The Importance of the Correlation in Crossover Experiments

Barbara Kitchenham, *Member, IEEE,* Lech Madeyski, *Senior Member, IEEE,* Giuseppe Scanniello, *Member, IEEE Computer Society,* and Carmine Gravino

Abstract—*Context:* In empirical software engineering, crossover designs are popular for experiments comparing software engineering techniques that must be undertaken by human participants. However, their value depends on the correlation (r) between the outcome measures on the same participants. Software engineering theory emphasizes the importance of individual skill differences, so we would expect the values of r to be relatively high. However, few researchers have reported the values of r.
Goal: To investigate the values of r found in software engineering experiments.
Method: We undertook simulation studies to investigate the theoretical and empirical properties of r. Then we investigated the values of r observed in 35 software engineering crossover experiments.
Results: The level of r obtained by analysing our 35 crossover experiments was small. Estimates based on means, medians, and random effect analysis disagreed but were all between 0.2 and 0.3. As expected, our analyses found large variability among the individual r estimates for small sample sizes, but no indication that r estimates were larger for the experiments with larger sample sizes that exhibited smaller variability.
Conclusions: Low observed r values cast doubts on the validity of crossover designs for software engineering experiments. However, if the cause of low r values relates to training limitations or toy tasks, this affects *all* Software Engineering (SE) experiments involving human participants. For all human-intensive SE experiments, we recommend more intensive training and then tracking the improvement of participants as they practice using specific techniques, before formally testing the effectiveness of the techniques.

Index Terms—empirical software engineering, experiments, crossover experiments, crossover design, repeated measures correlation.

1 Introduction

Crossover designs are frequently used in software engineering (SE) experiments aiming to compare different methods, techniques and procedures proposed for human-based SE tasks [1].

The correlation between two measures made on the same participant in a repeated measures study is exactly the same as the correlation between two different variables measured on the same experimental unit in a regression analysis. I.e., it is the Pearson correlation coefficient and can be calculated using the standard correlation formula. However, in repeated measures experiments, the measures take place at different points in time, and r is calculated somewhat differently to allow for the structure imposed by the experimental design. r plays a critical role in constructing a valid $t-$test for repeated measures designs and the construction of effect sizes and their variances [2]. It is, also, useful to have some *a priori* knowledge of r because it permits pre-experiment power analysis to identify appropriate sample sizes for crossover experiments. These issues are discussed in more detail in Section 2

However, in 12 papers reporting repeated measures studies that we reviewed [3], the value of r was reported only once (see Laitenberger et al. [4]). The 12 r estimates Laitenberger et al. reported came from three experiments and four outcome metrics, and varied between 0 and 0.78, with an average of 0.38. This average is quite low compared with the value of 0.7 that Dunlap et al. reported to be found in test-retest studies [5]. We also found r estimates varying between 0.66 and 0.05 (with a mean of 0.47) when we re-analysed raw data from one family of crossover experiments [6].

Low values of r might imply that there is little performance consistency among participants, i.e., participants who performed well using one technique would not necessarily perform well using another technique. This seems to contradict standard assumptions in software engineering management that there are large and persistent skill differences among software practitioners. For example, the personnel and team capability are the most important cost factor in COCOMO II, with a range of 3.5:1 [7]. Thus, if r values are genuinely low in SE experiments, it suggests either that our assumptions about skilled performance in SE are false or that there is some inherent problem with the use of crossover design in SE. Furthermore, any problem related to skilled performance is a potential problem for *any* experimental design involving human participants performing intellectual tasks.

The motivation for this paper is concern about the valid-

- *Barbara Kitchenham is with the School of Computing and Mathematics, Keele University, Keele, Staffordshire, ST5 5BG, UK. ORCID: 0000-0002-6134-8460*
- *Lech Madeyski (✉) is with the Department of Applied Informatics, Wroclaw University of Science and Technology, Wyb.Wyspianskiego 27, 50370 Wroclaw, Poland.*
  E-mail: Lech.Madeyski@pwr.edu.pl ORCID: 0000-0003-3907-3357
- *Giuseppe Scanniello is with the Department of Mathematics, Computer Science, and Economics, University of Basilicata, Viale Dell'Ateneo n 10, Macchia Romana, 85100, Potenza, Italy.*
  E-mail: giuseppe.scanniello@unibas.it ORCID: 0000-0003-0024-7508
- *Carmine Gravino is with the Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132 - 84084 Fisciano (SA), Italy*
  E-mail: gravino@unisa.it ORCID: 0000-0002-4394-9035

Manuscript received Month XX, 2021

ity of human-centric experiments in SE. Our goal is to investigate the distribution of r values observed in human-based SE crossover experiments and to discuss the implications of our findings with respect to the design of all human-based SE experiments.

