# Preliminary Reporting Guidelines for Experience Papers

David Budgen & Cheng Zhang
Department of Computer Science
Durham University, U.K.
{david.budgen,cheng.zhang2}@durham.ac.uk

#### **Abstract**

Context: When undertaking a systematic literature review or a mapping study software engineering, it is likely that only a small set of experimental studies will be available. In conducting a mapping study on the theme of software design patterns, we found only 11 papers describing experiments that studied the use of patterns.

Objectives: To investigate whether we could obtain further evidence by examining the experiences offered in papers that were essentially observational in nature. To use this experience to suggest how such studies can best be reported.

Method: We identified suitable studies from those identified in our systematic search and undertook data extraction from them. We then analysed those that were of most use, to identify what characteristics made their reporting useful.

Results: We found 18 experience papers, but after analysis, this set was reduced to four. Only one of these provided a clear link between practical experiences and the lessons they reported. Our preliminary reporting guidelines are based upon both good and poor papers, as well as the guidelines proposed for other forms of empirical study. Conclusions: We draw upon our experiences of data extraction, and of the one good example to suggest reporting guidelines for experience papers.

Keywords: Evidence-based, empirical studies, observational studies, design patterns

#### 1. INTRODUCTION

Since the idea of adapting the *evidence-based* paradigm for use in software engineering was first proposed in 2004 (Kitchenham et al. 2004), it has become widely accepted as a useful addition to the research toolset used for empirical software engineering. The original 2004 guidelines were updated in 2007, based upon the available experiences (Kitchenham & Charters 2007). A major tool of the evidence-based paradigm is the Systematic Literature Review (SLR), and since 2004 we are aware of at least 20 published papers that describe secondary studies—whether complete SLRs or Mapping Studies (Kitchenham et al. 2009).

In the discipline that pioneered the adoption of evidence-based practices, namely clinical medicine, the 'gold standard' for the conduct of a primary study is that of the Randomised Controlled Trial (RCT). An RCT usually employs 'double blinding', by which neither the researcher administering a treatment, nor the participants in the trial (the recipients of the treatment), are aware of who is a member of the control group and who is actually receiving the experimental treatment. Where an SLR draws together a set of RCTs, there is then scope to use statistical techniques for aggregating the results, so increasing confidence in the outcomes of the review.

However, performing an RCT is impractical for those disciplines in which the participants in a study are required to deploy specific knowledge or skills in order to address the needs of the experimental treatment, rather than simply acting as a *recipient*—as is the case for software engineering. To date though, the SLRs and mapping studies that have been undertaken in software engineering have concentrated on aggregating the outcomes from controlled laboratory experiments, with little attention being paid to other forms of study. Indeed, in terms of the various forms of controlled study that is not surprising, since software engineering currently makes little

use of forms such as case studies or quasi-experiments. This in turn leaves little more than observational studies—usually in the form of informal 'experience' reports.

We have been undertaking a mapping study to determine the extent to which the claims of the software design patterns community is supported by experimental studies (Zhang & Budgen 2009). One of our problems has been that, despite a very thorough searching process, we have found very few experimental studies about object-oriented patterns (in terms of studying their effectiveness). This is by no means a unique situation, and it would appear that for many key topics in software engineering we have very little in the way of useful empirical studies (Budgen, Turner, Brereton & Kitchenham 2008). Our eventual shortlist contained only 11 papers, and although some of these do report the results from multiple studies, the overall total is still very small. In particular, we found that only a small set of patterns had been studied within these, corresponding to a little over half of the 23 patterns described in the 'classical' textbook on design patterns (Gamma et al. 1995). This means that we can find relatively little in the way of reliable reporting of the effectiveness of specific patterns. We therefore decided to investigate whether we could obtain any useful supplementary information from experience reports, in the hope that using these might provide some degree of triangulation in our assessments.

