validating search processes in systematic literature reviews, in *'Proceeding of the 1st International Workshop on Evidential Assessment of Software Technologies'*, pp. 3–9.

Kitchenham, B. A. & Pfleeger, S. L. (2002*a*), 'Principles of survey research part 2: Designing a survey', *ACM Software Engineering Notes* **21**(1), 18–20. (For Part 1, see under Pfleeger).

Kitchenham, B. A. & Pfleeger, S. L. (2002*b*), 'Principles of survey research part 4: Questionnaire evaluation', *ACM Software Engineering Notes* **27**(3), 20–23.

Kitchenham, B. A. & Pfleeger, S. L. (2008), Personal opinion surveys, in F. Shull, J. Singer & D. I. Sjøberg, eds, *'Guide to Advanced Empirical Software Engineering'*, Springer-Verlag London, chapter 3.

Kitchenham, B. & Brereton, P. (2013), 'A systematic review of systematic review process research in software engineering', *Information and Software Technology* **55**(12), 2049–2075.

Kitchenham, B., Brereton, P. & Budgen, D. (2010), The educational value of mapping studies of software engineering literature, in *'Proceedings ICSE'10'*, ACM.

Kitchenham, B., Brereton, P. & Budgen, D. (2012), Mapping study completeness and reliability—a case study, in *'Proceedings of 16th EASE Conference'*, IET Press, pp. 1–10.

Kitchenham, B., Brereton, P., Budgen, D., Turner, M., Bailey, J. & Linkman, S. (2009), 'Systematic literature reviews in software engineering — a systematic literature review', *Information & Software Technology* **51**(1), 7–15.

Kitchenham, B., Burn, A. & Li, Z. (2009), A quality checklist for technology-centred testing studies, in *'Proceedings EASE '09'*.

Kitchenham, B. & Charters, S. (2007), Guidelines for performing systematic literature reviews in software engineering, Technical report, Keele University and Durham University Joint Report.

Kitchenham, B., Dybå, T. & Jørgensen, M. (2004), Evidence-based software engineering, in *'Proceedings of ICSE 2004'*, IEEE Computer Society Press, pp. 273–281.

Kitchenham, B., Mendes, E. & Travassos, G. H. (2007), 'Cross versus within-company cost estimation studies: A systematic review', *IEEE Transactions on Software Engineering* **33**(5), 316–329.

Kitchenham, B., Pfleeger, S. L., McColl, B. & Eagan, S. (2002), 'An empirical study of maintenance and development estimation accuracy', *Journal of Systems and Software* **64**, 57–77.

Kitchenham, B., Pfleeger, S. L., Pickard, L., Jones, P., Hoaglin, D., Emam, K. E. & J.Rosenberg (2002), 'Preliminary Guidelines for Empirical Research in Software Engineering', *IEEE Transactions on Software Engineering* **28**, 721–734.

Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., El Emam, K. & Rosenberg, J. (2002), 'Preliminary guidelines for empirical research in software engineering', *IEEE Transactions on Software Engineering* **28**(8), 721–734.

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, P., Turner, M., Niazi, M. & Linkman, S. (2010), 'Systematic literature reviews in software engineering — a tertiary study', *Information & Software Technology* **52**, 792–805.

Kitchenham, B., Sjøberg, D. I., Dybå, T., Brereton, P., Budgen, D., Höst, M. & Runeson, P. (2013), 'Trends in the quality of human-intensive software engineering experiments–a quasi-experiment', *IEEE Transactions on Software Engineering* **39**(7), 1002–1017.

Kocaguneli, E., Menzies, T. & Keung, J. W. (2012), 'On the value of ensemble effort estimation', *IEEE Transactions on Software Engineering* **38**(6), 1403–1416.

Kocaguneli, E., Menzies, T., Keung, J. W., Cok, D. & Madachy, R. (2013), 'Active learning and effort estimation: Finding the essential content of software effort estimation data', *IEEE Transactions on Software Engineering* **39**(8), 1040–1053.

Kollanus, S. & Koskinen, J. (2009), 'Survey of software inspection research', *The Open Software Engineering Journal* **3**, 15–34.