In Section 2, we explain (as mentioned before) why r is so important in crossover designs in terms of analysing crossover data, calculating effect sizes and their variances, and underpinning the power advantage of crossover designs compared with between-groups experiments. In Section 3, we identify the main properties of the Pearson correlation coefficient with the help of simulation, and we explain how to calculate r in crossover experiments. In Section 4, we report an empirical study of r values based on 35 experiments reported in 15 studies. We discuss our results in Section 5 and present our conclusions and recommendations in Section 6.

2 The Role of r in Crossover Studies

In this section, we explain the role of r in the analysis of crossover experiments, including the construction of effect sizes and their variances, and its impact on the crossover-experiment power. In the section, we present the basic analysis formulas. The analysis of crossover data is based on the fact that because of the structure of the AB/BA crossover and the four-group crossover, the $t-$test for a crossover is based on the *difference values* for each participant. In the Supplementary Material [8], we explain in more detail how the formula for the $t-$ tests arises from the structure of a crossover design.

2.1 Tests of Significance

In the context of an AB/BA crossover design, the formula for a $t-$test is:

$$t = \frac{2ES}{\sqrt{2s^2(1-r)(1/n_1 + 1/n_2)}} \tag{1}$$

where ES is the difference between the mean outcome for a participant using one treatment and the mean outcome for participants using the other treatment, $2ES$ is the difference between the mean of the difference data in each sequence group, r is the correlation between the measures on each participant taken in each time period, n_1 is the number of participants in sequence group 1, n_2 is the number of participants in sequence group 2, and s^2 is the variance of the response measured on an individual participant,[1] and $2s^2(1-r)$ is the difference data variance. If $n_1 = n_2 = n$, the above equation simplifies to:

$$t = \frac{ES}{s\sqrt{(1-r)/n}} \tag{2}$$

It must be emphasised that although we have two measures from each participant, i.e., $4n$ observations, we still have only $2n-2$ degrees of freedom. The extra measures have increased the *precision* of our sample statistics and provided information about the proportion of total variance related to within-participant variance and between-participant variance, but they have not increased the *accuracy* of our estimates of the *population* statistics.

2.2 The Power of Crossover Experiment

If we had $2n$ participants and undertook a standard between groups experiment with n participants assigned to each group, the $t-test$ would be:

$$t = \frac{ES}{s\sqrt{2/n}} \tag{3}$$

again we have $2n-2$ degrees of freedom.

Comparing Equation 2 and Equation 3, it is clear that with the same number of participants, and the same estimates of ES and s, the crossover design would deliver a $t-$value larger than the $t-$value for the between-groups design, because unless $r = -1$, $(1-r) < 2$. Furthermore, even if $r \leq 0$, we would obtain a larger $t-$value. This means that, for the same number of participants, the power[2] of the crossover design is greater than the power of a between-groups experiment.

Cohen [9] reported that for a medium standardized effect size (i.e., 0.5) and an alpha level of 0.05, a between-groups experiment would need 64 participants per group to have a power of 0.8. However, from Equation 2 and Equation 3, if $r = 0$, everything else being equal, a crossover design would require only 32 participants per sequence group. Senn [10] points out crossovers require more time and effort on the part of both experimenters and participants. He provides a more realistic discussion of the comparison between crossovers and between groups designs that still strongly favours crossover designs (see [10], Section 9.2). However, he also points out that there are other things to consider when deciding to use a crossover design than just improved power, such as drop-outs, carry-over, inconvenience to participants, and analysis difficulty.

