

Reporting computing projects through structured abstracts: a quasi-experiment

David Budgen · Andy J. Burn · Barbara Kitchenham

Published online: 6 August 2010
© Springer Science+Business Media, LLC 2010

Editor: Claes Wohlin

Abstract Previous work has demonstrated that the use of structured abstracts can lead to greater completeness and clarity of information, making it easier for researchers to *extract* information about a study. In academic year 2007/08, Durham University’s Computer Science Department revised the format of the project report that final year students were required to write, from a ‘traditional dissertation’ format, using a conventional abstract, to that of a 20-page technical paper, together with a structured abstract. This study set out to determine whether inexperienced authors (students writing their final project reports for computing topics) find it easier to *produce* good abstracts, in terms of completeness and clarity, when using a structured form rather than a conventional form. We performed a controlled quasi-experiment in which a set of ‘judges’ each assessed one conventional and one structured abstract for its completeness and clarity. These abstracts were drawn from those produced by four cohorts of final year students: two preceding the change, and the two following. The assessments were performed using a form of checklist that is similar to those used for previous experimental studies. We used 40 abstracts (10 per cohort) and 20 student ‘judges’ to perform the evaluation. Scored on a scale of 0.1–1.0, the mean for completeness increased from 0.37 to 0.61 when using a structured form. For clarity, using a scale of 1–10, the mean score increased from 5.1 to 7.2. For a minimum goal of scoring 50% for both completeness and clarity,

D. Budgen (✉) · A. J. Burn
School of Engineering & Computing Sciences,
Durham University, DH1 3LE, Durham, UK
e-mail: david.budgen@durham.ac.uk

A. Burn
e-mail: a.j.burn@durham.ac.uk

B. Kitchenham
School of Computing & Maths, Keele University,
ST5 5BG, Staffordshire, UK
e-mail: b.a.kitchenham@cs.keele.ac.uk

only 3 from 19 conventional abstracts achieved this level, while only 3 from 20 structured abstracts failed to reach it. We conclude that the use of a structured form for organising the material of an abstract can assist inexperienced authors with writing technical abstracts that are clearer and more complete than those produced without the framework provided by such a mechanism.

Keywords Evidence-based · Systematic literature review · Structured abstract

1 Introduction

A key requirement for Evidence-Based Software Engineering (EBSE) is the ability to be able to find, evaluate and aggregate all of the appropriate sources of evidence. In particular, the evidence-based paradigm is one that relies heavily upon the use of *systematic literature reviews* to assemble the (empirical) evidence that is needed to address a research question (Kitchenham 2004; Webster and Watson 2002). A *secondary* study such as a systematic literature review requires exhaustive searches of the literature in order to identify potentially relevant *primary* studies. Such searches involve two stages: firstly researchers need to perform a wide search to identify as many candidate primary studies as possible; secondly they must undertake a more detailed review of these candidates against specific inclusion and exclusion criteria. Indeed, the first step of the search process is very likely to identify a large number of studies, of which many will actually be irrelevant.

Current procedures, based on experience from clinical medicine, suggest that a review of the title and abstract of a primary study should be sufficient to enable the researcher to determine whether or not it is relevant to the study being undertaken (Kitchenham and Charters 2007). However, recent attempts to conduct systematic literature reviews in the domain of software engineering have reported difficulties with identifying whether or not primary studies are relevant to a topic of interest. This is because the information provided in abstracts is often incomplete, with the effect that the researchers find it necessary to read other parts of the paper to determine whether or not it is of interest (Brereton et al. 2007; Dybå and Dingsøyr 2008).

One approach to improving the standard of abstracts that has been adopted in medicine and in other domains such as psychology is to adopt the use of *structured abstracts* (Hartley 2004). The results of empirical studies conducted in Educational Psychology suggest that structured abstracts are a potentially valuable approach to improving the readability and value of abstracts. Hartley (2003), and Bayley and Eldredge (2003) also identify other benefits of adopting this form to help improve the design of studies.

To investigate whether the same improvement could be obtained if structured abstracts were to be adopted in software engineering, the members of the EBSE project conducted two studies. One was an observational study, based upon using the abstracts presented for the 2004 and 2006 EASE conferences (Evaluation and Assessment in Software Engineering) that measured length and readability (Kitchenham et al. 2008). The second was a controlled randomised laboratory experiment, that assessed completeness and clarity, with the outcomes being reported in Budgen

et al. (2008). Taken together, the results from these clearly demonstrate the benefits of using structured abstracts in terms of both completeness and clarity. Subsequently, a number of conferences and workshops have begun to require structured abstracts, and one journal (*Information & Software Technology*) has also adopted their use.

All of these previous studies have concentrated on the task of *extraction* of information from abstracts. However, the primary purpose of the study reported here was to study improvements in the *generation* of information in technical abstracts, for which the only comparable study known to have been undertaken was that of Hartley et al. (2005), using abstracts produced by first year psychology students. For the study reported here, we were able to make use of the early adoption of this form by Durham University's Computer Science Department, in order to examine whether inexperienced (student) authors could produce better abstracts for technical papers, in terms of completeness and clarity, when these are in a structured form. As such, a secondary purpose was to seek evidence that could be used to help persuade the software engineering education community to teach students about the use of structured abstracts. The research question addressed was therefore:

Are inexperienced authors likely to produce clearer and more complete abstracts when they use a structured form?

Based upon this, we were able to formulate the following two hypotheses that were then investigated in this study.

- *Null Hypothesis 1:* Structured and conventional abstracts written by inexperienced authors are not significantly different with respect to completeness.
- *Alternative Hypothesis 1:* Novice authors write structured abstracts that are significantly more complete than conventional abstracts.
- *Null Hypothesis 2:* Structured and conventional abstracts written by inexperienced authors are not significantly different with regard to clarity.
- *Alternative Hypothesis 2:* Novice authors write structured abstracts that are significantly clearer than conventional abstracts.

We also had a more informal research question to the effect that:

Do inexperienced researchers prefer to read and write structured abstracts?

In the rest of this paper we discuss the roles performed by abstracts and the ways that they are organised to address these (Section 2); describe the design of our experimental study (Section 3); report how it was performed (Section 4); and provide an analysis of the outcomes (Section 5). In the discussion that follows we position our results against those obtained by others; discuss the threats to validity; and draw some conclusions about the use of structured abstracts for inexperienced authors (Sections 6 and 7). In addition, the reporting structure used for our paper also seeks to conform to the guidelines for reporting empirical studies that are provided in Kitchenham et al. (2002) and Kitchenham (2004).

2 Abstracts

In this section we begin by examining the role and history of abstracts in scientific and technical papers, and why they are important for software engineering papers. We then briefly review the nature and form of structured abstracts, and the rationale for their adoption.

2.1 Roles of Abstracts

Although abstracts are now considered to be a standard element of scientific and technological papers, their inclusion is (mostly) a relatively recent development. For most scientific journals, the inclusion of abstracts only dates from the end of the 1950s and the start of the 1960s (Berkenkotter and Huckin 1995). This may well reflect the rapid expansion of published material in the second half of the twentieth century, leading in turn to the development of abstracting services, and later encouraging researchers to adopt electronic forms of searching for information.

van der Tol (2001) identifies four main purposes for abstracts:

1. Enabling *selection*, so that researchers and practitioners can decide whether an article merits further inspection.
2. Providing *substitution* for the full document, such that for some readers the information needed is provided without the need to read the full article.
3. Providing an *orientation* function in the form of a high-level structure that assists with reading all or part of the article.
4. Assisting with *retrieval* by providing information that is needed by indexing services, in particular, by highlighting the relevant keywords.

Unfortunately, too few authors of software engineering papers seem to be aware of any of these roles. Notably though, Mary Shaw's discussion of how to write a good research paper (Shaw 2003) discusses the role of an abstract in getting a paper accepted and proposes a structure for abstracts that is less formal than a structured abstract, but essentially incorporates the same content.

The evidence-based paradigm places emphasis upon employing an objective and systematic process for deciding which papers should be included in a review (the inclusion/exclusion criteria). When performing the task of making a decision about whether or not to include a primary study in a review, abstracts perform an important role in terms of *selection*. In later stages, they also provide some elements of *orientation* and *retrieval* when performing data extraction (see Kitchenham and Charters 2007 for more details about these activities).

To illustrate the importance of the use of abstracts for *selection*, Table 1 summarises the four-stage process that was followed in the systematic literature review

Table 1 Figures for inclusion/exclusion process, taken from Dybå and Dingsøy (2008)

Stage	Task	Resulting no. of studies
1	Searching journals and conferences	1,996
2	Exclude studies on basis of titles	821
3	Exclude studies on basis of abstracts	270
4	Appraise studies on basis of full papers	36

of agile methods reported in Dybå and Dingsøyr (2008). Deciding whether or not to include a paper is both an important task for a secondary study as well as a potentially time-consuming one, especially when it involves having to read the papers themselves. We note that, when describing stage 3, the authors reported that “we found that abstracts were of variable quality” as well as that “some abstracts were missing, poor and/or misleading, and several gave little indication of what was in the full article”. So this suggests that having access to better abstracts would have considerably reduced the work involved in obtaining and checking the contents of the remaining 270 papers—especially as the final number retained was only 36.