Kothari, A. & Armstrong, R. (2011), 'Community-based knowledge translation: unexplored opportunities', *Implementation Science* **6**(59).

Krippendorff, K. (1978), 'Reliability of binary attribute data', *Biometrics* **34**(1), 142–144.

Laitenberger, O., Emam, K. E. & Harbich, T. G. (2001), 'An internally replicated quasi-experimental comparison of checklist and perspective-based reading of code documents', *IEEE Transactions on Software Engineering* **27**(5), 387–421.

Li, Z., Zhang, H., O'Brien, L., Cai, R. & Flint, S. (2013), 'On evaluating commercial cloud services: A systematic review', *Journal of Systems & Software* **86**, 2371–2393.

Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gøtzsche, P. C., Ioannidis, J. P. A., Clarke, M., Devereaux, P. J., Kleijnen, J. & Moher, D. (2009), 'The prisma statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: explanation and elaboration', *BMJ* **339**.

Lindsay, R. M. & Ehrenberg, A. S. C. (1993), 'The design of replicated studies', *The American Statistician* **47**(3), 217–228.

Lisboa, L. B., Garcia, V. C., Lucrédio, D., de Almeida, E. S., de Lemos Meira, S. R. & de Mattos Fortes, R. P. (2010), 'A systematic review of domain analysis tools', *Information and Software Technology* **52**(1), 1–13.

López, A., Nicolás, J. & Toval, A. (2009), Risks and safeguards for the requirements engineering process in global software development, in *'Proceedings of 4th International Conference on Global Software Engineering'*, IEEE Computer Society Press, pp. 394–399.

Lucia, A. D., Gravino, C., Oliveto, R. & Tortara, G. (2010), 'An experimental comparison of ER and UML class diagrams for data modelling', *Empirical Software Engineering* **15**, 455–492.

MacDonell, S. & Shepperd, M. (2007), Comparing local and global software effort estimation models – reflections on a systematic review, in *'Empirical Software Engineering and Measurement, 2007.* ESEM 2007. First International Symposium on', pp. 401–409.

MacDonell, S., Shepperd, M., Kitchenham, B. & Mendes, E. (2010), 'How reliable are systematic reviews in empirical software engineering?', *IEEE Transactions on Software Engineering* **36**(5), 676–687.

Madeyski, L. & Kitchenham, B. (2014), How variations in experimental designs impact the construction of comparable effect sizes for meta-analysis. Available from Barbara Kitchenham.

Magdaleno, A. M., Werner, C. M. L. & de Araujo, R. M. (2012), 'Reconciling software development models: a quasi-systematic review', *Journal of Systems & Software* **85**, 351–369.

Mair, C., Shepperd, M. & Jørgensen, M. (2005), An analysis of data sets used to train and validate cost prediction systems, in *'Proceedings PROMISE '05'*, ACM.

Marques, A., Rodrigues, R. & Conte, T. (2012), Systematic literature reviews in distributed software development: A tertiary study, in *'Global Software*

Engineering (ICGSE), 2012 IEEE Seventh International Conference on', Global Software Engineering pp. 134–143.

Marshall, C., Brereton, O. P. & Kitchenham, B. A. (2014), Tools to support systematic literature reviews in software engineering: A feature analysis, *in* 'Proceedings of 18th International Conference on Evaluation and Assessment in Software Engineering (EASE'14)', ACM Press, pp. 13:1–13:10.

Marshall, C. & Brereton, P. (2013), Tools to support systematic literature reviews in software engineering: A mapping study, *in* 'Proceedings ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)', IEEE Computer Society Press, pp. 296–299.

McGarry, F., Burke, S. & Decker, B. (1998), Measuring the impacts individual process maturity attributes have on software products, *in* 'Proceedings 5th Software Metrics Symposium', pp. 52–60.

Mierzewski, P. (2001), Developing a methodology for drawing up guidelines on best medical practices, Technical report, Council of Europe.

Miles, M. B., Huberman, A. M. & Saldaña, J. (2014), *Qualitative Data Analysis A Methods Sourcebook*, 3rd edn, Sage Publications Inc.

