# Lessons from Conducting a Distributed Quasi-experiment

David Budgen *, Barbara Kitchenham ¶, Stuart Charters †, Shirley Gibbs †, Amnart Pohthong ‡, Jacky Keung § and Pearl Brereton ¶

* *School of Engineering & Computing Sciences, Durham University, Durham, U.K.*
*Email: david.budgen@durham.ac.uk*

† *Department of Applied Computing, Lincoln University, NZ*
*Email: {stuart.charters;shirley.gibbs}@lincoln.ac.nz*

‡ *Prince of Songkla University, Thailand, Email: amnart.p@psu.ac.th*

§ *Hong Kong City University, HK, Email: jacky.keung@cityu.edu.hk*

¶ *School of Computing & Maths, Keele University, Staffordshire, U.K.*
*Email: {b.a.kitchenham;o.p.brereton}@keele.ac.uk*

## Abstract

*Context: Due to the lack of suitably skilled participants, software engineering experiments often lack the statistical power needed to detect the levels of effect that may be encountered.*
*Aim: To investigate whether this can be remedied by running an experiment across multiple sites, organised as a single study rather than as a set of replications.*
*Method: We performed a 'trial' of the idea using a topic (structured abstracts) that some of us had studied previously and which required no participant training. We used five sites, each with 16 participants.*
*Results: We were able to demonstrate the benefits of increased statistical power (and of structured abstracts). We report on our experiences with designing and conducting the study and identify some key lessons about how future studies of this form might be organised.*
*Conclusions: The distributed model offers a flexible, robust form that is capable of delivering better statistical power than would be achieved by running a set of parallel replicated studies.*

## Keywords

*D2 Software Engineering, D2.7c Documentation, D2.9l Review and Evaluation.*

## 1. Introduction

There is a reasonably well-established tradition of conducting empirical studies in software engineering [1], most notably in the form of experiments and quasi-experiments [2]. However, one of the criticisms of such studies as conducted by software engineering researchers is that they are often too small to have the necessary degree of statistical power to detect the levels of effect that are likely to be produced by software engineering practices [3], [4], [5]. (The *statistical power* is the probability that a given test will correctly reject the null hypothesis.) The empirical research community has also begun to recognise the value of performing replicated studies, although there is still debate about how best to organise these [6], [7], [8], [9]. Both of these issues are increasingly important because such 'primary' studies provide the inputs for the secondary studies that form the key tool of the *evidence-based* paradigm [10]. Because secondary studies aim to synthesise the results from many primary studies, they need a sufficient number of these, and also need them to be of good quality and meaningful (which implies having a good level of statistical power) [3], [5].

This paper describes our experience with using a model for performing controlled experiments on a scale adequate for delivering appropriate statistical power. This is achieved by using a number of different sites, each with their own group of participants

978-0-7695-5056-5/13 $26.00 © 2013 IEEE
DOI 10.1109/ESEM.2013.12

and their own experimental material. For software engineering, where experimental studies often require participants who possess particular technical skills and expertise, such people may only be available in small numbers in any one place, this offers a potentially valuable development. In particular, it offers the means of producing results with the degree of statistical power needed for higher quality secondary studies.

In order to investigate the effectiveness of this approach, we undertook an exploratory study based upon an experimental assessment of the effectiveness of *structured abstracts* when used with software engineering papers. A structured abstract (as provided with this paper) is one that makes use of a number of standard headings, and is a mechanism used across a range of disciplines for ensuring that the abstract of an empirical paper reports key aspects of the study [11], [12], [13]. The main reason for our choice of this topic was methodological—arising from the way that conducting secondary studies such as systematic literature reviews necessarily involves the researchers in making a large number of decisions about the inclusion or exclusion of primary studies. If the use of structured abstracts means that more of these decisions can be made on the basis of reading the abstract rather than needing to read the complete paper, then this will greatly improve the systematic review process. A second reason was that we had identified the opportunity to perform a cohort study with abstracts that were written by the original authors of the associated papers, unlike our previous studies that had either involved using structured abstracts constructed by the research team [14] or by student authors [15]. A third reason was that it was practical to undertake this study using student participants, and that no formal skill training was required, making it a convenient test for our ideas.

For this paper, the research question that we address is: *"Is a centralised organisational model suitable for conducting a widely distributed experimental study in software engineering?"*. So, in describing our study, we focus largely upon a retrospective analysis of the organisational aspects.