2.2 Abstracts in Software Engineering Papers

As in the example above, various authors have commented on the poor quality of many software engineering and computing abstracts, perhaps reflecting the relative immaturity of the evidence-based paradigm in this field. There is also little indication in the literature that authors give much consideration to indexing services: again, this might reflect the relatively small number of key software engineering journals.

Budgen et al. (2008) reported on the process of rewriting existing abstracts into a structured form. Here too it was noted that many of the abstracts used in the study were incomplete in some way. So overall, the impression is that software engineering authors tend to give low priority to the task of writing an abstract and few give much consideration to how this might be used. The practice adopted by some conferences of placing a limit on the length of an abstract may also be unhelpful. (However, to be fair, the comments about quality in terms of completeness would also apply to many of our own abstracts that were produced before we began studying evidence-based practices.)

So, part of the motivation for this particular study was to investigate whether the quality of abstracts might be improved in the longer term if we were to train our students (as prospective authors) in ‘better’ practices.

2.3 Structured Abstracts

The basic model of the structured abstract is essentially one of providing a ‘standard’ set of headings to assist the author(s) in providing a précis of the key elements of the study. (There is potentially a secondary benefit in that the headings used for writing a structured abstract can act as a ‘prompt’ to ensure that key items are included in the paper. However, this benefit might be difficult to demonstrate empirically.) In terms of the four purposes identified by van der Tol (2001), their use can therefore improve the process of *selection* by ensuring greater completeness of information; can possibly perform some element of *substitution* (a slightly less evident role in software engineering); can aid *orientation*—especially if the structure of the paper is aligned with that of the abstract; and simplify *retrieval* by making clear where relevant information is provided (and also encouraging the authors to provide it).

Table 2 identifies some of the key distinctions between structured abstracts and conventional (non-structured) forms. Obviously a conventional abstract can be

Table 2 Distinctions between structured and conventional abstracts

Feature	Conventional abstract	Structured abstract
Organisation	Informal, single block paragraph of text	Uses explicit headings (often 5) in a specific sequence to organise presentation of information
Length	Unconstrained, usually determined by journal requirements	Average is around 20% longer than conventional abstracts (Hartley 2002); guidelines suggest 2–3 sentences per heading
Content	No standard	Determined by the headings and any guidance provided about the expectations for each heading
Minimum information	No standard	At least minimal coverage of the topics of the headings

structured, but the lack of headings means that the structure is not as visible to the reader.

The adoption of structured abstracts in software engineering has been advocated as one of the means of improving reporting standards for the discipline (Jedlitschka and Pfahl 2005; Jedlitschka et al. 2008). However, almost all previous studies of the effectiveness of structured abstracts (in any discipline) have been based upon:

- Performing assessments that are concerned with data *extraction*;
- Using abstracts that have been rewritten into a structured form, either by the researchers as in Budgen et al. (2008) or, as in the study reported in Hartley and Benjamin (1998), where this task was undertaken by the original authors.

So, a distinctive feature of this study is the use of structured abstracts that have been created naturally in this form, rather than having been rewritten. They were also evaluated by student ‘judges’, who themselves would in turn be expected to write structured abstracts, rather than being evaluated by the researchers.

One issue for structured abstracts is the question of the choice of headings. Although structured abstracts are associated with systematic literature reviews of empirical studies, especially in clinical medicine, for software engineering they have also been used to review topics such as research trends and models. The set used by the Department were the same as those used in Budgen et al. (2008), namely a set of five keywords (background, aim, method, results, conclusions) that can readily be interpreted for a range of study types (and hence of forms of student project). Table 3 describes the guidance given to students as to what might be included under these headings. Since the research method employed for many student projects is *concept implementation* (Glass et al. 2004), then for such projects the students are encouraged to interpret ‘method’ as concept implementation through their design, and the ‘results’ as being the resulting ‘proof of concept’ implementation, along with the outcomes of any evaluation they might perform.

We should note that this set of headings differs slightly from those proposed by Jedlitschka and Pfahl (2005) and Jedlitschka et al. (2008), which being concerned with reporting experiments, also include an explicit heading for *limitations*. We might also note that all of the student projects concerned are expected to have a practical element.

Table 3 Interpretation of the structured headings

Heading	Description provided to students
Background	Previous research or rationale for a study
Aim	Hypotheses/propositions to be tested, or goal of the study
Method	Description of the type of study, treatments, number and nature of experimental units (people, teams, algorithms, programs, tasks etc.), experimental design, outcome being measured
Results	Treatment outcome values, level of significance
Conclusions	Future work, limitations of study

3 Experimental Design

Before beginning the study, two of us produced a detailed research protocol (plan) detailing how we intended to conduct this. The protocol was then reviewed by the third researcher who at that point was not involved in the study.

The experiment was conducted as a repeated measurement design using the participants as a blocking factor, whereby participants were asked to read the abstracts of two different papers, one of which was in structured form, while the other was not. The participants were asked to assess the information content of each abstract. The allocation of abstracts was randomised across participants.

3.1 Populations Studied

This section discusses the population of abstracts, abstract authors and abstract judges employed in this experiment.

3.1.1 Project Report Abstracts

The abstracts used in this study were written as part of the final year project report produced by undergraduate Computer Science and Software Engineering students at Durham University. The final year project is an individual ‘research-style’ task that is performed during academic year and takes up approximately one third of a student’s time. Unlike the previous studies, where the papers selected were all reporting on empirical studies, final year projects do range over a wide variety of forms and topics. At Durham University it is a requirement that all projects should involve some degree of implementation, and hence none were likely to be wholly empirical (or theoretical) in form. Consequently, in terms of ‘research method’, final year projects are usually of the form that Glass et al. characterise as *concept implementation* (Glass et al. 2004).

In academic year 2007/08, the department made a major change in the format of the report that students were required to produce. Prior to that year, the report was formatted as a traditional dissertation, which could well be 60–70 pages long, and included a conventional form of abstract. In that year, the form of the report was changed to be that of a 20-page technical paper (meant to be similar in form and style to a journal paper) using a standardised set of section headings together with a structured abstract.

Our study included abstracts from the reports produced by four cohorts, namely those graduating in the years 2006–2009. In 2006 and 2007, there were 79 project reports with conventional abstracts. In 2008 and 2009, there were 88 project reports with structured abstracts. From these abstracts we selected 20 conventional abstracts and 20 structured abstracts (see Section 3.2)

3.1.2 Abstract Authors

We selected abstracts that had been written by native English speakers. The authors had little prior experience of writing formal reports requiring abstracts, and so were considered to be representative of native-English speaking undergraduate Computing students at most U.K. universities. We should note here that the distinction between the Computer Science and Software Engineering programmes was not considered to be significant for the purpose of this study (it mainly concerns some differences in core and elective modules, but none that would affect writing skills).

Authors of abstracts written in years 2006 and 2007 had no formal training in writing abstracts and produced conventional abstracts. Authors of abstracts written in 2008 and 2009 were all required to produce structured abstracts. They were provided with a small amount of additional training, which consisted of a short session that introduced the idea of structured abstracts; demonstrated how a (well-written) abstract could be reorganised using the headings; and presented the description of the headings listed in Table 3. The interpretation of the headings for individual projects was left as a task for the students.

Since no re-writing of the abstracts was necessary, we did not involve the original authors in any way. (Durham University owns all work produced by students, so no permissions were required.)

3.1.3 Abstract Judges

The judges of the abstracts were undergraduate students at Durham University who had just completed their second year of study.¹ While this group would be expected to produce structured abstracts for their third year project, they had not yet been exposed to structured abstracts as part of their course work. We did not attempt to exclude students whose first language is not English since these participants were representative of Computing undergraduates in most U.K. universities.

The size of this cohort was approximately 60 students and volunteers were sought by sending e-mails (and reminders). No reward was offered for participating. Our study design required a minimum of 20 participants, and although we had hoped to recruit more than that, in the end our study was performed using the minimum number.

3.2 Abstract Selection

We employed 40 abstracts, taking 10 from each of the academic years 2005–2006, 2006–2007 (conventional abstracts) and 2007–2008 and 2008–2009 (structured

¹The period of university study in England is normally three years.

abstracts). These were all available electronically and hence could be accessed relatively easily. Project reports without abstracts were excluded from the selection process.