Miller, J. (2005), 'Replicating software engineering experiments: a poisoned chalice or the holy grail', *Information & Software Technology* **47**, 233–244.

Mitchell, S. & Seaman, C. (2009), A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review, *in* 'Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on', pp. 511–515.

Moe, N. B., Dingsøyr, T. & Dybå, T. (2010), 'A teamwork model for understanding an agile team: A case study of a Scrum project', *Information & Software Technology* **52**, 480–491.

Mohagheghi, P. & Conradi, R. (2007), 'Quality, productivity and economic benefits of software reuse: a review of industrial studies', *Empirical Software Engineering* **12**, 471–516.

Mohagheghi, P. & Dehlen, V. (2008), Where is the proof? – a review of experiences from appying MDE in industry, in *'Model Driven Architectures–Foundations & Application'*, Vol. 5095/2008, Lecture Notes in Computer Science, Springer, pp. 432–443.

Moher, D., Liberati, A., Tetzlaff, J. & Group, D. G. A. T. P. (2009), 'Preferred reporting items for systematic reviews and meta-analyses: The prisma statement', *PLoS Med 6(7)* .

Moløkken-Østvold, K., Tanilkan, M. J. S. S., Gallis, H., Lien, A. C. & Hove, S. E. (2004), A survey on software estimation in the Norwegian industry, in *'Proceedings of 10th Internation Symposium on Software Metrics (METRICS'04)'*, pp. 1–12.

Morris, S. B. (2000), 'Distribution of the standardized mean change effect size for meta-analysis on repeated measures', *British Journal of Mathematical and Statistical Psychology* **53**, 17–29.

Morris, S. B. & DeShon, R. P. (2002), 'Combining effect size estimates in meta-analysis with repeated measures and independent-groups designs', *Psychological Methods* **7**(1), 105–125.

Munir, H., Moayyed, M. & Peterson, K. (2014), 'Considering rigor and relevance when evaluating test driven development: A systematic review', *Information & Software Technology* **56**, 375–394.

Myrtveit, I. & Stensrud, E. (2012), 'Validity and reliability of evaluation procedures in comparative studies of effort prediction models', *Empirical Software Engineering* **17**, 23–33.

Myrtveit, I., Stensrud, E. & Shepperd, M. (2005), 'Reliability and validity in comparative studies of software prediction models', *IEEE Transactions on Software Engineering* **31**(5), 380–391.

Nascimento, D., Cox, K., Almeida, T., Sampaio, W., Almeida Bittencourt, R., Souza, R. & Chavez, C. (2013), Using open source projects in software engineering education: A systematic mapping study, in *'Frontiers in Education Conference, 2013 IEEE'*, pp. 1837–1843.

NICE (2009), *The Guidelines Manual*, National Institute for Clinical Excellence (NICE).

Nicolás, J. & Toval, A. (2009), 'On the generation of requirements specifications from software engineering models: A systematic literature review', *Information and Software Technology* **51**(9), 1291–1307.

Noblit, G. & Hare, R. (1988), *Meta Ethnography: Synthesizing Qualitative Studies*, Sage Publications Ltd.

Noyes, J. & Lewin, S. (2011), Chapter 6: Supplemental guidance on selecting a method of qualitative evidence synthesis, and integrating qualitative evidence with cochrane intervention reviews, in J. Noyes, A. Booth, K. Hannes, A.Harden, J. Harris, S. Lewin & C. Lockwood, eds, *'Supplementary Guidance for Inclusion of Qualitative Research in Cochrane Systematic Reviews of Interventions.', version* 1 (updated august 2011) edn, Cochrane Collaboration Qualitative Methods Group.

Oates, B. (2006), *Researching Information Systems and Computing*, SAGE.

Oza, N. V., Hall, T., Rainer, A. & Grey, S. (2006), 'Trust in software outsourcing relationships: An empirical investigation of Indian software companies', *Information and Software Technology* **48**, 345–354.

Pacheco, C. & Garcia, I. (2012), 'A systematic literature review of stakeholder identification methods in requirements elicitation', *Journal of Systems & Software* **85**, 2171–2181.