2.3 Crossover Effect Sizes and Their Variances

The calculation of effect sizes and their variances for crossover designs are discussed in detail in [2]. In this section, we summarise the role of r in such calculations.

There are two different standardized mean difference effect sizes of interest in any repeated measures experiment. Firstly, there is δ_{RM}, which is referred to as the repeated measures effect size and measures the average improvement for individual participants. δ_{RM} is estimated as:

$$d_{RM} = \frac{ES}{s\sqrt{(1-r)}} \tag{4}$$

Secondly, there is δ_{IG}, which is referred to as the equivalent independent groups effect size and measure the difference between the two methods:

$$d_{IG} = \frac{ES}{s} \tag{5}$$

It is intended to provide an effect size that is comparable to that obtained from a standard between groups experiment.

1. This assumes that the variance is unaffected by time period or treatment, which is the standard assumption for the analyses of complex statistical designs whether or not repeated measures are used.

2. I.e., the likelihood of detecting a significant effect when the alternative hypothesis is true.

Although we can calculate the value of d_{IG} without knowing the value of r, we need to estimate r to calculate the variance of d_{IG}.

The variance of a standardized mean difference effect size is based on the relationship between the estimate and a valid $t-$variable. Since d_{RM} is directly related to a $t-$variable (see Equation 2), but d_{IG} is not, the variance of d_{IG} can only be estimated by considering the relationship between d_{RM} and d_{IG}. From Equation 4 and Equation 5, we can see that:

$$d_{IG} = d_{RM}\sqrt{(1-r)} \tag{6}$$

Thus, the variance of d_{IG} is obtained by multiplying the variance of d_{RM} by $(1-r)$. If the number of participants in each sequence group is the same (i.e., n) and n is not small, the normal approximation of the variance of d_{RM} is:

$$var_{dRM} = \frac{1}{n} + \frac{d_{RM}^2}{2 \times f} \tag{7}$$

where f is the number of degrees of freedom which will be $2(n-1)$ for a crossover design, but, assuming n is relatively large, is often replaced by the term $f = 2n$. Then, the variance of d_{IG} is:

$$var_{dIG} = \frac{(1-r)}{n} + \frac{d_{IG}^2}{4n} \tag{8}$$

Thus, as well as being essential for statistical tests, r also plays a critical role in defining crossover effect sizes and their variances.

3 The Between Participant Correlation and Its Properties

In this section, we explain how to calculate r for individual sequence groups, and we demonstrate the basic properties of r with the help of a simulation study.

3.1 Estimating the Value of r

As mentioned previously, r is the Pearson correlation coefficient, so for the pair of values from each participant in a specific sequence group, we could use the equation:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \times \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{9}$$

where x_i is the measure obtained in time period 1 for participant i in a specific sequence group and y_i is the measure obtained in time period 2 for participant i, and there are n participants in the sequence group.

Equation 9 confirms that r is unaffected by differences in the mean values of x and y. In the context of a crossover, when we measure the same attribute (e.g., response time to complete a SE task or the correctness of the task outcome), r is unaffected by whether or not x and y are significantly different. Also, if we measure the same response attribute, we expect the variance of x and the variance of y to be estimating the same underlying variance, i.e., σ^2. The best estimate of the σ^2 is the average of the variance of x values (s_x^2) and the variance of the y values (s_y^2), i.e., $s^2 = (s_x^2 + s_y^2)/2$, so Equation 9 becomes:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{(n-1)s^2} \tag{10}$$

However, in the context of crossover experiments r is usually calculated somewhat differently using the relationship between the variance of the x_i values, the variance of the y_i values and the variance of the difference values s_{diff}^2, which gives the following equation for the exact correlation estimate (r_e)

$$r_e = \frac{(s_x^2 + s_y^2 - s_{diff}^2)}{2s_x s_y} \tag{11}$$

Again, if we assume $s_x^2 = s_y^2 = s^2$, we can calculate r based on the average variance, i.e., the pooled correlation estimate (r_p), and we have:

$$r_p = \frac{(2s^2 - s_{diff}^2)}{2s^2} \tag{12}$$

This form of the equation is useful when repeated measures analysis tools are used, because they usually report the best estimates of s^2 and the within-participant variance, i.e., $s_e^2 = s_{diff}^2/2$ for the full data set. Also, r_p and r_e can sometimes be calculated from reported descriptive statistics, even when the raw data are not available. We present a worked example of estimating r_e, r_p and r_{exp} (which is the estimate of r for all the participants in a single experiment) in the Supplementary Material [8].