We aimed for a balance of those classified as computer science or software engineering topics. We also employed stratified sampling based on the final grade awarded to the students, in order to investigate whether students with lower grades are helped more than those with higher grades. (The ‘capstone’ nature of a project, which counts for 20% of the overall degree classification, meant that the project grade was normally very similar to the overall grade.) Our stratification therefore consisted of projects for students who received final awards of first, upper second, and lower second/third (there were too few third class projects for these to form a separate category).² We selected from these three categories in the proportions 3:4:3 for each cohort.

Values of the *Flesch Reading Ease* measure (Flesch 1948) and *Gunning Fog Index* (Gunning 1952) were recorded for all of the abstracts produced by each cohort. While in Hartley and Sydes (1997) the authors note that these metrics ignore many factors, they do also observe that when applied to two versions of the same abstract then the results should provide some indication of whether or not one version might be easier to read than the other. The length of each abstract (in words) was also recorded. These statistics were used to confirm that the abstracts included in the study had no systematic difference from those that were not included.

3.3 Allocation to Participants

The allocation of abstracts to participants was performed by randomising the list of abstracts and assigning them to participants via a pre-defined design. This process is described below:

1. The 20 structured abstracts were assigned a random number between 0 and 1 and sorted on the random number. They were assigned an identifier S_1 to S_{20} , such that S_1 corresponded to the smallest random number and S_{20} corresponded to the largest random number.
2. The 20 conventional abstracts were assigned a random number between 0 and 1 and sorted on the random number. They were assigned an identifier C_1 to C_{20} , such that C_1 corresponded to the smallest random number and C_{20} corresponded to the largest random number.
3. The 20 participants were assigned an identifier from P_1 to P_{20} , based on the order in which they signed up for the study.
4. The abstracts were then assigned to participants, such that P_i was assigned Structured Abstract S_i and Conventional Abstract C_i , with odd-numbered participants viewing the Structured Abstract first, while even-numbered participants viewed the Structured Abstract second. This is illustrated for four participants in Table 4.

²For non-UK readers, British universities usually use three classes for degrees, with the second class being split into upper and lower seconds.

Table 4 Example of allocation process

Participants	Structured first	Structured abstract	Conventional abstract
P1	Yes	S1	C1
P2	No	S2	C2
P3	Yes	S3	C3
P4	No	S4	C4

3.4 Avoiding Bias

The selection of a stratified profile of abstracts was performed by one of us (Budgen) and then the process of assignment to participants was performed independently by another (Burn), who was kept blind to the grades. Data was collected by one of the authors (Burn). Since there was no direct interaction with the participants when they were undertaking the study, the most likely sources of bias were considered to arise from any previous experiences of the participants (which we sought to minimise by the selection process).

We should also note that while we favour structured abstracts, their adoption for the student projects was not instigated by any of us, nor did we develop the short training session provided for the latter cohorts (showing how an abstract could be reorganised into a structured form), although this was presented to the 2009 cohort by one of us (Budgen).

3.5 Questionnaire Design

Data was collected from participants via a questionnaire. The general structure of this survey instrument was based upon that used in Budgen et al. (2008), which in turn was derived from those employed in Hartley and Benjamin (1998) and in Hartley (2003). However, we had to make some changes in order to reflect a key difference with previous studies, since for these the abstracts used were taken from papers that described empirical studies, whereas for student projects, the emphasis is usually upon a *concept implementation* approach. Table 5 compares the questions between those used in Budgen et al. (2008) and those used in this study and shows how they cover the same general areas, allowing for the different abstract structures and study forms. Questions about participants were not considered to be relevant for this study.

In a recent study, Hartley and Betts (2009) have also successfully demonstrated the effective use of a simple yes/no, nine-item checklist for assessing completeness, suggesting that the instrument is unlikely to be unduly sensitive to changes in the detail of questions. We should also add that from the viewpoint of academics who review and mark student project reports, the questions used do address the issues that we would expect to be discussed in such reports.

Since the number of judges was small, and hence encoding their responses was a fairly quick task, we adopted the simple expedient of implementing the questionnaire by using a paper form.

Table 5 Comparisons between data collection instruments

No.	Original form from Budgen et al. (2008)	Form used in this study
1.	Is the rationale for the study reported?	Is the reason for the project clear?
2.	Is the aim/purpose of the study reported?	Is the aim/purpose of the project clear? <i>For any software developed in the project</i>
3.	Is a hypothesis (or hypotheses) provided?	Is the source of the application requirements specified?
4.	Is there any indication of where this study took place?	Is the main functionality of the application defined?
5.	Is the number of participants reported?	
6.	Are the the types of participants (eg students) reported?	
7.	Is any information about the experience of the participants reported?	
8.	Is the skill level of the participants reported?	
9.	Is there any description of how the study was performed?	Is the basic development method defined? <i>For any evaluation of the software</i>
10.	Does it report how the participants were allocated to different tasks or conditions?	Is the evaluation method defined?
11.	Is the way that the data was collected reported?	
12.	Is there any description of the form of analysis performed?	
13.	Are the main results summarised in the abstract?	Are the results of the evaluation specified?
14.	Are actual numbers from the results presented in the abstract?	
15.	Is any statistical information provided about the results?	Are any specific limitations of the evaluation specified?
16.	Are any conclusions drawn?	
17.	Are any limitations of the study identified?	Are any general limitations of the project specified?
18.	Is there any discussion of required future research?	Are any ideas for future work specified?

4 Conduct of the Experiment

In this section we report on the conduct of the experiment.

4.1 Recruitment of the Participants

As noted in the previous section, we recruited students who had completed two years of study. The initial invitation was made verbally by one of us (Budgen) at a workshop session that was held in the penultimate week of the last term to help students plan their project activities for the following year. This was followed up by

an e-mail to the class giving more detail—and pointing out that participation might help them with planning the evaluation of their own project! Students were invited to attend at one of four scheduled laboratory sessions, or to contact one of us (Burn) to arrange a specific time.

With hindsight, this was too close to the end of term, although we did recruit 14 participants. A further e-mailed invitation at the end of the summer vacation then obtained six further participants, giving us the set of 20.

As each student signed up they were given an identity number (P1 to P20) which allowed for anonymity as well as for indexing and checking, and with these being allocated sequentially.

4.2 The Judging Process

This was undertaken in a separate laboratory, supervised by one of us (Burn) who was also available to answer any questions. As indicated, students either turned up to a scheduled session, or arranged one by e-mail.

At the session, each participant was briefed as to the task, and provided with instructions and a data collection form (see below). They were also asked to sign a consent form, as required by the University's ethical procedures.

We estimated that the task should normally take about ten minutes (Hartley and Benjamin 1998 suggest that reading an abstract requires about 3–4 minutes). Although we did not measure the time taken, the presence of one of the investigators meant that we did have some estimates of this, and informally, most candidates did take about ten minutes.

For the judging process, participants were asked to record their review for each abstract using the paper *abstract evaluation form*. Appendix A shows the task description that was provided to participants, and Appendix B describes the set of questions that were used for collecting the review data.

The order of judging is deemed to be important for such studies, and therefore participants were allocated the abstracts in a pre-defined order, as described in Section 3.3. This was controlled by presenting the abstracts using a lap-top in a session that was supervised by one of the team. The participants were presented with each abstract in turn and were not permitted to backtrack. The abstracts were presented to the participants on a lap-top screen as MS Word documents, formatted using a standard layout so that the appearance was consistent for all participants.

After completing the assessment tasks, participants were asked for some demographic information that could be used to assist with analysis. The questions used for this are described in Appendix C.

4.3 Feedback

After completion of the initial analysis, a short two-page summary (essentially a condensed version of the first draft of this paper), including Fig. 4, was posted on the class web-site and made available to all students in the class, not just those who participated.

5 Analysis

The analysis was performed by one person (Kitchenham) who conducted a blind analysis without knowing which group of abstracts were the structured ones.

A simple two factor model was used to assess the impact of structure on the clarity and completeness scores:

$$y_{ijk} = \mu + \alpha_i \times abtype + \beta_j \times participant + \epsilon_{ijk}$$

Where:

- y_{ijk} is the value of the clarity or completeness score for the abstract k in $abtype$ group i (where $i = 1$, meaning structured, or $i = 0$, meaning conventional), $participant$ j (where j denotes the j th participant)
- μ is the mean value of the completeness or clarity score
- $abtype$ takes the value 1 if the abstract was structured, and 0 if it was not
- $participant$ takes the value 1 to $n - 1$, where n is the number of participants
- α_i is the effect on the mean of $abtype$ group i
- β_j is the effect on the mean of $participant$ j
- ϵ_{ijk} is assumed to be Normally distributed with mean 0 and an unknown variance

After analysis using the two factor model, the blinding was removed and the impact of several other factors was assessed (i.e. the order in which abstracts were viewed, the interaction between year the abstract was produced nested within $abtype$ factor, the grade awarded to the abstract author, the reading preference of each participant, and whether or not the participants had previous knowledge of structured abstracts). The more detailed analyses could not be performed blind because the year effect needed to be assessed as a nested factor, which meant that the year could not be concealed.