Paternoster, N., Giardino, C., Unterkalmsteiner, M. & Gorschek, T. (2014), 'Software development in startup companies: A systematic mapping study', *Information & Software Technology* **56**, 1200–1218.

Penzenstadler, B., Raturi, A., Richardson, D., Calero, C., Femmer, H. & Franch, X. (2014), Systematic mapping study on software engineering for sustainability (SE4S) — protocol and results, ISR Technical Report UCI-ISR-14-1, Institute for Software Research, University of California, Irvine.

Persson, J. S., Mathiassen, L., Boeg, J., Madsen, T. S. & Steinson, F. (2009), 'Managing risks in distributed software projects: An integrative framework', *IEEE Transactions on Engineering Management* **56**(3), 508–532.

Petersen, K., Feldt, R., Mujtaba, S. & Mattsson, M. (2008), Systematic mapping studies in software engineering, in *'Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering'*, EASE'08, British Computer Society, Swinton, UK, pp. 68–77.

Peterson, K. (2011), 'Measuring and predicting software productivity: A systematic map and review', *Information & Software Technology* **53**, 317–343.

Petersson, H., Thelin, T., Runeson, P. & Wohlin, C. (2004), 'Capture-recapture in software inspections after 10 years research—theory, evaluation and application', *Journal of Systems and Software* **72**, 249–264.

Petre, M. (2013), UML in practice, in *'Proceedings of the 2013 International Conference on Software Engineering (ICSE)'*, IEEE Computer Society Press, pp. 722–731.

Petticrew, M. & Roberts, H. (2006), *Systematic Reviews in the Social Sciences A Practical Guide*, Blackwell Publishing.

Pfleeger, S. L. (1999), 'Understanding and improving technology transfer in software engineering', *Journal of Systems & Software* **47**, 111–124.

Phalp, K., Vincent, J. & Cox, K. (2007), 'Improving the quality of use case descriptions: empirical assessment of writing guidelines', *Software Quality Journal* **15**(4), 383–399.

Pino, F. J., Garcia, F. & Piattini, M. (2008), 'Software process improvement in small and medium software enterprises: a systematic review', *Software Quality Journal* **16**, 237–261.

Popay, J., Roberts, H., Sowden, A., Petticrew, M., Arai, L., Rodgers, M., Britten, N., Roen, K. & Duffy, S. (2006), Guidance on the conduct of narrative synthesis in systematic reviews, Technical report, Lancaster University, UK, Available from j.popay@lancaster.ac.uk.

Publication Manual of the American Psychological Association (2001), 5th edn, American Psychological Association, Washington, DC, USA.

Radjenović, D., Heričko, M., Torkar, R. & Živkovič, A. (2013), 'Software fault prediction metrics: A systematic literature review', *Information & Software Technology* **55**, 1397–1418.

Rafique, Y. & Misic, V. (2013), 'The effects of test-driven development on external quality and productivity: A meta-analysis', *IEEE Transactions on Software Engineering* **39**(6).

Ragin, C. C. (1989), *The Comparative Method*, University of California Press.

Remenyi, D. (2014), *Grounded Theory. A reader for Researchers, Studentts, Faculty and Others*, 2nd edn, Academic Conferences and Publishing International Ltd, Reading, UK.

Renger, M., Kolfschoten, G. L. & de Vreede, G.-J. (2008), Challenges in collaborative modelling: A literature review, in *'Proceedings of CIAO!* 2008 and EOMAS 2008', Vol. LNBIP 10, Springer-Verlag Berlin, pp. 61–77.

Riaz, M., Mendes, E. & Tempero, E. (2009), A systematic review of software maintainability prediction and metrics, in *'Proceedings of Third International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)'*, pp. 367–377.

Robinson, H., Segal, J. & Sharp, H. (2007), 'Ethnographically-informed empirical studies of software practicel studies of software practice', *Information and Software Technology* **49**(540-551).

Robson, C. (2002), *Real World Research*, 2nd edn, Blackwell Publishing, Malden.

Rogers, E. M. (2003), *Diffusion of Innovations*, 5th edn, Free Press, New York.