Our hopes for this were twofold. First of all to find out if the observational studies reported on the same set of patterns as the experiments, so that some degree of triangulation was practical. Second was to identify how far the relatively small-scale experience from experimental studies was consistent with larger-scale use of patterns—providing some indication of the external validity of such experiments.

In this study we therefore set out to answer the following questions:

- 1. Can we find useful supplementary forms of evidence in observational 'experience papers'?
- 2. Based upon our experiences, can we recommend better ways of reporting such evidence?

For this purpose, we regard an 'experience paper' as one that provides a set of observations that are based upon practical experience (in this case, experience of using design patterns), and where these observations are summarised as 'lessons' that have been learned from this experience.

The following sections describe how we identified a set of experience papers; and the result of our attempts to obtain useful data from these. In this paper our main interest is in examining why these attempts only met with limited success. Indeed, only one paper reliably linked its 'lessons' to specific experiences. We therefore look at the quality and form of reporting in the papers examined, and seek to obtain some preliminary reporting guidelines from the limited set that provided useful data. Our aim in so doing is to encourage the empirical community to give more attention to these reports and to encourage better standards of reporting.

## 2. METHOD

# 2.1. Identifying 'experience' papers

The search process for the mapping study employed four rounds of searching: two using electronic search engines and a range of search strings; a round of manual searching of major journals; and a 'snowball' search based upon the references in the papers found in the previous three rounds. From these four rounds, we identified 181 candidate empirical papers, which we then classified in terms of their research strategy. For this paper we are interested in the category of 'experience', where these were essentially papers that provided observations that were focused upon extracting lessons from the application of patterns. The initial set of these was 27, with this being reduced to 18 after applying our inclusion and exclusion criteria.

The electronic search engines used were: ACM, IEEExplore, CiteSeer, Science Direct, Web of Science, and Google Scholar. We used the terms "software + pattern", together with:

**Round 1:** experience, investigation, experiment, study, experimental, empirical, apply, use, implementation, application;

**Round 2:** investigate, investigating, experimentation, utilise, utilisation, employ, practice, survey, work, sketch, analyse, analysis, usage, exercise, implement, construct.

The decisions about inclusion and exclusion were made in three stages:

- For all papers found, we initially selected books and papers describing software design patterns, and where there were several papers reporting the same study we included only the most recent. We specifically excluded literature that was only available as abstracts, technical reports or PowerPoint presentations, as well as papers that were submitted for publication.
- 2. We then filtered further, including only papers that specifically reported on empirical studies while also excluding those that used design patterns rather than reporting on them. The resulting set provided the basis for our study of papers describing experiments.
- 3. For the part of the study reported here, we specifically included papers that contained experience about using design patterns, while excluding any that did not focus on development and maintenance, or which did not provide any 'lessons' from the experience.

In the first two stages, the inclusion/exclusion decisions were made by one of us (CZ) and a sample were then checked by the second (DB).

#### 2.2. Data Collection

In devising data extraction needs (and hence data extraction forms) for experimental studies, researchers can draw upon the experiences of other disciplines and the experiences from other software engineering studies (Kitchenham & Charters 2007). However, there is little available to aid with devising the format to use for performing data extraction from observational studies. We therefore began by adapting an existing form (used for experiments), and revised this after consultation with an expert researcher (Kitchenham).

The resulting form that we employed is shown in Appendix A. As we will show later, the structure we used did not prove wholly satisfactory, although the problems encountered were relatively minor. The form itself is subdivided into four main sections as follows:

- 1. *Q1–4. Citation details.* These are relatively standard.
- 2. Q5-11. Study context. In these questions we sought to identify the characteristics that might influence the way that any outcomes should be interpreted and weighted. We were interested in knowing how independent the authors were (essentially by looking at the references to see if they were authors of patterns or books about patterns); in knowing about the system(s) that provided the source of the experiences; in knowing whether these experiences were first hand or not; the level of abstraction at which the experiences were discussed; and how these were related to the software life-cycle.
- 3. *Q12–14. Information provided.* Here our interests were in the details of the patterns involved; the conclusions about them; and how these were derived.
- 4. Q15. Decision about inclusion. This simply recorded our final decision.