Although this paper primarily describes our study from a methodological perspective, it is useful to note that we did determine that using structured abstracts improves both the *completeness* and the *clarity* of abstracts, albeit that only the former is statistically significant. Overall therefore, this study reinforced the experiences from our previous studies, as well as the findings of similar studies in other disciplines [16].

The following two sections briefly enlarge upon some relevant issues related to both replicated studies and structured abstracts respectively. We then report on the design and conduct of this study, followed by an evaluation of its organisation and a brief summary of the outcomes (these will be reported more fully elsewhere [17]). Lastly, we identify the methodological lessons that were learned about how to conduct such a distributed empirical study, review the relevant threats to validity, and assess the effectiveness of this model.

## 2. Replication of Experiments

The role of replicated studies, and what precisely constitutes a replicated study is an ongoing topic of debate for software engineering [6], [7], [8], [9]. (As a caveat, in this paper, we are only really interested in replications of human-centric studies.) In particular, we should note that both Sjøberg, and later Juristo and Vegas found that, in software engineering, replications conducted by the same experimenters usually obtain the same results as the original studies, whereas those performed by other researchers do not always lead to similar results [1], [9].

The vocabulary introduced by Lindsay and Ehrenberg [18] is quite widely used when discussing replication, and distinguishes between two main forms.
**Close replication** is not 'identical' (the authors observe that an identical replication would be virtually impossible and of little value). For a close replication the aim is to keep the known conditions for a study "much the same, or at least, very similar" with regard to such issues as population, environment, sampling, measurement and analysis. The value of close replication then lies in obtaining the "same results *despite* the differences", and if this does not occur, then this implies that there is a variable factor that *does* matter. Hence the authors suggest that a close replication should normally precede any differentiated ones, but if this produces the 'same' results, there is little need for further close replications.
**Differentiated replication** involves introducing a deliberate (or at least, known) variation in some relatively major aspects of the study. The purpose of doing this is to explore the boundaries of the conditions within which a specific set of outcomes hold true. In particular experimenters should aim to vary:

- those conditions or factors that might affect the hypothesis being tested
- confounding factors likely to be relevant

Table 1. How replication can be used to address threats to validity

| Threat to validity | How addressed in the physical sciences | How it should be addressed in SE |
|---|---|---|
| *Construct validity:* The study as designed may not lead to outcomes that answer the research question. | A replicated study will normally occur in a different laboratory, using apparatus designed and built by a different team of experimenters. | Any replication (and especially 'close' forms) should be organised by a separate group of researchers working at a different site to the original study, using different materials and subject population, and so requiring an independently-written experimental protocol. |
| *Internal validity:* Relates to the way that a study is conducted and whether there are unknown factors that may influence the results. | This is less of an issue for the physical sciences but it might include flaws in the construction of experimental apparatus, leading to erroneous readings. | For *close* replication a study should be conducted by a team of researchers that is distinct from the original team, using participants who have similar profiles to those who took part in the original experiment. |
| *External validity:* Determines the 'scope' of the outcomes, how widely they might apply. | The greater control of an experiment available in science usually makes it possible to conduct separate series with different materials. | This is really something that is ensured through the use of *differentiated* replications, where larger variations in conditions can be investigated. |
| *Conclusion validity:* Applies to the analysis performed in order to derive an answer to the research question. | Can be achieved by a team conducting their own analysis, and where appropriate, writing their own software for this. | The situation is similar to the general scientific position, in that each team should conduct a separate analysis. |

Lindsay and Ehrenberg also observed that a single study may involve some "internal" differentiation (eg treating morning observations differently from afternoon ones), and indeed, that this really should be seen as essential. However, they also emphasise the importance of differentiated replications, and observe that, as a topic becomes better understood, it is mainly differentiated studies that are needed.

We originally planned to develop a distributed operational model that would allow us to run a set of parallel studies that could be considered as replications, and that could include both close and differentiated forms. However, as our ideas developed, our model evolved into one for conducting a single larger-scale distributed study. Indeed, our concern about replication was the risk that a set of small low-power experiments, that were not sufficiently independent to be synthesised, would do little to advance software engineering knowledge. In Table 1 we explain our ideas about what constitutes replication in software engineering, in terms of how these forms are used to address the widely-used categories of *threats to validity*. We compare our interpretations to those commonly employed in the 'physical sciences' (which of course, do not normally involve human participation, other than as observers).