Fig. 1 Box plot of clarity scores for structured and conventional abstracts

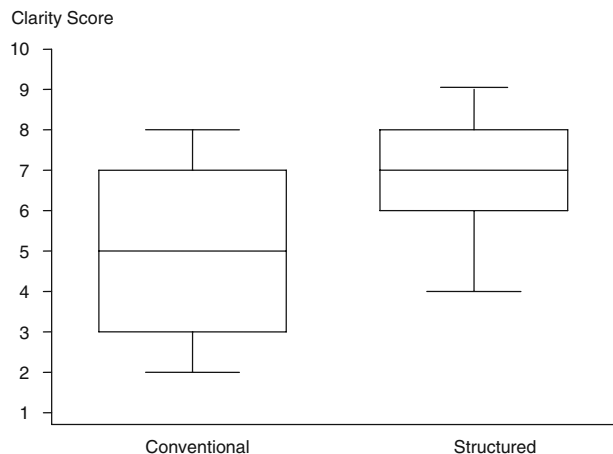


Table 6 Two-way ANOVA for clarity

Source	Partial SS	Degrees of freedom	Mean square	F	Prob > F
abtype	40.0263	1	40.0263	15.84	0.0009***
participant	52.8421	18	2.9357	1.16	0.3767
Residual	45.4737	18	2.5263		
Total	138.3421	37	3.7390		

*** $p < 0.001$

5.1 Clarity

The box plots of the distribution of clarity scores within each group are shown in Fig. 1. These suggest that structured abstracts are better with respect to clarity than conventional abstracts.

A two-way analysis of variance including two factors i.e. “abtype” and “participant” with two groups and 19 participants confirmed that the difference between abstract types was statistically significant ($p \leq 0.001$) and that participant differences were not significant (see Table 6). Note this analysis is based on 19 participants because one participant did not provide a clarity score for either of the abstracts s/he viewed. The analysis is correct in spite of the imbalance caused by the missing data points because the statistical package used to analyse this data (i.e. STATA) does not require the data to be fully balanced. The summary statistics are shown in Table 7.

Various other models were investigated and we found that:

- A two-way analysis of variance including abstract type and the order in which the participant viewed abstracts (i.e. structured first or structured second) found the group differences significant but that the order of viewing structured abstracts was not significant. Note: Overall, the dataset appears to be Normally distributed, but there is always a risk that subsets of the data will be less Normal (i.e. a small sample from a Normal population may not appear very Normal). However, ANOVA is fairly robust unless there are severe differences in variance among subsets of the data, and it also remains the only valid method of analysing the impact of multiple factors.
- A two-way analysis of variance including group and a year within group factor found both factors significant (see Table 8 and Fig. 2). Figure 2 suggests that the clarity scores for year 2 are higher than might be expected for non-structured abstracts. However, the clarity scores are somewhat non-Normally distributed within the four years so this result must be treated with caution.

Table 7 Summary statistics for clarity

Group	Mean	Standard deviation	No. of observations
Conventional abstracts	5.11	1.940	19
Structured abstracts	7.16	1.302	19

Table 8 ANOVA with nested factor for clarity

Source	Partial SS	Degrees of freedom	Mean square	F	Prob > F
abtype	40.6810	1	40.6810	17.11	0.0002***
yearnum × abtype	17.4602	2	8.7301	3.67	0.0360*
Residual	80.8556	34	2.3781		
Total	138.3421	37	3.7390		

* $p < 0.05$

*** $p < 0.001$

Fig. 2 Box plot of clarity scores per year

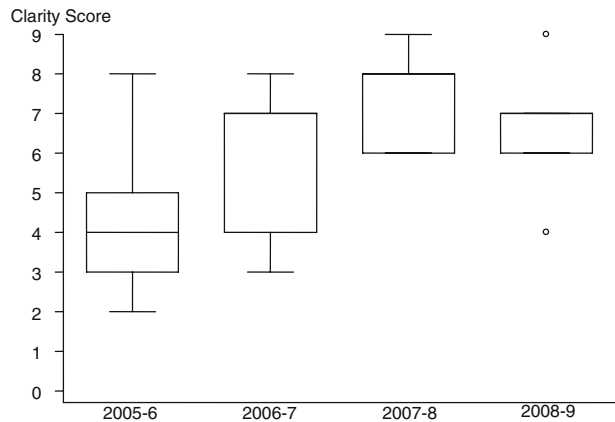


Table 9 ANOVA with two main factors and a nested factor for clarity

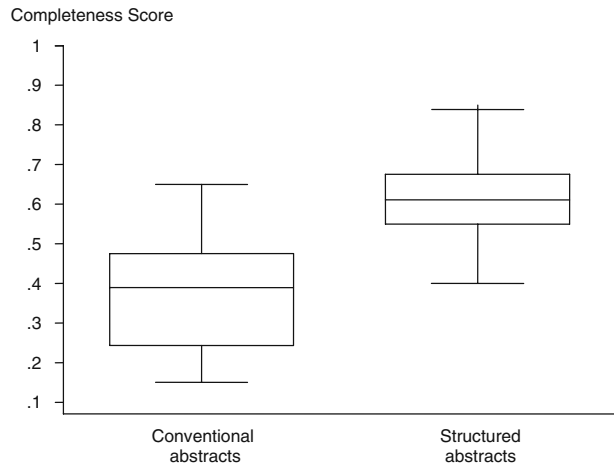
Source	Partial SS	Degrees of freedom	Mean square	F	Prob > F
abtype	40.6810	1	40.6810	20.17	0.0001***
Grade	16.3043	2	8.1521	4.04	0.0272*
yearnum × abtype	16.7677	2	8.3838	4.16	0.0249*
Residual	64.5512	34	3.0172		
Total	138.3421	37	3.7390		

* $p < 0.05$

*** $p < 0.001$

Table 10 Clarity score for different grades

Grade	Structured abstracts			Conventional abstracts		
	Number	Mean	St. deviation	Number	Mean	St. deviation
1 (first)	6	6.17	1.329	6	4.17	1.602
2 (upper second)	7	7.29	1.113	7	5.57	2.225
3 (lower second/third)	6	8	0.894	6	5.5	1.871

Fig. 3 Completeness scores for structured and conventional abstracts

- Adding the factor “grade”, that specifies the final degree obtained by the person who wrote the abstract, shows that grade has a significant impact on clarity (see Table 9).
- We also tested whether the participant’s previous knowledge of structured abstracts or their personal reading preference affected the model. In neither case were these factors significant in the model.

Overall therefore, our analysis of clarity provides clear support for Alternative Hypothesis 2.

The impact of grade is shown in Table 10. Curiously, for both structured and conventional abstracts, those written by individuals who gained a first class degree scored worse than those who were awarded lower degrees. We have no explanation for this observation, other than that these projects might have been intrinsically more complex and hence harder to summarise.

5.2 Completeness

The answers to questions 1–10 were converted into a numerical value (Yes = 1, Partly = 0.5, No = 0). Missing values were treated as not applicable. An overall clarity score was obtained by summing the numerical scores for the 10 questions and dividing the sum by the number of valid answers (i.e. 10–number of not applicable answers).

Table 11 Two-way ANOVA for completeness

Source	Partial SS	Degrees of freedom	Mean square	F	Prob > F
abtype	0.5921	1	0.5921	43.53	0.0000***
participant	0.3797	19	0.0120	1.47	0.2047
Residual	0.2584	19	0.0136		
Total	1.2302	39	0.0315		

*** $p < 0.001$

Table 12 Summary statistics for completeness

Group	Mean	Standard deviation	No. of observations
Conventional abstracts	0.371	0.1445	20
Structured abstracts	0.614	0.1127	20

The box plots for the completeness scores are shown in Fig. 3. These suggest that structured abstracts are more complete than conventional abstracts.

A two-way analysis of variance shown in Table 11 confirmed that the difference between groups is statistically significant ($p < 0.0001$) and that differences among participants is not statistically significant. No other factors were significant when added to the model.

The summary statistics are shown in Table 12. Our overall analysis therefore provides clear support for Alternative Hypothesis 1.

5.3 Relationship Between Clarity and Completeness

Figure 4 shows the relationship between Completeness and Clarity (perhaps inevitably, there are some overlapping data points). There is a statistically significant correlation between the two variables ($r = 0.64$, $R^2 = 0.40$, $p < 0.0001$).

The relationship observed is consistent with that found in Budgen et al. (2008) for improvement in both completeness and clarity. Also, as there is a consistent internal relationship between the completeness and clarity metrics, this provides some further support to the view that our revisions to the data collection instrument produced a valid completeness checklist, even though some of the questions were modified.