Ropponen, J. & Lyytinen, K. (2000), 'Components of software development risk: How to address them. A project manager survey', *IEEE Transactions on Software Engineering* **26**(2), 98–111.

Rosenthal, R. & DiMatteo, M. (2001), 'Meta-analysis: Recent developments in quantitative methods for literature reviews', *Annual Review of Psychology* **52**, 59–82.

Rosenthal, R. & Rubin, D. B. (2003), 'r(equivalent): A simple effect size indicator', *Psychological Methods* **8**(4), 492–496.

Rosnow, R. L. & Rosenthal, R. (1997), *People Studying People Artifacts and Ethics in Behavioural Research*, W.H. Freeman & Co., New York.

Rovegard, P., Angelis, L. & Wohlin, C. (2008), 'An empirical study on views of importance of change impact analysis issues', *IEEE Transactions on Software Engineering* **34**(4), 516–530.

Runeson, P., Andersson, C., Thelin, T., Andrews, A. & Berling, T. (2006), 'What do we know about defect detection methods?', *IEEE Software* **23**(3), 82–86.

Runeson, P. & Höst, M. (2009), 'Guidelines for conducting and reporting case study research in software engineering', *Empirical Software Engineering* **14**(2), 131–164.

Runeson, P., Höst, M., Rainer, A. & Regnell, B. (2012), *Case Study Research in Software Engineering: Guidelines and Examples*, Wiley.

Sackett, D., Straus, S., Richardson, W., Rosenberg, W. & Haynes, R. (2000), *Evidence-based medicine: how to practice and teach EBM*, second edn, Churchill Livingstone.

Salleh, N., Mendes, E. & Grundy, J. (2009), 'Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review', *IEEE Transactions on Software Engineering* **37**(4), 509–525.

Sandelowski, M. & Barroso, J. (2003), 'Creating metasummaries of qualitative findings', *Nursing Research* **52**(4), 226–233.

Sandelowski, M., Barroso, J. & Voils, C. I. (2007), 'Using qualitative metasummary to synthesize qualitative and quantitative descriptive findings', *Research in Nursing and Health* **30**(1), 99–111.

Sandelowski, M., Docherty, S. & Emden, C. (1997), 'Focus on qualitative methods qualitative metasynthesis: issues and techniques', *Research in Nursing and Health* **20**, 365–372.

Santos, R. E. S. & da Silva, F. Q. (2013), Motivation to perform systematic reviews and their impact on software engineering practice, in *'Proceedings ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)'*, IEEE Computer Society Press, pp. 292–295.

Schünemann, H. J., Fretheim, A. & Oxman, A. D. (2006), 'Improving the use of research evidence in guideline development: 1. guidelines for guidelines', *Health Research Policy and Systems* **4**(13).

Seaman, C. B. (1999), 'Qualitative methods in empirical studies of software engineering', *IEEE Transactions on Software Engineering* **25**(4), 557–572.

Seaman, C. B. & Basili, V. R. (1998), 'Communication and organization: An empirical study of discussion in inspection meetings', *IEEE Transactions on Software Engineering* **24**(6), 559–572.

Shadish, W., Cook, T. & Campbell, D. (2002), *Experimental and Quasi-Experimental Design for Generalized Causal Inference*, Houghton Mifflin Co.

Shahin, M., Liang, P. & Babar, M. A. (2014), 'A systematic review of software architecture visualization techniques', *Journal of Systems & Software* **94**, 161–185.

Shang, A., Huwiler-Müntener, K., Nartey, L., Jüni, P., Dörig, S., Sterne, J. A. C., Pewsner, D. & Egger, M. (2005), 'Are the clinical effects of homoeopathy placebo effects? comparative study of placebo-controlled trials of homoeopathy and allopathy', *The Lancet* **366**(9487), 726–732.

Sharp, H., Baddoo, N., Beecham, S., Hall, T. & Robinson, H. (2009), 'Models of motivation in software engineering', *Information and Software Technology* **51**, 219–233.

Sharp, H. & Robinson, H. (2008), 'Collaboration and co-ordination in mature extreme programming teams', *International Journal of Human-Computer Studies* **66**, 506–518.