As we demonstrate in Section 4, our model does not conform to this model of replication in some key particulars: we have used one experimental protocol; one study population; and one allocation process for assigning materials to participants. We therefore consider that it forms a single distributed quasi-experiment, rather than a set of replicated studies. (This form of distributed study is also used for much the same purpose with randomised controlled trials in clinical medicine and related disciplines, where it is usually referred to as a "multi-site trial".)

Related thinking can be found when looking at the concept of *families of studies*. This term was originally coined in a paper by Basili *et al.* [19]. The idea is to employ "a framework that makes explicit the different models used in the family of experiments". Therefore their aim in developing their framework is to enable better use of replication, as a step towards building a body of software engineering knowledge. While this offers a potentially useful concept, it does not explicitly specify a procedural approach for addressing it (in contrast to [9]).

It is also worth noting that even when a study is analysed and presented as a single item, it may in fact represent a 'family' in this sense. An example of this is the study reported in [20], which reports on a family of experiments undertaken at a number of sites, but presents the outcomes as a single entity, rather as we have done with the study reported here.

## 3. Structured Abstracts

This section reports on the previous studies of the use of structured abstracts that have influenced the design of this study. We might usefully also note that all of the studies discussed in this section, did find evidence that, for papers reporting empirical studies, the use of structured abstracts produced an improvement in both *completeness* and *clarity* (where measured) when

compared with the use of a conventional format for the abstract.

The first study, undertaken using software engineering material [14], was a randomised controlled experiment using a set of 25 abstracts from the set of 103 experimental studies identified in [1]. The research team rewrote each of these into a structured format, checking this with the original authors where possible, and then used the resultant pairs in a cross-over study in which each participant acted as a 'judge', assessing two abstracts (one in each format) for completeness and clarity. The 64 participants were a mix of academics, practitioners and students.

The second study took the form of a quasi-experiment [15]. Reports for the computing projects undertaken by students in their final year at Durham University adopted a structured form from 2007/08 onwards. The study used a set of 10 abstracts taken from each cohort in the two years before this change, and 10 abstracts from each cohort in the two years following the change. (It was therefore the 'cohort' nature of the experimental material that made this a quasi-experiment.) The judges were 20 students, and again, each assessed two abstracts, one in each format.

After each of these studies, the first author (Budgen) wrote to the editors of leading software engineering journals, as well as the chairs of major conferences, recommending that they adopt this format for abstracts. Along with a number of conferences and workshops, the journal *Information & Software Technology* did adopt this format, and since the middle of 2010, abstracts for this journal have used a structured format.

The third study that influenced our design was a field experiment conducted by Sharma and Harrison [16]. They looked at a selection of 100 abstracts from each of three orthodontics journals that had adopted a structured format, (150 from before the change, and 150 from afterwards), and compared them with an equal sized selection from three other journals that had retained a conventional format. What they found was that while the structured abstracts had better completeness than the conventional ones (they didn't look at clarity), the extent of the improvement was less than that usually found under 'laboratory' conditions (such as where experimenters rewrote conventional abstracts into a structured format).

# 4. Experimental Design

The original model for the study was to perform a set of replicated studies, with one organisation (Durham University) devising the experimental protocol, producing the experimental materials, and coordinating analysis. Although this involved consultation, it was essentially a centralised model. However, there was also no sense of having an 'original' study plus 'replications', simply a set of parallel studies. These were planned to take place at three university sites: Durham (UK), Lincoln (NZ) and Prince of Songkla (PSU, Thailand) to provide variation across key elements such as sites, experimenters and populations.

However, in thinking about the analysis (and the 'closeness' of all the studies in terms of replication), we decided to revise our design and investigate whether our model could provide the means of conducting larger-scale studies using small groupings of specialist resources that were widely distributed. We therefore included two further sites (Hong Kong and Keele) in order to be able to achieve greater statistical power for the outcomes.

## 4.1. Form of the study

This study formed a quasi-experiment because we selected abstracts from particular volumes of the journals *Information & Software Technology* (IST) and *Journal of Systems & Software* (JSS), and for IST the first block used conventional abstracts while the second used structured abstracts. All JSS abstracts were conventional.

The categories of quasi-experiment identified by Shadish et al. [2] relate to repeated measures using the same participants before and after a change, whereas for us it is the change in experimental material that is of interest. We suggest that this form can be classified as "*a one-group pretest-posttest design with a non-equivalent control group"*. Here the change relates to the transition to use of structured abstracts over the period 2009-2011, and the non-equivalent control group consists of the two blocks of abstracts from JSS for that same period.