## 2.3. Data Extraction

For data extraction we used the form described above, and this was completed for each paper by both authors, working independently. Our conclusions were then put into a spreadsheet and checked for consistency. The final selection process was based upon two main criteria:

- That the paper identified specific design patterns (necessary for our original purpose of triangulation with the results from the experiments).
- That the 'lessons' described in a paper were linked to specific experiences.

We agreed on the final decision for 16 of the 18 papers, excluding all but three of these. After a further review of the two where we disagreed, we decided to discard one, while keeping the other (this did not have lessons on specific patterns but did embody some useful experience).

## 3. OUTCOMES

The number of experience papers was smaller than anticipated, and substantially fewer than the number of papers reporting experiments. However, there are two factors that might influence this. One is that most of the papers describing experiments were generated by only two research groups, and with two of the others being replications. The second is that many experience papers are reporting on the application of patterns without reflecting on the patterns themselves.

Our expectation was that we might gain some qualitative assessments about individual patterns from these studies. These could then be combined with the qualitative elements that are provided in some of the reports on experiments. However, the key limitation affecting almost all of the papers was the lack of any clear link between the experiences described in the paper and the conclusions (or 'lessons') drawn by the authors. While this does not necessarily render the conclusions invalid in any way, it does mean the the 'expert assessment' elements of the reports are implicit rather than explicit. The only papers retained for further analysis were those where we considered this element to be sufficiently explicit.

In the next sub-section we describe the four papers that reported on specific patterns, and then we report on how well we were able to extract data from the set of 18 papers. Finally, we examine how well the experiences could be linked to the outcomes from the experiments.

## 3.1. The included papers

For convenience we will refer to these four papers as **exp-1** to **exp-4** in the rest of this paper.

- 1. exp-1. This paper, by Doug Schmidt (1995), is focused upon reuse in developing communications software. A relatively early paper, and hence partly proselytising in nature, it provides a long list of lessons but does link these (directly and indirectly) to the experiences gained from developing three systems. The paper concludes that using the *Reactor* pattern (a variant of *Observer*) has the benefit of increasing portability and avoiding the need for threads when handling events from multiple devices, offset by the loss of handler preemption as well as creating more complex flow of control to complicate debugging. As such it assesses the outcomes in terms of both benefits and disadvantages.
- 2. exp-2. The paper by Wang Yuanhong et al. describes experiences gained in developing an IDE (Yuanhong et al. 1997). Again, it is a relatively early paper, but does specifically report on the use of two specific patterns (*Composite* and *State*), although their assessment (that Composite eases the use of similar operations with different elements, while State simplifies the design of event responses) is not illustrated by specific examples from the system they developed. The emphasis is also upon benefits, with disadvantages being related to rather generic issues rather than the specific patterns.
- 3. **exp-3**. This report, by Peter Wendorff (2001), provides a valuable example of clear reporting of experiences. Its focus is upon the experiences derived from maintenance rather than development, related to a large system that had been developed with patterns being used for many parts. It links the discussion of four particular patterns (*Proxy*, *Observer*, *Bridge* and *Command*) to specific experiences with changing the structure of the system. The paper distinguishes between where patterns may be intrinsically less useful (such as the added indirection involved in using *Proxy*) as well as where they were used inappropriately (and why this was so) and the consequences for system maintenance.
- 4. **exp-4**. This paper, by Gou Masuda *et al.* describes the experience of applying design patterns to construct a flexible system (Masuda et al. 1998). While the lessons are based upon specific patterns, they are not directly linked to them and we primarily retained this paper because of its contribution in terms of the experiences related to flexibility.

We now go on to examine how well the different characteristics were reported by discussing our experiences of extracting the necessary data. While our discussion ranges across the 18 papers reviewed, we will particularly draw examples from the first three.