Fig. 4 Relationship between completeness and clarity (o indicates a conventional abstract, x indicates a structured abstract) note: overlapping points have been jittered

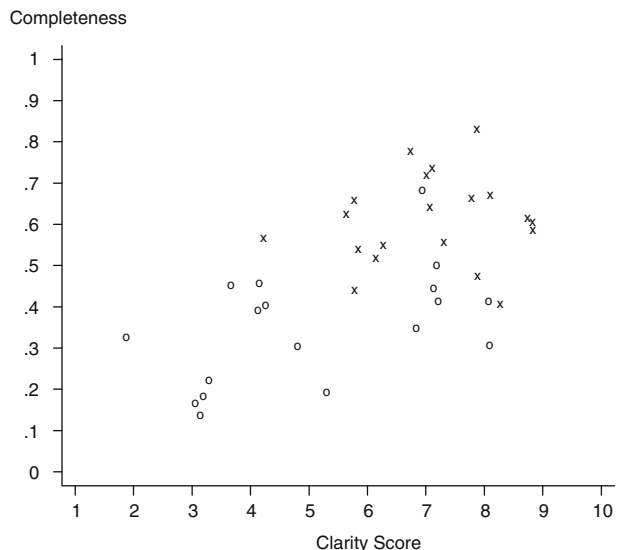


Table 13 Summary statistics for syntactic metrics

Variable	Conventional		Structured	
	Mean	Std. deviation	Mean	Std. deviation
Flesch reading ease	44.465	12.6205	37.355	16.7972
Automated readability	14.735	2.7430	15.915	2.7605
Flesch-Kincaid grade level	12.59	2.3290	13.77	2.5330
Coleman-Liau index	14.11	2.0084	15.18	2.9506
Gunning fog index	16.205	2.4676	17.105	2.6562
SMOG index	13.86	1.7227	14.455	1.9324
Words***	187.1	57.675	287.8	50.1667

*** $p < 0.001$

5.4 Relationship Between Abstract Group and Other Variables

The summary statistics of the syntactic metrics gathered from the abstracts in each group are shown in Table 13. Only the number of words showed a significant difference due to abstract type, with conventional abstracts being significantly shorter than structured abstracts ($p < 0.0001$). Thus, increases in completeness and clarity come at the cost of longer abstracts.

5.5 Individual Questions

The number of answers in each category (excluding missing and not applicable values), is shown in Table 14. For all questions the structured abstracts performed better (i.e. there were fewer abstracts with no information and more abstracts that addressed the topic). However, the differences were only significant for 6 of the 10 questions.

Many of the questionnaires had missing values for some of the answers. We expected some of the answers to be inappropriate for some projects, but we do not know whether the missing values were due to the nature of the project or whether

Table 14 Answers to individual questions for each abstract type

Question	Structured			Conventional		
	Yes	Partly	No	Yes	Partly	No
Q1	15	4	1	12	7	1
Q2***	19	1	0	11	8	1
Q3	5	6	7	0	9	8
Q4	12	6	0	10	6	2
Q5	6	5	7	4	3	10
Q6*	9	6	5	2	8	10
Q7**	13	5	1	4	5	10
Q8*	5	3	12	0	1	16
Q9*	5	2	12	0	2	18
Q10***	8	3	9	1	1	18

* $p < 0.05$

** $p < 0.01$

*** $p < 0.001$

Table 15 Preferences of participants

Read preference	Write preference			Total
	Structured abstract	Conventional abstract	No preference	
Structured abstract	9	2	0	11
Conventional abstract	2	1	1	4
No preference	4	0	1	5
Total	15	3	2	20

the respondents could not tell whether the information was missing or inappropriate. This is a particular problem for conventional abstracts.

5.6 Analysis of Qualitative Data

This section summarises the qualitative data that was obtained from the participants.

5.6.1 Preference for Type of Abstract

The participants were asked to specify which type of abstract they preferred to read and which they preferred to write. The results are summarised in Table 15.

Neither read nor write preference is consistent with a hypothesis of random preference. Looking at the cross-tabulation of reading preference against knowledge of structured abstracts, there seems to be no evidence that participants were influenced by their prior experience (see Table 16).

5.6.2 Points in Favour of Each Abstract Type

The participants were also asked to identify up to three things they liked about conventional abstract and structured abstracts. Kitchenham identified a set of codes for comments. These were agreed by the other researchers (see Tables 17 and 18).

All three researchers coded each comment and were allowed to classify a comment in more than one category.

The initial agreement for 24 conventional abstract comments was:

- 16 classifications were in complete agreement
- 8 classifications were in partial agreement (i.e. the selected categories overlapped)
- 0 classifications were in complete disagreement

Table 16 Reading preference and previous knowledge of structured abstracts

Read preference	Previous knowledge of structured abstracts		Total
	No	Yes	
Structured abstract	6	5	11
Conventional abstract	2	2	4
No preference	3	2	5
Total	11	9	20

Table 17 Categories used to classify comments made about conventional abstracts

Categories	Meaning
Able to report relevant information not covered by headings	The heading for a structured abstract can lead to important information being omitted
Better flow of language	The flow of language in a conventional abstract is more natural
Better English	A conventional abstract is better in terms of use of English
More readable/clearer	A conventional abstract is more readable
Less time consuming	A conventional abstract is short and takes less time to read
Makes authors think	Makes the authors think in more depth about what they want to write
More user friendly	Less formalised and more story-like

Table 18 Categories used to classify comments made about structured abstracts

Categories	Meaning
Clarity/readability	A structured abstract is clear and easy to read
Easy to access relevant information	A structured abstract is easy to skim for relevant information
Structure	The structure is useful in its own right. Structured abstracts are more well-defined/precise/standardised/comparable than conventional abstracts
Completeness	The structure assists writers to report all required material
Easy to write	Structured abstracts are easier to write
More professional	More scientific/professional

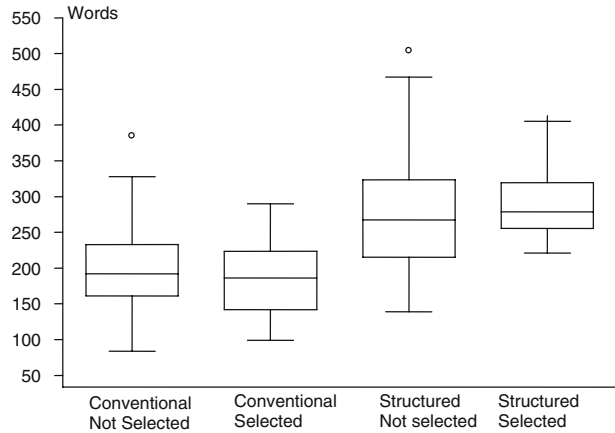
Table 19 Comments in favour of conventional abstracts

Categories	Reading preference			Total
	Conventional	None	Structured	
Able to report relevant information not covered by headings	1	2	2	5
Better flow of language	2	2	3	7
Better English	0	1	1	2
More readable/clearer	2	1	3	6
Less time consuming	0	0	2	2
Makes authors think	0	1	0	1
More user friendly	1	1	1	3
Total	6	8	12	26

Table 20 Comments in favour of structured abstracts

Categories	Reading preference			Total
	Conventional	None	Structured	
Clarity/readability	1	3	5	9
Easy to access relevant information	3	2	9	14
Structure	0	2	5	7
Completeness	0	1	1	2
Easy to write	1	0	0	1
More professional	0	1	1	2
Total	5	9	21	34

Fig. 5 Box plots of word length for structured and conventional abstracts, selected or not for the experiment



The initial agreement for the 31 comments on structured abstracts was:

- 17 classifications were in total agreement
- 14 classifications were in partial agreement
- 0 classifications were in total disagreement

All disagreements were discussed and resolved.

The numbers of comments per category tabulated against the reading preference of subjects are given in Table 19 (for conventional abstracts) and Table 20 (for structured abstracts). Six participants provided no comments in favour of conventional abstracts. Three participants provided no comments in favour of structured abstracts. As can be seen, irrespective of their personal preference, participants were able to identify factors that favoured either type of abstract.