We identified three *independent* variables:

1) the *source* of the abstracts (JSS ; IST)
2) the time of publication (*Block1* ; *Block2*) For both journals, these blocks consisted of the issues for a period of roughly eighteen months.

For JSS, the boundary between blocks was based upon issue number (mid-2010). For IST, where the transition from conventional to structured abstracts was gradual, the boundary was across all issues from 2010, dividing them by format of abstract.

3) the location of the study/participants (UK ; NZ ; Thailand; Hong Kong)

The *dependent* variables for the study were measures of *completeness* and *clarity*, assessed using a similar set of questions to those used in [14] and [15].

Other variables that we sought to control, as having potentially confounding effects, were the *experience* of the participants and the time taken to complete the study. For the first, we tried to recruit participants who were at approximately the same level of knowledge and experience, for the second, we recorded the time taken.

For the 'judges' of abstract quality we planned to use undergraduate students studying 'computing' in some form, and who would be at approximately the same level of technical educational attainment and who were either native English speakers or accustomed to working in English for technical purposes. Our design used 16 participants at each site (total 80).

### 4.2. Organisation

The population of abstracts used for the study was the abstracts for empirical papers published in IST and JSS during the years 2009, 2010 and 2011. The relevant volumes and issues were checked by the first author (Budgen), who excluded any papers not considered to be empirical (a category that can include observational papers that report on forms such as *concept implementation* [21]), based on an inspection of the paper's on-line 'preview' (abstract plus section headings). Each of these was then indexed by source, block, and an index variable used for random selection.

Participants were asked to view a series of four abstracts, one selected from each journal/year combination, using a data collection form derived from the one that was used in [15]. We also asked them to provide basic demographic details on another form. After completing an ethical consent form, each participant was to complete the demographics form before starting, and then complete a data collection form for each abstract in turn, and hand it to the experimenter before viewing the next abstract.

In addition to the two data collection forms, we also provided two sets of instructions. The document titled *Replicated Study: Instructions for Experimenters* provided directions for conducting the study at a given site. The *Instructions for Judges* provided information for the participants, and was provided to them at the start of the study. The participants at PSU were additionally provided with translations of the instruction and data collection questions.

Each participant viewed four abstracts. For this purpose, we divided the participants into groups of four who all saw the same set of four abstracts, but in a different order, using the counter-balanced Latin Square shown below.

| | | | |
|---|---|---|---|
| JSS-block1 | JSS-block2 | IST-block2 | IST-block1 |
| JSS-block2 | IST-block1 | JSS-block1 | IST-block2 |
| IST-block1 | IST-block2 | JSS-block2 | JSS-block1 |
| IST-block2 | JSS-block1 | IST-block1 | JSS-block2 |

The aim of this was to ensure that each abstract was viewed four times, once in each position in the sequence and that it also preceded (and was preceded by) each of the others.

We devised a debriefing questionnaire to be completed at the end of the study by anyone who had been involved in conducting part of the experiment at each site. This allowed us to evaluate our design as well as the organisation of the study, and the resulting data provided the material for the 'lessons learned' element of this paper.

## 5. Conduct of the study

Since all of the collaborating researchers were known to one or both of the two lead authors (Budgen and Kitchenham), we chose to conduct the study almost entirely through the use of e-mail. This was used for development of the protocol, circulation of the experimental materials, and collection of results.

The *protocol* went through three revision cycles between February 2012 (version 1.0) and May 2012 (version 1.3). In all, 546 abstracts were identified as meeting our criteria, divided as shown in Table 2. Abstracts were selected using a random number generator, taking into account the size of the block.

The data collection form for each abstract and the abstract itself were organised as a pdf document consisting of two side-by-side A5 images, so that a judge could view both simultaneously and so there could be

Table 3. Divergences that occurred when conducting the study

| Site | How study was organised | Divergences identified | Cause of the divergences |
|---|---|---|---|
| Durham | Envelopes prepared by one experimenter, and used by another who supervised the judging session. | In two cases, judges viewed a pair of abstracts in the wrong order. | Human error in preparing the envelopes. |
| Hong Kong | The experimenter prepared the materials and also supervised the study. | For the first block, the third and fourth abstracts were viewed in reverse order. (Because this block was re-run for other reasons, this was then corrected.) | Human error in preparing the envelopes. |
| Lincoln (NZ) | One experimenter prepared the materials, the second one conducted the judging sessions. | Abstracts were ordered wrongly, using an unbalanced Latin Square. | The local document repository had not been updated to the latest version of the protocol and the instructions in the e-mail were overlooked. |
| PSU (Thailand) | One experimenter prepared the material. In a single session, he and three assistants each managed four judges. | All judges saw the abstracts in the same order. | When making up the material, the earlier e-mail containing the ordering directions and example was accidentally overlooked. |
| Keele (UK) | Both experimenters prepared the materials and data was collected by a junior experimenter. | Some judges returned incomplete forms. | Conflict in the instructions to experimenters. |