## 3.2. Describing the source of experiences (Q6-Q11)

Q6 ("form of study") proved difficult to code for this group of papers (experience papers are mostly observational, but organised in different ways), and most of the issues involved were essentially captured by the remaining questions. Hence we will not separately analyse the results for Q6.

For Q7 ("type of system"), few papers reported this information, although it is quite important for the reader who might want to be reassured that the experience provided is relevant to their needs. The first three papers described in the previous section did report the basic purpose of the systems involved (although Wendorff gave only a very vague indication). There may well be good commercial reasons for not giving much detail, but that should not prevent enough being given for the reader to be able to determine how it relates to their interests.

Q8 was intended to extract the size of the system that formed the source of these experiences. Only three papers in the set of 18 gave any indication of this, and then all three used totally different measures (KLOC, number of classes, and years of development plus number of developers). Again, this is information that is really needed by the reader in order to assess the relevance and quality of the information—especially as this is one aspect of experience papers that is likely to differ very substantially from controlled laboratory experiments.

Q9 addressed the issue of whether the experience was from the authors' own development work or that of others (allowing for the possibility of being both) and extracting this generally proved straightforward. However, there were still papers where this was not made clear to the reader.

Q10 sought to distinguish between experience that was gained from working at a design level of abstraction, or from using the (code-oriented) realisations of patterns. Again, this was usually made evident in the discussion provided in the papers.

Finally, Q11 focused on a rather important distinction, which was whether a paper described experiences gained from development or from maintenance. While the former was by far the most common, maintenance does provide a better retrospective view of the effects that the use of patterns to structure a system can have upon its form and performance.

Overall, these proved to be an important set of questions in terms of making a decision about inclusion, but many of the aspects involved were either poorly reported or not reported at all.

## 3.3. Describing the experiences (Q12-Q14)

As originally constructed, the data extraction form had no real structure for Q12 ("patterns discussed"). In retrospect, when creating sub-questions for Q12, we should have removed Q13, which was effectively subsumed into Q12. We should also have moved Q14 to be one of the subquestions of Q12, since our interest was in obtaining this information (about how the experience was linked to the conclusions) on a per-pattern basis.

Few of the papers provided descriptions of specific patterns, and of their experiences regarding these. Even fewer linked their experiences to their conclusions. The one positive example was the paper by Wendorff, which did discuss specific patterns, and provided some specific conclusions about these that arose from the experiences.

#### 4. DISCUSSION

## 4.1. Threats to Validity

For a paper such as this, which is seeking to derive lessons from analysis of the literature, there are some possible threats to validity to consider.

## 4.1.1. Internal Validity

Two obvious issues here, related to our primary purpose of studying design patterns are our search process and our classification processes. Searching used a broad set of search engines with a wide range of search strings, backed up by both a manual search of the major journals and also a 'snowball' search using the references in the papers as selected. For classification, in the first two stages we used a model of one analyst and a checker who looked at 10% of the papers and did an independent classification. For the third stage, we used two analysts independently checking papers and got a good level of consistency as reported above.

In terms of the lessons derived, we have used both established guidelines for other forms of study and also drawn from our (admittedly small) sample of selected experience papers. As such, we believe that we have addressed most of the factors relating to our chosen topic (design patterns).

## 4.1.2. External Validity

Our main concern for this is whether the guidelines derived in part from studying design patterns are equally applicable to other forms of software engineering study. While we could argue that other forms of 'strategy' such as testing are probably equivalent, we cannot be confident as yet that these guidelines would be as appropriate for (say) studies of specific artefacts or of processes.

## 4.2. Reporting—some preliminary guidelines

In proposing a preliminary set of guidelines we have drawn heavily upon two sources:

- The one good experience paper that we could use for data extraction (Wendorff 2001).
- Existing guidelines for reporting of experimental studies (Kitchenham et al. 2002, Jedlitschka et al. 2008).

The rest of this section identifies particular ideas that we can draw from these sources.