Shaw, M. (2003), Writing good software engineering research papers (minitutorial), in *'Proceedings of 25th International Conference on Software Engineering (ICSE 2003)'*, IEEE Computer Society Press, p. 726.

Shepperd, M., Bowes, D. & Hall, T. (2014), 'Researcher bias: The use of machine learning in software defect prediction', *IEEE Transactions on Software Engineering* **40**(6), 603–616.

Shepperd, M. J. & MacDonell, S. G. (2012), 'Evaluating prediction systems in software project estimation', *Information and Software Technology* **54**(8), 820–827.

Shepperd, M., Song, Q., Sun, Z. & Mair, C. (2013), 'Data quality: Some comments on the NASA software defect datasets', *IEEE Transactions on Software Engineering* **39**(9), 1208–1215.

Sigweni, B., Shepperd, M. & Jørgensen, M. (2014), 'An extended mapping study of software development cost estimation studies', *IEEE Transactions on Software Engineering*. Under review.

Sjøberg, D., Hannay, J., Hansen, O., Kampenes, V., Karahasanović, A., Liborg, N.-K. & Rekdal, A. (2005), 'A survey of controlled experiments in software engineering', *IEEE Transactions on Software Engineering* **31**(9), 733–753.

Sjøberg, D. I. K., Dybå, T. & Jørgensen, M. (2007), The future of empirical methods in software engineering research, in *'Future of Software Engineering (FOSE'07)'*, Future of Software Engineering.

Skoglund, M. & Runeson, P. (2009), Reference-based search strategies in systematic reviews, in *'13th International Conference on Evaluation and Assessment in Software Engineering (EASE)'*.

Smite, D., Wohlin, C., Gorschek, T. & Feldt, R. (2010), 'Empirical evidence in global software engineering: a systematic review', *Empirical Software Engineering* **15**, 91–118.

Spencer, L., Ritchie, J., Lewis, J. & Dillon, L. (2003), *Quality in Qualitative Evaluation: A framework for assessing research evidence*, Cabinet Office.

Staples, M. & Niazi, M. (2008), 'Systematic review of organizational motivations for adopting CMM-based SPI', *Information and Software Technology* **50**, 605–620.

Steinmacher, I., Chaves, A. & Gerosa, M. (2013), 'Awareness support in distributed software development: A systematic review and mapping of the literature', *Computer Supported Cooperative Work (CSCW)* **22**(2-3), 113–158.

Straus, S. E., Tetroe, J. & Graham, I. (2009), 'Defining knowledge translation', *Canadian Medical Association Journal* **181**(3-4), 165–168.

Sun, Y., Yang, Y., Zhang, H., Zhang, W. & Wang, Q. (2012), Towards evidence-based ontology for supporting systematic literature review, in *'Proceedings of 16th International Conference on Evaluation and Assessment in Software Engineering (EASE 2012)'*, pp. 171–175.

Tahir, A., Tosi, D. & Morasca, S. (2013), 'A systematic review on the functional testing of semantic web services', *Journal of Systems & Software* **86**, 2877–2889.

Thomas, J. & Harden, A. (2008), 'Methods for the thematic synthesis of qualitative research in systematic reviews', *BMC Medical Research Methodology* **8**(45).

Thorne, S., Jensen, L., Kearney, M. H., Noblit, G. & Sandelowski, M. (2004), 'Qualitative metasynthesis: Reflections on methodological orientation and ideological agenda', *Qualitative Health Research* **14**(10), 1342–1365.

Tichy, W. F. (1998), 'Should Computer Scientists Experiment More?', *IEEE Computer* **31**(5), 32–40.

Tomassetti, F., Rizzo, G., Vetro, A., Ardito, L., Torchiano, M. & Morisio, M. (2011), Linked data approach for selection process automation in systematic reviews, in *'Proceedings of 15th International Conference on Evaluation and Assessment in Software Engineering (EASE 2011)'*, pp. 31–35.

Torres, J., Cruzes, D. S. & do Nascimento Salvador, L. (2012), Automatic results identification in software engineering papers, is it possible?, in *'Proceedings of 12th International Conference on Computational Science and Its Applications (ICCSA 2012)'*, pp. 108–112.