6 Representativeness of Abstracts

The abstracts used in this study were selected (using stratified random sampling) from a set of 167 abstracts comprising 88 structured abstracts and 79 conventional

Table 21 Syntax metrics for structured abstracts not included in the study

Variable	Obs	Mean	Std. dev.	Min.	Max.
fleschre	68	42.563	10.155	12.4	62.4
automate	68	15.20	2.184	11	21.6
fleschki	68	13.025	1.958	9.5	19.1
colemanl	68	14.312	1.668	10.5	19.2
gunningf	68	16.571	2.164	12.6	23
smoginde	68	14.175	1.46831	11.3	18.1
words	68	270.059	77.069	139	504

Table 22 Syntax metrics for conventional abstracts not included in the study

Variable	Obs	Mean	Std. Dev.	Min.	Max.
fleschre	59	43.127	9.707	16.9	60.9
automate	59	15.166	2.520	10.4	23.6
fleschki	59	12.907	2.003	9.3	18.1
colemanl	59	4.354	1.887	8.9	18.3
gunningf	59	16.312	0.263	11	20.9
smoginde	59	13.870	1.525	10.4	17.3
words	59	202.627	61.489	84	385

abstracts. To check that the abstracts used in the study were not systematically different from abstracts not selected for inclusion in the study, we performed a two-way analysis of variance on each of the complexity metrics and the word length using the model:

$$y_{ijk} = \mu + \alpha_i \times abtype + \beta_j \times selected + \epsilon_{ijk}$$

Where:

- y_{ijk} is the value of the complexity or word metric for the abstract k in *abtype* group i (where $i = 1$, meaning structured; or $i = 0$ meaning conventional), *selected* group j (where $j = 0$ meaning not included in the experiment, or 1 meaning included in the experiment).
- μ is the mean value of the complexity metric or word length
- *abtype* takes the value 1 if the abstract was structured and 0 if it was not
- *selected* takes the value 1 if the abstract was used in the experiment and 0 if it was not.
- α_i is the effect on the mean of *abtype* group i .
- β_j is the effect on the mean of *selected* group j .
- ϵ_{ijk} is the random error associated with abstract k in *abtype* group i and *selected* group j . ϵ_{ijk} is assumed to be normally distributed with mean 0 and an unknown variance.

Using the above model, none of the complexity metrics showed a significant effect due to the *selected* variable or the *abtype* variable. Consistent with the experimental effect reported above, the size of the abstracts in terms of number of words was found to be statistically larger for structured than conventional abstracts ($p < 0.0001$), but there was no significant effect associated with the *selected* variable. The distribution of word length for the abstracts is shown in Fig. 5. These results confirm that there was no significant difference in terms of complexity or length between the abstracts included in the experiment and those not included.

For completeness, Tables 21 and 22 provide a summary of measures for the abstracts that we did not use in the study.

7 Discussion

This study has clearly demonstrated the benefit of getting students to write their abstracts in a structured form. To reinforce this point, we suggest that it is reasonable to take 50% clarity and 50% completeness as being the minimum requirements for acceptable abstracts. As Fig. 4 then shows, for those abstracts not using a structured form, only 3 from 19 abstracts achieved this level, whereas when using a structured form, only 3 out of 20 abstracts failed to achieve this level. (This count excludes one conventional abstract that scored more than 50% for completeness, but was not evaluated for clarity.)

7.1 Comparisons with Previous Studies

This section compares our results with previous studies in computing and other disciplines.

7.1.1 Quantitative Results

There have been several previous studies of the completeness and clarity of structured abstracts. In particular, Hartley has published three papers which assessed the completeness and clarity of structured and conventional abstracts in psychology (Hartley and Benjamin 1998; Hartley 2003; Hartley et al. 2005). In a recent field study Sharma and Harrison (2006), confirmed that structured abstracts included more relevant information but suggested that actual improvements were somewhat less than those found in laboratory studies. They found an increase of 12.2% completeness in abstracts from three journals that adopted structured abstract (comparing 50 abstracts from each journal prior to the introduction of structured abstracts with 50 abstracts produced after the introduction of structured abstracts) and no improvement (i.e. a change of -0.01%) in 100 abstracts (covering the same time period) from each of three journals that did not adopt structured abstracts. Also, two previous studies have compared the structured and conventional versions of 25 empirical software engineering abstracts (Budgen et al. 2007, 2008). Note that these studies are not independent because the same set of abstracts were used in both; only the individuals assessing the abstracts changed

The contextual information about all of those studies, as well as this study, are shown in Table 23. The quantitative improvements in completeness and clarity are shown in Table 24. All seven studies showed an improvement in completeness with the improvements being statistically significant in all but one of the studies. The largest improvements were found in studies where the researchers constructed the structured abstracts. Of the four studies where authors of the reports wrote the abstracts, this study showed the largest increase in completeness. Five studies also included a subjective assessment of clarity or quality on a score of 1 to 10. In all cases the structured abstracts scored significantly higher values than the conventional abstracts.

Table 23 Context of related studies

Study	Study type	Person who produced structured abstract	Persons who assessed abstracts	Number of abstracts (S for structured, C for conventional)
Hartley and Benjamin (1998)	Laboratory experiment	Authors of original abstract (retrospectively)	Hartley and psychology students	30 (of each type)
Hartley (2003)	Laboratory	Hartley	Authors of papers published in two educational psychology journals	24 (of each type)
Sharma and Harrison (2006)	Field experiment	Authors of abstracts	Researchers	600 (150S, 450C)
Hartley et al. (2005)	Laboratory experiment with field data	Authors (first year psychology students)	2 graduate teaching assistants (quality), 2 researchers (checklist), 2 experienced staff (quality)	50 (25S, 25C)
Budgen et al. (2007)	Laboratory experiment	Research team members	Research team members	25 (of each type)
Budgen et al. (2008)	Laboratory experiment	As above	64 SE academics, practitioners & students	25 (of each type) same abstracts as above
This study	Laboratory experiment with field data	Authors (third year CS/SE students)	20 students (2nd year CS/SE students)	40 (20S, 20C)

Table 24 Summary of results from this and previous studies showing scores for conventional (C) and structured (S) abstracts

Study	Completeness (checklist)				Clarity/quality		
	No. of items in scale	%C	%S	% point increase	C	S	Point increase
Hartley and Benjamin (1998)	22	29	41.4	12.4	n/a	n/a	n/a
Hartley (2003)	14	39.3	69.3	30.0	6.2	7.4	1.2
Sharma and Harrison (2006)	29	52.2	64.2	12.2	n/a	n/a	n/a
Hartley et al. (2005)	17	50.6	54.1	3.5 (n.s.)	6.1	6.9	0.9 (Q)
					6	7.1	1.1 (Q)
Budgen et al. (2007)	18	31.3	65.5	34.2	5.64	7.08	1.44
Budgen et al. (2008)	18	26.8	64.5	37.3	4.75	7.79	3.04
This study	10	37.1	61.4	24.3	5.11	7.16	2.05

7.1.2 Qualitative Data

In Budgen et al. (2008), participants were asked which type of abstract they preferred. 57 participants answered this question. In this study, participants were asked which type of abstract they preferred to read and which type they preferred to write. The results are compared in Table 25. The results of the previous study (which did not differentiate between reading and writing preference) are mostly similar to the writing preference results found in this study. However, both studies confirm that a majority of (self-selected) participants prefer structured abstracts.

In Budgen et al. (2008) participants were asked what they liked and disliked about structured abstracts. In this study, we only asked what participants liked about each type of abstract. The things participants liked about structured abstracts are contrasted in Table 26. The results show a strong overlap between the two studies although fewer participants in this study considered the use of structured abstracts to assist authors.

In this study we asked what subjects liked about conventional abstracts whereas in Budgen et al. (2008) subjects were asked what they disliked about structured abstracts. Clearly these questions are not directly comparable. However we did notice that some negative points that were made about structured abstracts in the previous study corresponded to positive points made about conventional abstracts in this study (see Table 27). The most significant issues are that participants from both studies consider structured abstracts to be more restrictive than conventional

Table 25 Summary of preferences

Study	Preference			Total
	Conventional	No preference	Structured	
Budgen et al. (2008)	4 (7%)	13 (23%)	40 (70%)	57
Reading preference	4 (20%)	5 (25%)	11 (55%)	20
Writing preference	3 (12%)	2 (8%)	15 (75%)	20

Table 26 What participants like about structured abstracts

Budgen et al. (2008)			This study	
Category	Description	Number of comments	Category	Number of comments
Methodology	Provide guidelines for abstract writer	23	Easy to write	1
Readability	Easier to read, clearer	19	Clarity/Readability	9
Structure	Structure itself is good	18	Structure	7
Search	Easier to locate specific bits of information	17	Easy to access relevant information	14
Completeness	Good summary	12	Completeness	2
Standardisation		9		
Timesaving	Save time reading paper	7		
Paper relevance	Easy to decide whether to read paper	6		
Comparability	Makes it easier to compare results	2		
Conciseness	Makes abstract more concise	2		
			More professional	2
Total		115		34

Table 27 Comparison of problems with structured abstracts and points in favour of conventional abstracts

Problems with structured abstracts	Number of comments	Good points for conventional abstracts	Number of comments
Over-restrictive (forced to use specific headings)	12	Able to report relevant information not covered headings	5
Poor English	3	Better English	2
		Better flow of language	7
Poor readability	2	More readable, clearer	6
Misuse – may stop people thinking	2	Makes authors think	1
Other comments	37	Other comments	5
Total comments	56	Total Comments	26

abstracts and conventional abstracts to be more readable and better written than structured abstracts.