One distinctive aspect of experience papers is that they are naturally retrospective in nature, and as a result, data collection seems rarely to have been planned or based upon any form of study protocol. So, some of what is proposed in existing guidelines is unlikely to be available, especially where this relates to the planning of a study. However, as our one good example indicates, it is still possible to extract some useful information from such papers—and if we can encourage improved reporting we may also be able to encourage better planning of such studies.

We have based our guidelines (rather loosely) upon the structure proposed in (Jedlitschka et al. 2008), adapting this to the needs of observational data collection as necessary. the key structures are summarised in Table 1, using the same format as used in Table 2 of (Jedlitschka et al. 2008). While addressing the items that we consider to be relevant, we have only expanded upon the issues that seem particularly appropriate to this form. Note too, that although we refer to products (systems), much of this is also applicable to processes too.

## 4.2.1. Title and Authorship

The title should contain such keywords as 'experience', 'observation' and 'lessons' to indicate that it is empirical but not experimental. Authorship issues are as in Jedlitschka *et al.* 

## 4.2.2. Structured Abstract

Our own study of the usefulness of structured abstracts in software engineering emphasises the value of these and their role in reducing the overheads of searching for relevant papers (Budgen, Kitchenham, Charters, Turner, Brereton & Linkman 2008). Our preferred formulation is that used in the abstract for this paper.

	Section	Content	Scope	Priority
4.2.1	Title &		Does it make the 'observational' aspects	required
			of the study clear?	
	Authorship		Are valid contact details provided?	
4.2.2	Structured	context	Why is this research important?	required
	Abstract	objectives	What question does this address?	required
		method	What is the form of study used?	required
		results	What are the main findings?	required
		conclusions	What are the implications?	required
4.2.3	Keywords		Area of research and study form	If required
			·	by publisher
4.2.4	Introduction	author role	Identify author responsibilities	required
		measures	How related to purpose of study?	required
		bias	How might the observations be biased?	required
4.2.5	Background	system details	Source of the experiences	required
		system size	Details of the source size	if available
		lifecycle role	How related to development cycle?	required
		motivation	Original problem leading to study	required
		direct or not	Whether own experiences or of others?	required
		source data	How data was collected?	required
		participants	Who took part?	if appropriate
4.2.6	Lessons	element details	What is the 'technology unit'?	required
		element description	How is the technology unit defined?	required if
				not published
				elsewhere
		advantages	What were the positive observations?	if available
		disadvantages	What were the negative observations?	if available
		conclusions	What was the outcome for the element?	required
		how derived	How conclusions link to observations?	required
		limitations	What might influence interpretation of	required
			any conclusions	
4.2.7	Threats to	construct	Do measured used relate to questions?	required
	Validity	internal	Quality of causal link between	required
			observations and any lessons derived	l . <u>.</u>
		external	Limitations on scope of experiences	if possible
		conclusion	Identify any possible sources of bias	if possible
4.2.8	Conclusions &	summary	Provide a concise summary	required
	Further Work	future work	What might be done to explore the	if possible
			lessons any further?	
4.2.9	Acknowled-		Sponsors, participants and other	if appropriate
	-gements		contributors not qualifying as authors	

TABLE 1: Summary of proposed guidelines

# 4.2.3. Keywords

Again, these assist with searching for relevant evidence and should be selected to as to emphasise both the observational element of the study as well as the topic.

## 4.2.4. Introduction

We can identify three important issues that an introduction to such a report should address:

- 1. the role of the author(s) in the study (for example, did they commission it, manage it, perform the tasks involved, etc.;
- 2. the measures of importance that were used, such as reducing the size of a system, improving its performance, making it easier to maintain it, . . . ;
- 3. factors that might bias the reporting for the example of our own study we looked at the references to try to establish whether any of the authors were themselves involved in promoting the use of design patterns through books or similar forms.