3.2 The Basic Properties of the Correlation Coefficient

In this section, we recap some of the basic properties of the Pearson correlation coefficient as a parameter of the bivariate normal distribution. We illustrate these properties using simulation studies, all of which were obtained using the `rSimulations` function available in our `R` package `reproducer` [11].

We simulated bivariate normal distributions with the means of the two variables specified by μ_1 and μ_2, the variances being specified by σ_1^2 and σ_1^2 and the correlation between specified by ρ. For each sample size N, we obtained 10000 samples where each set of simulations was initiated with a different seed value. We calculated the value of r for each sample. Then, for each set of r values, we calculated the mean, median, and variance of the r estimates. We also calculated variables related to the accuracy and stability of the variance estimates. The variance proportion (VP) metric measures the extent of variance stability:

$$VP = \frac{s_1^2}{s_1^2 + s_2^2} \tag{13}$$

where s_1^2 is the estimate σ_1^2 and s_2^2 is the estimate σ_2^2. If $\sigma_1^2 = \sigma_2^2$ and $VP \approx 0.5$, this indicates variance homogeneity, if $VP < 0.25$ or $VP > 0.75$, then there is a $3:1$ difference between the variances and we considered this to be an indicator of substantial variance instability. We classify VP values outside the range as anomalies.

The results reported in Table 1 show the r statistics and the VP statistics for difference sample sizes and are supported by graphical representation of the distribution of r estimates shown in Figure 1 which are based on sample sizes of 1000.[3] The left panes of Figure 1 show a scatter plot

3. We have reduced the number of simulations for plotting, because too many observations can make it difficult to assess the distribution of scatter plots. In contrast, a large number of simulations are required to provide confidence in the results of investigating mean and median bias in r estimates.

of the r estimates plotted against the VP values for samples of size 30. The right panes show box plots of the r estimates for sizes 10, 20, 30 and 60. The top, middle, and bottom panes show the effect of different mean values and different variances.

Equation 12 shows that r is functionally related to the participant variance and difference data variance, so we also investigated the impact of the accuracy of these variances. The $VarAcc$ metric measures the accuracy of the participant variance estimates:

$$VarAcc = \frac{s_1^2 + s_2^2}{\sigma_1^2 + \sigma_2^2} \tag{14}$$

If $VarAcc \approx 1$ this is an indication that estimates of the variance are accurate. If $\sigma_1^2 = \sigma_2^2$ but $VarAcc < 0.5$ or $VarAcc > 1.5$, we considered this to be an indicator of substantial variance inaccuracy. We classify accuracy values outside this range as anomalies. $VarAcc$ has some inbuilt bias because its lower values are bounded but its upper values are not. In addition, it is not symmetric about 1 in terms of the standard deviations (s_1 and s_2). However, we consider it a reasonable heuristic for the purpose of comparing the extent of instability across different sample sizes.

The $DiffVarAcc$ measures the accuracy of difference values variance estimates:

$$DiffVarAcc = \frac{s_{diff}^2}{(\sigma_1^2 + \sigma_2^2) - 2\rho(\sigma_1 \times \sigma_2)} \tag{15}$$

It has similar properties to $VarAcc$ and is assessed in the same way.

$VarAcc$ and $DiffVarAcc$ statistics are reported in Table 2. Figure 2 displays box plots that show the relationship between sample size and r, VP, $VarAcc$ and $DiffVarAcc$. These box plots are based on 1000 replications of simulated data sets of size N=10, 20, 30, 60, 120, and 250 with $\rho = 0.25$, $\mu_1 = \mu_2 = 0$ and $\sigma_1^2 = \sigma_2^2 = 1$.