7.2 Limitations

7.2.1 Construct Validity

This study measured completeness in terms of the number of “yes” responses to a questionnaire comprising 10 items and clarity as a numerical value in the range 1–10. Both measures are based on subjective assessments by the participants. In both cases the measures were chosen because they used the same measurement approach as other related studies. However, the completeness checklist was coarser than the instruments used in other studies and included different items (i.e. individual questions) because the student projects were more varied in scope than the empirical studies used in related studies.

In general subjective measures must be treated with some caution. However, problems with the questionnaire should not have caused any major bias in the results since they affect both conventional and structured abstracts equally.

7.2.2 Internal Validity

We have a strong preference for structured abstracts and hence the experimental design and subsequent analysis were carefully structured so as to minimize threats to internal validity. In particular, we used blinding wherever possible in the selection of abstracts, allocation to participants, and analysis. We ensured that participants were randomly assigned to abstracts (i.e. the materials) and to whether they saw a structured and conventional abstract first (i.e. the treatment order). This design should protect against selection bias and learning effects.

We blocked the selection of abstracts to ensure the same number of abstracts were selected from each of the four years. In addition, the selection of the abstracts was based on a stratified random sample from a larger set of abstracts. The stratification ensured that that an equal number of conventional and structured abstracts were selected from authors with different final grades. We were also able to confirm that two uncontrolled factors (i.e. knowledge of structured abstracts, reading preference for structured abstracts) did not bias our results. We also checked that the selected abstracts were similar with respect to word length and syntactic complexity to the abstracts that were not included in the study.

The major remaining threat to internal validity was the number of participants. Based on previous studies, we originally aimed to have 40 participants. With only 20 participants the results should not be over-interpreted.

7.2.3 Generalisability

The abstracts used in this study were a stratified random sample from 167 abstracts including 88 structured abstracts and 79 conventional abstracts. The projects undertaken by the authors of the abstracts are typical of computer science and

software engineering taken by 3rd year students in British universities. However, the nature of the required report differed as well as the format of the abstracts.

The conventional abstracts were associated with a ‘traditional’ project report of up to 50–60 pages whereas the structured abstracts were associated with a short project paper constrained to a maximum of 20 pages. It is therefore possible that the students who produced structured abstracts were able to give more time to abstract construction than students who produced conventional abstracts—although the anecdotal evidence is that writing to a tighter set of constraints is at least as demanding than writing longer reports within the less constrained forms used previously. Furthermore, the student authors were alerted to the importance of the abstract in their project reports because they were given explicit (if somewhat limited) training in the construction of structured abstracts. These influences rather than just the use of structured abstracts themselves might have contributed to the improvement in completeness and clarity compared with conventional abstracts. Thus the improvements found in this study may be somewhat inflated, particularly for the completeness score.

In addition, our study participants constitute a convenience sample not a random sample, so their opinions about structured and conventional abstracts do not automatically generalize to other students. Nonetheless they constitute approximately 33% of the 2nd year students potentially suitable for inclusion in the study, and are the students who will be expected to use structured abstracts themselves when they report their 3rd year projects.

7.3 Interpretation and Implications

Table 23 reports on the results from a diverse set of studies, which were undertaken in several different disciplines, using a variety of experimental approaches. All of these studies confirm that employing a structured form does improve the clarity and completeness of abstracts. Our study confirms that the effect is not restricted to reporting of empirical studies, but we cannot confirm that the effect applies to all forms of research paper. An important outcome of this study is that it has demonstrated that the structured form is usable even by novice authors, with only a minimal amount of training. Furthermore, the results of this and of Budgen et al. (2008) suggest that, although the structured form is not universally preferred by authors, only a minority of researchers would actively dislike using this form.

The main downside of adopting the structured form is that it may lead to longer abstracts (Hartley 2002 suggests this is typically about a 20% increase). Another potential problem is that the headings may be too restrictive. It is also possible that a well-written conventional abstract including all relevant topics would be linguistically preferable to a structured abstract. Nonetheless, the structured form appears to offer a relatively straightforward way to improve the quality of abstracts without relying upon the linguistic capabilities of the author.

One characteristic of this study, and of previous studies, is that they involve the participants in performing an ‘artificial’ task (judging). Implicitly therefore, the results from this study suggest that the greater degree of completeness arising from using structured abstracts will simplify and shorten the task of making decisions about inclusion/exclusion during a systematic literature review (van der Tol’s *selection* activity). In addition, subjects in this study and our previous study noted

that structured abstracts allowed readers to find specific pieces of information (see Table 23), which corresponds to van der Tol's *substitution* activity. Thus, structured abstracts are likely to make it easier for researchers doing systematic reviews to apply inclusion/exclusion criteria from information in the abstract alone, without needing to read the full paper. For a mapping study in particular, a structured abstract may provide all of the data necessary for classification and so substitute completely for the paper. This is particularly so where the mapping study is concerned with identifying the frequency of publications within a pre-defined classification scheme that has been derived using thematic or content analysis. (Dixon-Woods et al. (2005) and Peterson et al. (2008) provide a fuller discussion of the ways in which mapping studies can be organised.)

Generalising beyond our specific study, we suggest that academic institutions should adopt structured abstracts, both in undergraduate project reports, and also in postgraduate MSc or PhD theses. It would improve the abstracts produced by students of all ranges of ability at the cost of a small amount of additional training. We also believe that empirically-oriented software engineering journals and conferences should adopt this form of abstract, because the benefits of structured abstracts have been confirmed empirically, and the limitations are well-understood. In addition, structured abstracts are considered to be good practice in many empirically-based disciplines, and they have many potential benefits for researchers performing secondary studies such as systematic reviews. Finally, if journals begin to adopt structured abstracts (or at least, no longer insist that all abstracts must have a conventional form), novice researchers would find it helpful if they had been introduced to structured abstracts as undergraduates.

8 Conclusions

As described in the preceding sections, we have conducted a quasi-experiment based on assessing the completeness and clarity of the abstracts produced by four cohorts of student authors who were describing their final-year projects. Two cohorts produced conventional abstracts and two produced structured abstracts. Assessment was undertaken by student 'judges' using an instrument derived from that used in similar experiments in which the abstracts described empirical studies.

Our results reinforce previous results in terms of demonstrating that conventional abstracts often omit large amounts of relevant information, whether written by experienced or inexperienced authors, and show that without the structure, many important facets of the studies will not be reported at all (see Section 5.5). In particular, they demonstrate that providing even a very limited amount of training in writing abstracts in a structured form can considerably improve the quality of abstracts produced by inexperienced authors. So in terms of the original research question ("are inexperienced authors likely to produce clearer and more complete abstracts when they use a structured form?") the answer is clearly 'yes'.

In Budgen et al. (2008) it was noted that the improvements arose, in part at least, because the structured abstracts included more information. Indeed, a major argument for adopting the use of structured abstracts is that they provide a framework for doing just this. While in principle this could be achieved by providing more training in writing conventional abstracts, our present study indicates that providing only a

small amount of training in the use of the structured form can produce a significant improvement.

The price of adoption is chiefly a matter of greater length—although as this is part of the consequence of providing more information, it can be argued that this is therefore a very small price. The headings were originally chosen for use with empirical papers, but as this study demonstrates, they are clearly easily adapted to other forms (at least, those forms involved in student projects, which tend to be predominantly concerned with *concept implementation* forms such as tool building).

Overall, our results reinforce those from previous studies as well as employing a more ‘natural’ source of experimental material, since it was not necessary to perform any rewriting of abstracts. Taken together, this argues strongly for the wider adoption by software engineering journals and conferences (as we have previously proposed) and also by educators—since students can obviously benefit by being trained in using this form.

In considering where future work could most usefully contribute, we should note that to date, there have been no empirical studies that have directly assessed the value of structured abstracts in the context of systematic reviews. For example, investigating whether structured abstracts do make the application of inclusion/exclusion criteria easier. It is also an open question as to what specific aspects of the structured form make them more readable, e.g. increased length, grouping information under headings, the logical order of the headings, etc.

Acknowledgements This work was performed as part of the EPIC (Evidence-based Practices Informing Computing) project, funded by the UK’s Engineering & Physical Sciences Research Council (EPSRC). We would like to thank those students who took part in the study as judges, the anonymous referees for their helpful comments and suggestions and Professor Jim Hartley of Keele University for his advice about structured abstracts.

Appendix A: Task Description

The purpose of our study is to investigate how information about student projects in software engineering and computer science can be extracted from the abstracts provided with the final dissertations. You are asked to act as a *judge* for the abstracts taken from two sample dissertations (allocated randomly) and for each one to complete a copy of the *evaluation form* supplied.