Table 2. Available abstracts per 'block'

| Journal | Block 1 | Block 2 | Total included | Total excluded |
|---|---|---|---|---|
| JSS | 132 | 173 | 305 | 277 |
| IST | 110 | 131 | 241 | 80 |
| Total | 242 | 304 | 546 | 357 |

no confusion about which data form belonged with which abstract. For the structured abstracts (IST, Block 2) headings were removed, and in a small number of cases, leading words related to these were also edited from the abstract, so that they were presented to the judge as if they were a conventional abstract.

Following the final revision of the protocol, we conducted the first 'run' at Durham University in June 2012. In order to provide an element of 'dry run' for this, most of the supervision of the sessions with the judge were conducted by an independent researcher who had not been involved in any preparation of materials. Material for these sessions was assembled by the first author (Budgen). The procedure used was to place all of the material for an individual judge in a labelled envelope (ethical consent form; judge instructions; demographic data form; and four abstracts with data forms, numbered 1–4 to indicate viewing order). The person supervising the session would then take the material from the envelope in order, returning as appropriate and noting any issues (including time taken) on the outside of the envelope.

The only significant modification that occurred during the first run was the need to provide some larger-sized data collection forms (A3) for use by one judge who had a vision problem. The process was therefore felt to be suitable for use within the team and so was recommended to the other sites.

The data collection forms were generated separately for each site (four abstracts from each block were used at each site, and no abstracts were used at more than one site). These forms were mailed to each site, along with the demographic forms, the instructions for experimenters and the instructions for judges. The e-mail also included instructions on how to use the Latin Square to make up the set for each judge, along with an example tailored to the site (the abstract ordering for Judges 9-12).

Over the summer, the data from Durham was coded, and in October a copy of the resulting spreadsheet was sent to three of the sites (HK, PSU and Lincoln). The remaining site (Keele) received its spreadsheet later, and this was pre-populated with the codes for the allocated set of abstracts, entered in the order of viewing for each judge.

Because of other commitments and such factors as very different term dates, the final data collection was only completed in January 2013. All sites were asked to return scanned forms to Durham where these were used for data entry or for checking data entry where this was performed locally.

While these procedures were generally effective, some divergences did occur, primarily with organising the sequencing of the abstracts for each judge. These are described in Table 3 where we also explain how each site organised the study and how the divergences arose. None were considered to have any significant affect upon the final analysis (see [17]).

In the cases of human error, the obvious remedy would be to have a better checking system. However,

going beyond that, we can identify three immediate design errors from Table 3.

**Error 1:** All of the *operational* instructions about preparing the materials, including details of the Latin Square, should have been provided in a separate formal document in the same manner as the instructions for experimenters and judges. In addition, documents should have been made available in a shared repository to ensure that versioning problems didn't occur.

**Error 2:** A data collection spreadsheet should have been prepared in advance for each site, providing a means to check the ordering in which abstracts should be viewed by each judge. (This was done for Keele, and proved to be very useful.)

**Error 3:** Instructions for the experimenters should have recognised the possibility of exceptions and identified priorities in handling these. (In the case of Keele, when the experimenter recognised that data was missing, lacking guidance as to how to handle this, he decided to give precedence to the instruction that judges should not be allowed to return to a data entry form once returned.)

The statistical analysis was conducted by one of the two lead authors (Kitchenham) and also revealed some detailed design issues with the data recording process. These were essentially a conflict between the preference for recording the data in forms that could easily be related to the original abstract and block, or the answers to specific questions, and the need for numeric codes that could be used in statistical analysis programs. As a result, the data collected did require quite extensive recoding and associated checking.

## 6. Evaluation

At the conclusion of the study, a short questionnaire was circulated to all sites, with the request that everyone who had helped in any way with the study should complete one. This consisted of 23 questions with the first four being to establish the experience of the respondent; the next four asking about any involvement in the design for the study; 10 questions asking about execution of the study (including about the materials provided); and 5 general questions about the study as a whole. The last two groups particularly asked how aspects of the study could be improved. Nine forms were received, one from everyone involved with the exception of one person who was recovering from an accident. The rest of this section is organised around each of these groups of questions.