From these tables and graphics, we can summarise the basic properties of r:

1) r values are slightly biased for small sample size. The first row in Table 1 shows the results of simulation with N=5, with $\rho = 0.25$, $\mu_1 = \mu_2 = 0$ and $\sigma_1^2 = \sigma_2^2 = 1$. The average of r estimate for the 10,000 simulations was 0.22, and the median r was 0.285. The next three rows of Table 1 confirm that as N increases, the bias decreases.
2) The variance of r is large for small sample sizes. The first four rows of Table 1 show the average variance for different sample sizes. As the sample size increases, the variance of r decreases, see also the right panes of Figure 1).
3) For small sample sizes and relatively small ρ, negative estimates of r are not unusual, see Figure 1
4) For small sample sizes and relatively small ρ, estimates of the sample variance are likely to be unstable. 30% of estimates of the variance of individual participants, obtained when the underlying variance was the same and sample size was 5, were different by order of $3:1$. See also the upper two left panes of Figure 1.
5) r is unaffected by the mean values of each variable. Row 4 in Table 1 shows the summary statistics for simulations with $N = 30$ and $\mu_1 = \mu_2 = 0$, row 5 shows a set of simulations with $N = 30$ $\mu_1 = 0$ and $\mu_2 = 1$. Although there is a difference between the means for row 5, there is only an insignificant difference between the average, median and variance of the r estimates in row 4 and row 5. This confirms that ρ is independent of the values of μ_1 and μ_2, which also implies that r is unaffected by whether or not μ_1 is significantly different from μ_2, see also the middle panes of Figure 1.
6) r is unaffected by variance heterogeneity. Row four of Table 1 shows a set of simulations with $\sigma_1^2 = \sigma_2^2 = 1$. Row six shows a set of simulations with $\sigma_1^2 = 1$ and $\sigma_2^2 = 3$, which gives an expected VP=0.25. Although there is a difference between the variances in the rows that is reflected in the different values for the mean of the variance proportion, there is only an insignificant difference between the average, median and variance of the r estimates. This confirms that ρ is independent of the values of σ_1^2 and σ_2^2. Figure 1 confirms that the distribution of r estimates is not affected by variance instability, whether it is due to small sample sizes or actually variance heterogeneity.
7) Row eight shows the impact of a sample size of 60. There is little difference between the mean and median of the r estimates for sample size 30 and 60. Furthermore, the average variance of the r estimates has halved and percentage of negative values and percentage of variance anomalies have both substantially decreased. Nonetheless, Figure 1 confirms that we can still expect a wide variation in r estimates from a single sample.
8) Table 2 and Figure 2 confirm that estimates of participant variance and the difference variance can be very inaccurate for small sample sizes but, like estimates of r and VP, become more accurate as sample sizes increase.

4 An Empirical Study of Within Participant Correlation in SE Experiments

This section reports an analysis of data from 35 crossover design experiments reported in 15 different papers shown in Table 3.

4.1 The Goals of Our Study

Our study is an *investigatory study*. We have used the data generated by previously undertaken experiments and did not collect any new data, so we do not have any formal hypotheses to test. We do, however, have issues that we want to investigate, in particular:

- G1: The magnitude and distribution of r over a relatively large data set, and the relationship between r and sample size. It is important to discover whether the values of r are low and, if so, whether low values are found for all sample sizes. Larger

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2021.3070480, IEEE Transactions on Software Engineering

TABLE 1
Basic Correlation Properties

ρ	N	μ_1	μ_2	σ_1^2	σ_2^2	Mean r	Median r	Variance r	% Negative r	Mean VP	Variance VP	% VP Anomalies
0.25	5	0	0	1	1	0.220	0.285	0.233	31.800	0.496	0.048	29.950
0.25	10	0	0	1	1	0.232	0.256	0.100	23.630	0.499	0.023	10.660
0.25	20	0	0	1	1	0.248	0.261	0.046	12.980	0.498	0.012	1.670
0.25	30	0	0	1	1	0.246	0.252	0.031	9.030	0.500	0.008	0.290
0.25	30	0	1	1	1	0.245	0.252	0.030	8.780	0.500	0.008	0.410
0.25	30	0	0	1	3	0.245	0.255	0.030	8.420	0.256	0.005	6.530
0.25	60	0	0	1	1	0.247	0.251	0.015	2.690	0.500	0.004	0.000