To perform the tasks, we ask that you view the two abstracts in the order specified, and that you view each of them on a computer screen, preferably using a browser using a *Mozilla* engine, such as *Firefox*, since the layout provided has been optimised for this. You should complete one form for each abstract—please ensure that you complete *Form 1* first and then *Form 2* as the ordering is important. You may take as long as is necessary to perform the task, but we would not expect that the task should take longer than about ten minutes.

Can you please also complete the third (short) form that will help us classify your input.

David Budgen and Andy Burn.

Appendix B: Abstract Evaluation Form

Registration Code allocated to you:

Number/Title of abstract:

For each of the following questions about the abstract, you should provide one of the following responses *Yes*, *No*, *Unsure* or *N/A (Not Applicable)* by drawing a ring around your chosen response.

- | | | | | |
|----|--|-----|--------|----|
| 1. | Is the reason for the project clear? | Yes | Partly | No |
| 2. | Is the aim/purpose of the project clear? | Yes | Partly | No |

For any software developed in the project:

- | | | | | | |
|----|--|-----|--------|----|----------------|
| 3. | Is the source of the application requirements specified? | Yes | Partly | No | Not applicable |
| 4. | Is the main functionality of the application defined? | Yes | Partly | No | Not applicable |
| 5. | Is the basic development method defined? | Yes | Partly | No | Not applicable |

For any evaluation of the software:

- | | | | | | |
|-----|---|-----|--------|----|----------------|
| 6. | Is the evaluation method defined? | Yes | Partly | No | Not applicable |
| 7. | Are the results of the evaluation specified? | Yes | Partly | No | Not applicable |
| 8. | Are any specific limitations of the evaluation specified? | Yes | Partly | No | Not applicable |
| 9. | Are any general limitations of the project specified? | Yes | Partly | No | |
| 10. | Are any ideas for future work specified | Yes | Partly | No | |

Please give an assessment of the clarity of this abstract by circling a number on the scale of 1–10 below, where a value of 1 represents *Very Obscure* and 10 represents *Extremely Clearly Written*.

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Appendix C: Additional Questions

To assist us with analysing your responses, please provide us with some additional information about yourself and your previous experience. Again, please ring the relevant words where appropriate.

Is English your first language: **Yes / No**

(This is so that we can check whether structured abstracts are more readable for non-native English speakers.)

1. Did you have any knowledge about structured abstracts before taking part in this study? **Yes / No**
If your answer was “Yes”, then please indicate the nature of your knowledge:
 - a. Heard about them, but not seen them before: **Yes / No**
 - b. Read papers with structured abstracts: **Yes / No**
 - c. Created structured abstracts yourself: **Yes / No**
 - d. Other (please specify):
2. Please describe up to three things that you *like* about conventional (non-structured) abstracts.
3. Please describe up to three things you *like* about structured abstracts.
4. Overall, do you prefer to read:
 - a. Structured abstracts
 - b. Conventional (non-structured) abstracts
 - c. No preference
5. Overall, would you prefer to write:
 - a. Structured abstracts
 - b. Conventional (non-structured) abstracts
 - c. No preference

References

- Bayley L, Eldredge J (2003) The structured abstract: an essential tool for researchers. *Hypothesis* 17(1):11–13
- Berkenkotter C, Huckin TM (1995) Genre knowledge in disciplinary communication, chap 2 (News value in scientific articles). Erlbaum, Hillsdale, pp 27–44
- Brereton O, Kitchenham B, Budgen D, Turner M, Khalil M (2007) Lessons from applying the systematic literature review process within the software engineering domain. *J Syst Softw* 80(4):571–583
- Budgen D, Kitchenham B, Charters S, Turner M, Brereton P, Linkman S (2007) Preliminary results of a study of the completeness and clarity of structured abstracts. In: *EASE 2007: evaluation & assessment in software engineering*, BCS-eWiC, pp 64–72
- Budgen D, Kitchenham BA, Charters S, Turner M, Brereton P, Linkman S (2008) Presenting software engineering results using structured abstracts: a randomised experiment. *Empir Softw Eng* 13(4):435–468
- Dixon-Woods M, Agarwal S, Jones D, Young B, Sutton A (2005) Synthesising qualitative and quantitative evidence: a review of possible methods. *J Health Serv Res Policy* 10(1):45–53
- Dybå T, Dingsøy T (2008) Empirical studies of agile software development: a systematic review. *Inf Softw Technol* 50:833–859
- Flesch R (1948) A new readability yardstick. *J Appl Psychol* 32:221–233
- Glass R, Ramesh V, Vessey I (2004) An analysis of research in computing disciplines. *Commun ACM* 47:89–94
- Gunning R (1952) *The technique of clear writing*. McGraw-Hill, New York
- Hartley J (2002) Do structured abstracts take more space? And does it matter? *J Inf Sci* 28(5):417–422

- Hartley J (2003) Improving the clarity of journal abstracts in psychology: the case for structure. *Sci Commun* 24:366–379
- Hartley J (2004) Current findings from research on structured abstracts. *J Med Libr Assoc* 92:368–371
- Hartley J, Benjamin M (1998) An evaluation of structured abstracts in journals published by the British Psychological Society. *Br J Educ Psychol* 68:443–456
- Hartley J, Betts L (2009) Common weaknesses in traditional abstracts in the social sciences. *J Am Soc Inf Sci Technol* 60(10):2010–2018
- Hartley J, Sydes M (1997) Are structured abstracts easier to read than traditional ones? *J Res Read* 20:122–136
- Hartley J, Rock J, Fox C (2005) Teaching psychology students to write structured abstracts: an evaluation study. *Psychol Teach Rev* 11:2–11
- Jedlitschka A, Pfahl D (2005) Reporting guidelines for controlled experiments in software engineering. In: *Proceedings of ACM/IEEE international symposium on empirical software engineering (ISESE'05)*. IEEE Computer Society Press, Los Alamitos, pp 95–104
- Jedlitschka A, Ciolkowski M, Pfahl D (2008) Reporting experiments in software engineering. In: Shull F, Singer J, Sjøberg D (eds) *Guide to advanced empirical software engineering*, chap 8. Springer, Berlin, Heidelberg, New York, pp 201–228
- Kitchenham B (2004) Procedures for undertaking systematic reviews. Technical Report TR/SE-0401, Department of Computer Science, Keele University and National ICT, Australia Ltd, Joint Technical Report
- 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, Pfleeger SL, Pickard L, Jones P, Hoaglin D, Emam KE, Rosenberg J (2002) Preliminary guidelines for empirical research in software engineering. *IEEE Trans Softw Eng* 28:721–734
- Kitchenham B, Brereton OP, Owen S, Butcher J, Jefferies C (2008) Length and readability of structure software engineering abstracts. *IET Softw* 2(1):37–45
- Peterson K, Feldt R, Mujtaba S, Mattson M (2008) Systematic mapping studies in software engineering. In: *Proceedings of EASE 2008*, pp 1–10
- Sharma S, Harrison JE (2006) Structured abstracts: do they improve the quality of information in abstracts? *Am J Orthod Dentofac Orthop* 130(4):523–530
- Shaw M (2003) Writing good software engineering research papers. In: *Proceedings of 25th international conference on software engineering (ICSE'03)*. IEEE Computer Society Press, Los Alamitos, pp 726–736
- van der Tol M (2001) Abstracts as orientation tools in a modular electronic environment. *Document Design* 2(1):76–88
- Webster J, Watson R (2002) Analysing the past to prepare for the future: writing a literature review. *MIS Q* 26:xiii–xxiii



David Budgen is a Professor of Software Engineering in the School of Engineering & Computing Sciences at Durham University in the UK. His research interests include software design, design

environments, healthcare computing and evidence-based software engineering. He was awarded a BSc(Hons) in Physics and a PhD in Theoretical Physics from Durham University, following which he worked as a research scientist for the Admiralty and then held academic positions at Stirling University and Keele University before moving to his present post at Durham University in 2005. He is a member of the IEEE Computer Society, the ACM and the Institution of Engineering & Technology (IET).



Andy J. Burn is a PhD student in the School of Engineering & Computing Sciences at Durham University in the UK. He was awarded a BSc(Hons) in Artificial Intelligence from Durham University, following which he worked as a research associate in the field of evidence-based software engineering. His research interests include software maintenance, software visualization and technology-enhanced learning.



Barbara Kitchenham is Professor of Quantitative Software Engineering at Keele University in the UK. From 2004–2007, she was a Senior Principal Researcher at National ICT Australia. She has worked in software engineering for nearly 30 years both in industry and academia. Her main research interest is software measurement and its application to project management, quality control, risk management and evaluation of software technologies. Her most recent research has focused on the application of evidence-based practice to software engineering. She is a Chartered Mathematician and Fellow of the Institute of Mathematics and Its Applications, a Fellow of the Royal Statistical Society and a member of the IEEE Computer Society.