**Researcher backgrounds.** These were mainly academic and almost all had prior experience of conducting empirical studies.

**Experimental Protocol.** There were very few comments about this, most had read it, and the section on study design was generally considered to be the most useful, although one person did note that the section addressing the research question was helpful in understanding the design.

**Execution of the study.** Table 4 shows how the *Instructions for Experimenters* and *Instructions for Judges* were rated by the team. It was noted that

Table 4. Ratings for the Instructions

| Instructions for | not very clear | reasonably clear | quite clear | extremely clear |
|---|---|---|---|---|
| Experimenters | 0 | 1 | 4 | 4 |
| Judges | 0 | 0 | 3 | 5 |

the question asking for a participant's age caused some confusion (some put down date of birth), and that this could have been simplified by simply giving a set of age ranges and asking them to tick one. Also, some participants were unsure as to whether to indicate gender in entering their names ('Mr', 'Miss' etc.). One site was conducting the study mid-year, so asking students for their experience in whole years had caused some problems with entering a choice. When asked how many technical abstracts they had read, some candidates were unclear as to what constituted a 'technical' paper. Two sites noted that the part of the instructions for judges that told them what they were to do might have been better expressed as a list of actions that they were to perform, rather than as a paragraph. One site also noted that the instructions for the experimenters might usefully have included an explicit list.

**The overall experience.** Several sites noted that two ideas that proved particularly useful were to put the forms in an envelope for each judge and to have no more than four judges at a time in order to make the process manageable. An additional suggestion was to have a printed checklist to attach to the envelopes (most sites had written one on each envelope). One site questioned the use of a ten-point scale for clarity as students seemed to find it hard to calibrate this, and suggested that a set of terms might have helped ('easily understood', 'some parts unclear' etc.). A further suggestion was for each site to have performed a small 'dry run' so that the experimenters could practise the procedures.

In summary, there were few issues related to the design of the study (the main one was perhaps the point about the clarity scale, which had been adopted for consistency with previous studies). However, there were clearly a number of operational aspects that could have been improved. One site also identified a minor problem that occurred when a participant asked whether the study was about structured abstracts? (Experimenters were asked to avoid identifying the topic, but the instructions did not anticipate this one!)

# 7. Discussion

We begin by examining the threats to validity related to this analysis of the way that we designed and conducted our study (but not for the study itself, which is reported elsewhere). After that, we consider some of the lessons learned from this, in terms of the design, conduct and analysis of a distributed study of this form—and conclude by considering how successfully this approach can achieve the aim of obtaining better power by aggregating limited local resources.

## 7.1. Threats to Validity

In terms of *construct validity*, the main question is whether our post-study questionnaire provided an adequate assessment of all aspects of the way that this study was organised? The questionnaire was devised and checked by the two lead authors (Budgen and Kitchenham), who have prior experience of such exercises, and it was also being used with an experienced group of researchers, who would be likely to identify any significant weaknesses. The other major contribution to this analysis was the log of e-mail messages relating to the study, and we have nothing to suggest that anything was missing from this.

We would suggest that the experience of the group meant that there were no significant threats to *internal validity*, other than perhaps, that our group might have been unrepresentative in some way. One other relevant aspect is that the lead authors had experience of conducting previous studies on this topic.

The issue of the experienced nature of the group might also affect *external validity*, but a more significant factor for this is likely to be the topic of the study itself. The use of structured abstracts is related to research methodology, rather than software engineering practice, and so we cannot be sure that our experiences, and related lessons (below) would necessary apply if the topic of the study were related to software development practices (say). In particular, we should note that this study did not require the participants to undergo any specific training, which is rarely the case for studies of software engineering practices (and as was needed for [20]). Another factor might be the use of student judges, although for this topic there is little to indicate that they are not representative of the software engineering community.

Overall, with some caveats, we consider that our study provides a reasonably sound investigation into our model for conducting a distributed experiment. However, we need to consider carefully how it might be extended to address more software development-related topics that need participant training.

## 7.2. Lessons Learned

Here we examine some of the lessons that we can identify from the material of the preceding sections. For each lesson, we indicate the underpinning factors.