TABLE 2
Variance Accuracy Statistics

ρ	N	μ_1	μ_2	σ_1^2	σ_2^2	Mean Var Accuracy	Variance Var Accuracy	% Var Accuracy Anomalies	Mean Diff Var Accuracy	Variance Diff Var Accuracy	% Diff Var Accuracy Anomalies
0.25	5	0	0	1	1	1.003	0.268	30.520	1.006	0.506	47.150
0.25	10	0	0	1	1	0.996	0.118	12.390	1.002	0.222	26.790
0.25	20	0	0	1	1	0.999	0.056	3.440	0.994	0.103	11.060
0.25	30	0	0	1	1	1.001	0.037	1.090	1.001	0.070	5.380
0.25	30	0	1	1	1	0.997	0.037	1.100	1.000	0.070	5.160
0.25	30	0	0	1	3	1.000	0.045	1.890	1.046	0.074	6.890
0.25	60	0	0	1	1	1.000	0.018	0.060	1.002	0.034	0.800

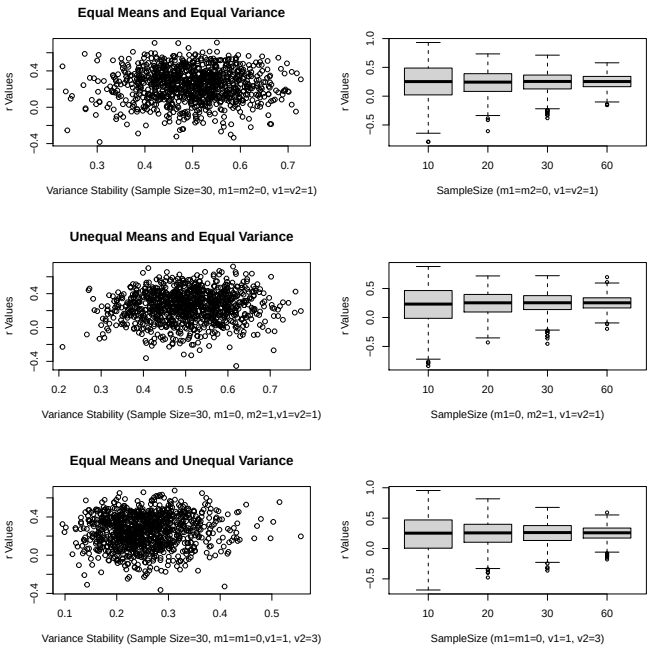


Fig. 1. The Impact of Variance Stability and Mean Difference Values on r

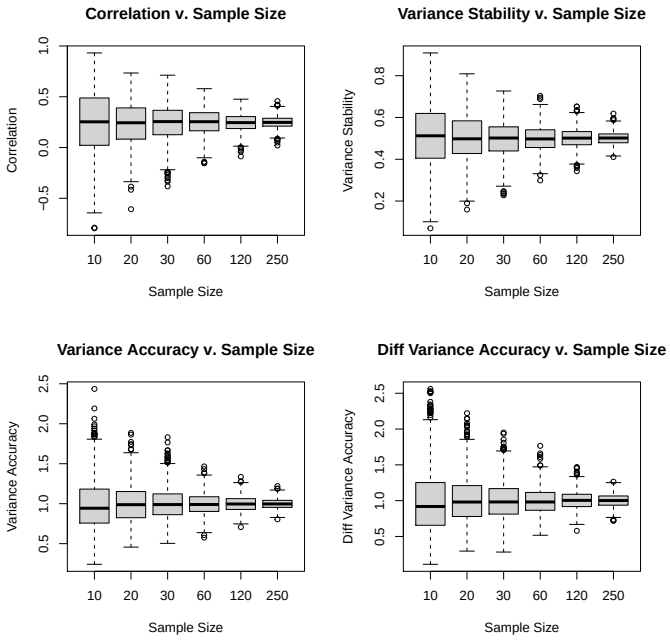


Fig. 2. The Impact of Sample Size on r and Variance Stability and Accuracy

sample sizes should exhibit more stable r values and if the r values for large sample size experiments are larger than those for small sample size experiments, then we do not have any special problem with SE crossover experiments. If, however, we see a relationship similar to that shown in the upper left pane of Figure 2, then we have a situation where r values are consistently small even for experiments with relatively large sample sizes, which is contrary to SE theory and requires further investigation.