#### 4.2.5. Background

This is a particularly important element for an experience paper. In particular this can provide the context for the paper as a whole, by providing details about both source material and the observation process itself:

- 1. the systems (or processes) providing the experience (such as purpose, domain of use, ...;
- 2. the size of the system(s) preferably using a measure such as LOC;
- 3. whether the experience arose from new development or maintenance;
- 4. the motivation for examining the theme of the study (for example, in our case, it was the need to find more evidence about the use of software design patterns);
- 5. whether the experience is direct (first-hand involvement of the author(s)) or indirect (reporting on the work of others) the purpose of this is to identify how separate the observer was from the subject, since this affects the likely objectivity of any records;
- 6. the type of records that were used to provide the source material for the observations;
- 7. where appropriate, details of any 'participants' (particularly where the experiences relate to processes rather than products).

#### 4.2.6. Lessons

A major objective for any observational study must be to provide 'lessons' that are derived from the observations that embody the 'experience'. Regardless of how these are expressed and presented, the key need is for a clear link between the observations and the lessons (widely absent in the case of the design patterns papers). Ideally the link should be presented in terms of other appropriate elements. Again, using the study on design patterns as our example, these elements were the specific patterns themselves. We would therefore suggest that for each element, this section should address:

- 1. details of the element;
- 2. where appropriate, sources for a description of the element;
- 3. advantages related to the use of the element;
- 4. disadvantages related to the use of the element;
- 5. any conclusions about the element:
- 6. how the conclusions were derived from the observations (with examples);
- 7. any limitations upon the conclusions (for example, any quality-related issues that might influence interpretation of the lessons).

## 4.2.7. Threats to Validity

This concept was completely absent from the papers on design patterns that we reviewed. However, we would argue that there is some scope to consider these, albeit in a less rigorous manner than would be appropriate for (say) a controlled randomised experiment. The following issues could possibly be addressed under this heading.

- 1. Construct validity. This would essentially be related to any measures employed (LOC changed, time to create/change) and how well they represent the questions being addressed. For example, in (Wendorff 2001) the author often uses 'LOC saved' as a measure of system improvement, but does not discuss how valid this is for his context.
- 2. *Internal validity.* This should really focus upon the causal link between observations and lessons—since this element is nearly always missing, encouraging discussion of this might motivate authors to consider how well their conclusions are supported by their data.
- 3. *External validity.* At the least this should involve a discussion about any aspects of the context that might limit the extent to which the lessons can be generalised.
- 4. Conclusion validity. This relates to the correctness of any lessons derived. It is not readily addressed by any analytical means, but it is a heading that could well capture the issue of observer bias. One of our concerns in selecting experience papers on design patterns was to try to identify whether any of the authors of a paper had a 'vested interest' that might bias the conclusions, such as being an author of a book about design patterns, or even the author of one or more patterns.

#### 4.2.8. Conclusions and Further Work

In the paper by Jedlitschka *et al.* they suggest three elements for this, including a summary, an assessment of impact and a discussion of future work. While there seem good arguments for providing both a summary and ideas about future work (especially where this might follow up particular results more systematically), the role of impact is less clear. By its nature, an 'experience' paper is unlikely to provide sufficient evidence that would give confidence to others to pursue particular policies or practices (although we accept this does not necessarily stop the authors of such papers from encouraging this). On that basis, we would suggest confining the conclusions to only the two main elements might also help to reduce the excessive element of advocacy so often encountered in software engineering papers.

## 4.2.9. Acknowledgements

At the minimum this should acknowledge the contributions of others who have worked on the 'system', and whose efforts have provided the source material for the lessons.

## 5. CONCLUSIONS

Here we briefly return to our own two research questions and examine how far we have managed to answer these.

Can we find useful supplementary forms of evidence in observational 'experience papers'? Our answer to this has to be 'yes', since we have found one paper that largely meets this, as well as three others that do at least offer partial evidence. However, a caveat should be also be made to the effect that poor reporting standards make it difficult to extract such evidence.

Based upon our experiences, can we recommend better ways of reporting such evidence? This was addressed in the previous section and indicates that such evidence can be reported in a manner that makes it of greater usefulness than in the examples identified in this study, while accepting that we have not yet really demonstrated that our proposals will provide 'better' reporting.