**Lesson 1: Communication in the research team needs shared resources and conversations as well as e-mails.** Our one-protocol-many-instances model is strongly centralised, and the team was distributed across a wide range of time zones, hence we conducted almost all communication by e-mail. Some of the issues that arose, such as using the wrong version of the protocol, clearly stemmed from this. While many of these would clearly have been avoided by using a shared repository of documents, some issues would also have been clearer had there at least been some dialogue to ensure appropriate emphasis was placed upon such issues as the ordering of abstracts. (In [22] the teams made extensive use of interaction to manage a set of close replications, and these seem to have identified some similar issues to those we experienced.)

**Lesson 2: Documentation needs to highlight actions very clearly.** The evaluation identified places where lists had been useful, and some points in the documents that would have been clearer had we used lists to indicate the sequence of actions that need to be undertaken by both experimenter and judges.

**Lesson 3: A centralised model needs to have comprehensive and complete documentation.** We can identify several places where information was conveyed in e-mails rather than in separate documents (the Latin Square model is an obvious one, but issues

such as using envelopes to manage the documents for the judges can also be identified under this category). The instructions also failed to include guidance about how to handle exceptions—while fortunately this was not commonplace in this instance, a distributed model does really require that these be handled consistently at the different sites. There were also issues related to the analysis which needed better planning to ensure that the material was encoded appropriately.

**Lesson 4: Material for use by participants needs to be clear and unambiguous.** This mainly stems from problems with the demographical questions we asked, and our failure to test these adequately with non-native speakers of English. The question about age could have been better organised, and the question about previous experience with reading technical abstracts probably needed a brief explanation of what we meant (reading the abstract to determine whether the article was relevant or worth reading).

**Lesson 5: Each site should perform a local 'dry run'.** This was undertaken at Keele and (indirectly) in Hong Kong, clearly giving the experimenters greater confidence in the procedures.

Overall, these all really point to a specific weakness in our experimental protocol. This was written in much the same way that we would write a protocol for a study being conducted at a single site, and although there was some recognition that additional information would be needed by each site, the issue of communication was not really planned in a systematic manner, by considering the overall experimental process. So this leads to our final lesson.

**Lesson 6: The protocol for a distributed experiment needs to plan communication as a distinct element of the protocol, not simply as a by-product.**

## 8. Conclusions

One to improve our knowledge about software engineering practices is by improving the quality and power of our empirical studies. We have demonstrated that conducting a distributed experiment offers a practical way of addressing this, aggregating the contributions of relatively small number of participants to produce a more significant result, as is shown by the outcomes from the study that we have analysed in this paper. If we had only the 16 abstracts from one site, we would not have been able to achieve a power greater than 0.5 (below the desired value of 0.8), assuming a large effect size and an $\alpha$ value of 0.1. However, for our study, the effect size for completeness can be regarded as being large, and we achieved a power in excess of 0.9 with $\alpha = 0.05$.

We can therefore conclude that the answer to our research question regarding the feasibility of this approach is a conditional 'yes' (in that the topic of the study was methodological and that no participant training was required).

The six lessons identified in the preceding section may be helpful to anyone considering using this model, and there are some other benefits that we can identify for this approach. In particular:

**Flexibility.** We were able to extend the number of judges *and* the associated set of abstracts very easily, making our study less likely to be biased by results specific to particular choice of abstracts or judges.

**Representativeness.** The academic (and developer) communities are international, and this approach meant that our participants did represent an international community, giving better external validity to the outcomes.

**Power.** The need for larger power than was available at individual sites is clear, because if the results were analysed separately for each site, only two of the five data sets would have found a statistically significant effect for completeness.

**Training.** This form of study can be used to help with developing the skills of junior researchers, who can gain first-hand experience with how to conduct an empirical study.

**Widening the Community.** Academics in small departments often have limited opportunity to pursue research, and lack the resources to design and run an empirical study. Collaboration therefore helps to motivate research, facilitates delivery of research-led teaching, and enables knowledge transfer. It also widens the empirical research community and increases its impact.

For future work, there is clearly the need to consolidate the lessons learned with regard to the organisation of such a study. Other issues to investigate include how the model needs to be adapted to studies where some element of training may be required, as well as including the use of practitioners as participants.

## Acknowledgements

We would like to thank those who helped with the running of this study, including Sarah Drummond (Durham), Chris Marshall (Keele), Chutimon Thitipornvanid, Supasit Kajkamhaeng and Wanicbut Wat-

tanamatiphot (PSU). We would also like to thank the 80 participants across the five universities who acted as the 'judges' for us.