Toye, F., Seers, K., Allcock, N., Briggs, M., Carr, E., Andrews, J. & Barker, K. (2013), 'Trying to pin down jelly - exploring intuitive processes in quality assessment for meta-ethnography', *BMC Medical Research Methodology* **13**(46).

Trendowicz, A. & Münch, J. (2009), Factors influencing software development productivitity—state-of-the-art and industrial experiences, in *'Advances in Computers'*, Vol. 77, Elsevier, pp. 185–241.

Truex, D., Baskerville, R. & Klein, H. (1999), 'Growing systems in emergent organisations', *Communications of the ACM* **42**(8), 117–123.

Tsafnat, G., Glasziou, P., Choong, M., Dunn, A., Galgani, F. & Coiera, E. (2014), 'Systematic review automation technologies', *Systematic Reviews* **3**(1), 74.

Turner, M., Kitchenham, B., Brereton, P., Charters, S. & Budgen, D. (2010), 'Does the technology acceptance model predict actual use? A systematic literature review', *Information and Software Technology* **52**(5), 463–479.

Verner, J., Brereton, O., Kitchenham, B., Turner, M. & Niazi, M. (2012), Systematic literature reviews in global software development: A tertiary study, *in* 'Evaluation Assessment in Software Engineering (EASE 2012), 16th International Conference on', pp. 2–11.

Verner, J., Brereton, O., Kitchenham, B., Turner, M. & Niazi, M. (2014), 'Risks and risk mitigation in global software development: A tertiary study', *Information and Software Technology* **56**(1), 54–78. Special sections on International Conference on Global Software Engineering – August 2011 and Evaluation and Assessment in Software Engineering – April 2012.

Viechtbauer, W. (2007), 'Accounting for heterogeneity via random-effects models and moderator analyses in meta-analyses', *Journal of Psychology* **215**(2), 104–121.

Viechtbauer, W. (2010), 'Conducting meta-analyses in r with the metafor package', *Journal of Statistical Software* **36**(3).

Walia, G. S. & Carver, J. C. (2009), 'A systematic literature review to identify and classify software requirement errors', *Information and Software Technology* **51**(7), 1087–1109.

WHO (2005), Bridging the "know-do" gap: Meeting on knowledge translation in global health, Technical report, World Health Organisation.

Wieringa, R., Maiden, N., Mead, N. & Rolland, C. (2006), 'Requirements engineering paper classification and evaluation criteria: A proposal and a discussion', *Requirements Engineering* **11**(1), 102–107.

Williams, B. J. & Carver, J. C. (2010), 'Characterizing software architecture changes: A systematic review', *Information & Software Technology* **52**(1), 31–51.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B. & Wesslen, A. (2012), *Experimentation in Software Engineering*, 2nd edn, Springer.

Yin, R. K. (2014), *Case Study Research: Design & Methods*, 5th edn, Sage Publications Ltd.

Yin, R. K. & Heald, K. A. (1975), 'Using the case survey method to analyze policy studies', *Administrative Science Quarterly* **20**, 371–381.

Zelkowitz, M. V. & Wallace, D. R. (1998), 'Experimental models for validating technology', *IEEE Computer* **31**, 23–31.

Zhang, C. & Budgen, D. (2012), 'What do we know about the effectiveness of software design patterns?', *IEEE Transactions on Software Engineering* **38**(5), 1213–1231.

Zhang, C. & Budgen, D. (2013), 'A survey of experienced user perceptions about design patterns', *Information & Software Technology* **55**(5), 822–835.

Zhang, H. & Babar, M. A. (2013), 'Systematic reviews in software engineering: An empirical investigation', *Information and Software Technology* **55**(7), 1341–1354.

Zhang, H., Babar, M. A. & Tell, P. (2011), 'Identifying relevant studies in software engineering', *Information and Software Technology* **53**(6), 625 – 637.

Zwarenstein, M. & Reeves, S. (2006), 'Knowledge translation and interprofessional collaboration: Where the rubber of evidence-based care hits the road of teamwork', *Journal of Continuing Education in the Health Professions* **26**, 46–54.

This page intentionally left blank