Our results are essentially preliminary, and are based upon the study of one topic in software engineering. Future work might usefully assess these guidelines against experience papers for other topics. Equally, we are keen to encourage others to adopt these guidelines in order to raise the standards of reporting for such studies. Overall, we would argue that, with better reporting, 'experience papers' have the potential to provide a potentially useful source of supplementary evidence that can be used to help interpret more rigorous (but also more restricted) studies.

#### 6. ACKNOWLEDGEMENTS

We would like to thank the EPSRC for support via project EP/E046983/1, Evidence-based Practices Informing Computing (EPIC), and to Prof. Barbara Kitchenham for her advice. We also thank the anonymous reviewers for their valuable comments and suggestions.

# **REFERENCES**

- 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., Turner, M., Brereton, P. & Kitchenham, B. (2008), Using Mapping Studies in Software Engineering, *in* 'Proceedings of PPIG 2008', Lancaster University, pp. 195–204.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.
- Jedlitschka, A., Ciolkowski, M. & Pfahl, D. (2008), Reporting experiments in software engineering, in F. Shull, J. Singer & D. Sjøberg, eds, 'Guide to Advanced Empirical Software Engineering', Springer-Verlag, London, chapter 8.

- 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. & Charters, S. (2007), Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE 2007-001, 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., 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.
- Masuda, G., Sakamoto, N. & Ushijima, K. (1998), Applying design patterns to decision tree learning system, *in* 'Proceedings of 6th ACM SIGSOFT International Symposium on Foundations of Software Engineering SIGSOFT'98/FSE-6', ACM Press, pp. 111–120.
- Schmidt, D. (1995), 'Using Design Patterns to develop reusable Object-Oriented Communication Software', *Communications of the ACM* **38**(10), 65–74.
- Wendorff, P. (2001), Assessment of design patterns during software reengineering: Lessons learned from a large commercial project, *in* 'Proceedings of 5th European Conference on Software Maintenance and Reengineering (CSMR'01)', IEEE Computer Society Press, pp. 77–84.
- Yuanhong, W., Hong, M. & Weizhong, S. (1997), Experience report: Using design patterns in the development of jb system, *in* 'Proceedings of Technology of Object Oriented Languages and Systems: TOOLS 24', IEEE Computer Society Press, pp. 159–165.
- Zhang, C. & Budgen, D. (2009), Assessing the claims for software design patterns. Paper in preparation.

#### A. DATA COLLECTION FORM

We used a data collection form based upon the following elements. Where we provided specific codes for classifying an element, this is indicated, otherwise entries were free text.

	Element	Code	Interpretation
1	Reference No.	no	Our own code related to how the paper was found
2	Authors	no	Paper details recorded for convenience
3	Title	no	
4	Where published	no	
5	Are authors pattern	yes	We tried to identify whether the authors had published
	developers?		tutorials or books about patterns, by looking at references
6	Form of study	no	How the experiences were observed
7	Type(s) of system	no	Details of any specific systems that provided the experience
8	Size of system(s)	no	Any indicators of size that were provided
9	Experience from own	yes	Used to identify how direct the sources of experience were
	development or others		
10	Design/Coding	yes	The level of abstraction used in the paper
11	Development/Maintenance	yes	Where in the life-cycle the experience originated
12	Patterns Discussed		(fields a-e repeated for each pattern discussed in the paper)
	a. Pattern Name	no	Specific pattern for which experience was provided
	b. Pattern Source	no	Where the pattern was published
	<ul><li>c. Advantages/benefits</li></ul>	no	Positive experiences from using the pattern
	d. Problems/disadvantages	no	Negative experiences from using the pattern
	e. Conclusion	no	Any conclusions identified about use of this pattern
13	Conclusions	no	Any overall conclusions about using patterns
14	Form of link between	no	How the conclusions (individual patterns or overall) were
	experience & conclusions		derived from specific experiences
15	Keep or reject	yes	The final recommendation