## References

[1] D. Sjøberg, J. Hannay, O. Hansen, V. Kampenes, A. Karahasanovic, N. Liborg, and A. Rekdal, "A survey of controlled experiments in software engineering," *IEEE Transactions on Software Engineering*, vol. 31, no. 9, pp. 733–753, 2005.

[2] W. Shadish, T. Cook, and D. Campbell, *Experimental and Quasi-Experimental Design for Generalized Causal Inference*. Houghton Mifflin Co., 2002.

[3] T. Dybå, V. Kampenes, and D. Sjøberg, "A systematic review of statistical power in software engineering experiments," *Information & Software Technology*, vol. 48, no. 8, pp. 745–755, 2006.

[4] V. Kampenes, T. Dybå, J. Hannay, and D. Sjøberg, "A systematic review of effect size in software engineering experiments," *Information & Software Technology*, vol. 49, no. 11-12, pp. 1073–1086, 2007.

[5] O. Dieste, E. Fernández, R. Garcia-Martinez, and N. Juristo, "The risk of using the q heterogeneity estimator for software engineering experiments," in *Proceedings of 2011 International Symposium on Empirical Software Engineering & Measurement (ESEM)*, 2011, pp. 68–75.

[6] J. Miller, "Replicating software engineering experiments: a poisoned chalice or the holy grail," *Information & Software Technology*, vol. 47, pp. 233–244, 2005.

[7] B. A. Kitchenham, "The role of replications in empirical software engineering—a word of warning," *Empirical Software Engineering*, vol. 13, pp. 219–221, 2008.

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

[9] N. Juristo and S. Vegas, "The role of non-exact replications in software engineering experiments," *Empirical Software Engineering*, vol. 16, pp. 295–324, 2011.

[10] B. Kitchenham, T. Dybå, and M. Jørgensen, "Evidence-based software engineering," in *Proceedings of ICSE 2004*. IEEE Computer Society Press, 2004, pp. 273–281.

[11] B. Kitchenham, P. Brereton, S. Owen, J. Butcher, and C. Jefferies, "Length and readability of structured software engineering abstracts," *IET Software*, vol. 2, pp. 37–45, Feb. 2008.

[12] J. Hartley, "Improving the clarity of journal abstracts in psychology: The case for structure," *Science Communication*, vol. 24, pp. 366–379, 2003.

[13] ——, "Current findings from research on structured abstracts," *Journal of Medical Library Association*, vol. 92, pp. 368–371, 2004.

[14] D. Budgen, B. A. Kitchenham, S. Charters, M. Turner, P. Brereton, and S. Linkman, "Presenting software engineering results using structured abstracts: A randomised experiment," *Empirical Software Engineering*, vol. 13, no. 4, pp. 435–468, 2008.

[15] D. Budgen, A. Burn, and B. Kitchenham, "Reporting student projects through structured abstracts: A quasi-experiment," *Empirical Software Engineering*, vol. 16, no. 2, pp. 244–277, 2011.

[16] S. Sharma and J. E. Harrison, "Structured abstracts: Do they improve the quality of information in abstracts?" *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 130, no. 4, pp. 523–530, 2006.

[17] D. Budgen, B. Kitchenham, S. Charters, S. Gibbs, A. Pohthong, J. Keung, and P. Brereton, "Using structured abstracts with computing papers: a distributed quasi-experiment," paper in preparation.

[18] R. M. Lindsay and A. S. C. Ehrenberg, "The design of replicated studies," *The American Statistician*, vol. 47, no. 3, pp. 217–228, 1993.

[19] V. R. Basili, F. Shull, and F. Lanubile, "Building Knowledge through Families of Experiments," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 456–473, 1999.

[20] T. Gorschek, M. Svahnberg, A. Borg, A. Loconsole, J. Börstler, K. Sandahl, and M. Eriksson, "A controlled empirical evaluation of a requirements abstraction model," *Information & Software Technology*, vol. 49, no. 7, pp. 790–805, 2007.

[21] R. Glass, V. Ramesh, and I. Vessey, "An Analysis of Research in Computing Disciplines," *Communications of the ACM*, vol. 47, pp. 89–94, Jun. 2004.

[22] N. Juristo, S. Vegas, M. Solari, S. Abrahao, and I. Ramos, "A process for managing interaction between experimenters to get useful similar replications," *Information & Software Technology*, vol. 55, no. 2, pp. 215–225, 2013.