- G2: The extent of variance instability and its relationship with r and whether there are systematic trends instability. If we have low r values across different sample sizes, we would like to know whether this can be explained by other properties of our set of experiments, for example, is there any evidence that data from the larger projects is unusually variable. If we see the variance instability decreasing as the size of experiments increases, as in the upper right pane of Figure 2, we can reject the hypothesis that

low values of r for larger experiments are due to unusually large variance instability.

- G3: Whether negative $r-$values are likely to be due to small sample sizes or require some other explanation. Negative r estimates are an extreme example of a situation that contradicts SE theory. They indicate a situation where a participant with a high score on one method has a low score on the other method and vice versa. This strongly contradicts the view of consistent skill differences between software engineers. If we can confirm that the likelihood of negative r values decreases as sample sizes increase, as shown in Table 1, we can be sure that the main cause of negative values is small sample sizes. In addition, if the average r values remain fairly consistent as sample sizes increase, we can have confidence that our estimates of the overall average r value are reasonably accurate. We can then conclude that the disagreement with SE theory is one of the magnitude of the expected effect, not the existence of the effect.

4.2 Study Materials and Methods

This section reports the origin of the data sets used in this study and the basic analysis methods used.

4.2.1 Data Sets

To investigate the distribution of r estimates found in SE crossover experiments in more detail, we calculated r estimates from our own published crossover experiments plus three other papers [4], [12] and [13]. Together, these studies provided data from a total of 930 individual participants, although two papers reported team-based outcome measures which reduces the number of observational units for those papers: Scanniello et al. [14] used 9 four-person teams[4] and Laitenberger et al. [4] used 29 two-person teams in three experiments. We present general summary information about the studies in Table 3, more details can be found in Section 6 of the Supplementary Material [8]. The experimental data for all the studies, except S14 [13] and S15 [4], are available in our `reproducer` package [11], as explained in Section 6 of the Supplementary Material. This will provide a resource for novice researchers wanting to try out various statistical techniques both for analysis of crossover experiments and for meta-analysis of multiple experiment studies.

When multiple experiments were reported in a paper, each experiment addressed the same hypotheses, used the same experimental data, and measured the same outcome variables (metrics). Different experiments reported in a specific paper always involved different participants, and, in most cases, different experimenters. The majority of the experiments used four-sequence group crossover design, and only seven of the experiments used a standard two-group AB/BA crossover design.

We assume that the r values obtained from different metrics are comparable because all are related to the performance of a human-intensive software engineering task.

4. This study also replicated the first experiment a second time using the same participants. We have averaged r-values for the same participants.

TABLE 3
Summary of the Studies in the Data Set

Study ref	Study ID	Num Exps	Num Mets	4G Exps	2G Exps	Partic-ipants
[15]	S1	1	3	1	0	24
[16]	S2	4	2	4	0	86
[17]	S3	5	1	5	0	112
[18]	S4	3	1	2	1	107
[14]	S5	1	2	0	1	36 (9 teams)
[19]	S6	2	2	2	0	87
[20]	S7	2	2	2	0	32
[21]	S8	3	3	3	0	88
[22]	S9	2	2	2	0	39
[23]	S10	1	2	0	1	22
[12]	S11	2	2	2	0	33
[24]	S12	4	3	4	0	100
[25]	S13	1	2	0	1	55
[13]	S14	2	2	2	0	51
[4]	S15	3	4	0	3	58 (29 teams)

4.2.2 Analysis Variables

We calculated the r estimates at two levels of granularity: the sequence group level (i.e., r_e and r_p estimates) and the experiment level (we refer to r estimates at this level as r_{exp} estimates). The sequence group level is important in crossover experiments because each sequence group defines a *cohort* of participants whose performance is measured under the same experimental conditions defined by the time period, treatment and software materials.

The r_e and r_p estimates were generated from the raw data from each experiment. The r-values for each sequence group in each experiment and for each metric are shown in Table 38 in the Supplementary Material [8]. The raw data from the Laitenberger study was not available, so no correlations from that study are included in the sequence level data set.

The experiment level data is shown in Table 39 in the Supplementary Materials [8]. It includes the correlations reported in [4]. However, [4] did not report sequence group variances, nor difference data variances.

From the variances used to calculate r_p, for all studies except [4], we calculated r_{exp} estimates by pooling the sequence group variances for each sequence group for each metric, in each experiment.

At the sequence level and the experiment level, we calculated the variance proportion measure (VP) to investigate variance stability. At the sequence level, we calculated:

$$VP = \frac{Var1}{Var1 + Var2} \tag{16}$$

where $Var1$ is the variance obtained from a specific sequence group and metric in time period 1 and $Var2$ is the variance for the same group and metric in time period 2. At the experiment level, we calculated:

$$VP = \frac{VarPooled1}{VarPooled1 + VarPooled2} \tag{17}$$

where $VarPooled1$ is the pooled variance of the sequence variances in time period 1, and $VarPooled2$ is the pooled variance of the sequence variances from time period 2. These metrics are exactly the same as the VP variable used in our simulations.

4.2.3 Data Analysis

We analysed both the sequence level r values and the experiment level r values, to obtain:

- The basic descriptive statistics of the r_e, r_p and r_{exp} values (i.e., mean, median, variance and standard error) and their distribution based on box plots and histograms.
- The relationship between r_e and r_{exp} values and sequence group size using scatter plots and tabulation. For tabulation, we identified a set of group size categories and calculated the descriptive statistics (mean, median, variance, standard deviation and standard error of the mean) for the r estimates in each category.

The sequence level data and the experiment level data both have analysis limitations, the sequence level has more r values, but they are based on small sample sizes. The experiment level has fewer r values, but they are based on larger sample sizes. We have more confidence in results that are consistent at the two different levels.

4.2.4 Variance Heterogeneity

We used the variance proportion metric at the sequence and experiment level to investigate whether r-estimates were more stable when variances were homogeneous.

4.2.5 Sensitivity Analysis

Our analysis method treated each estimate of r as an independent variable although in each experiment, many of the estimates came from the same group of participants, but were based on different metrics. We performed a sensitivity analysis to assess whether this had introduced bias into our results. The sensitivity analysis used a random effects analysis (REA) which treated r estimates from the same participants, but calculated on different metrics, as repeated values. The full REA results are reported in the Supplementary Material [8]. Specific REA outcomes are reported as part of the main analyses.

4.3 Analysis Results

In this section, we report the results of our analyses. To avoid possible experimenter or analyst bias, the analyses presented in this paper were all performed by the first author who was not involved in the data collection, nor in the experimental analyses reported in the published studies.

4.3.1 Estimates of the Correlations

The descriptive statistics for the r_e, r_p and r_{exp} estimates are shown in Table 4. For r_e and r_p, the mean is less than the median for the sequence level data, which is consistent with the simulation results for small sample sizes. For r_{exp} the mean is greater than the median, suggesting that some unusually large values are inflating the mean. The results obtained from the random effects analysis (REA) of the different r estimates are also shown the Table 4. For each estimate, the REA results are very close to the simple descriptive statistics. However, the REA estimates of the standard error of the mean are slightly larger than the descriptive statistics. The variance of the raw data is less than it should be because the repeated measures r values are slightly correlated, and so are less dispersed than completely independent r values would be. The variance bias is larger for the r_{exp} values than for the r_e or r_p values. Therefore, graphs displaying the distribution of r values will slightly under represent the dispersion of the values. However, the graphs should be accurate enough to highlight any major trends and for assessing the extent to which our results are consistent with the assumptions of the simulations.

The distribution of the r_e and the r_{exp} estimates are shown in Figure 3. As expected, the r_e values are extremely variable confirming that with small samples the values of estimates are very unreliable. The r_{exp} estimates are based on larger samples and have fewer extreme values. We report the distribution of the r_p estimates in [8]. It is similar to the distribution of the r_e.

TABLE 4
Descriptive Statistics of r Estimates

Source	Type	N	Mean	Median	Variance	SE
All data	r.e	249	0.2185	0.3015	0.268	0.0328
REAnalysis	r.e	249	0.2192			0.03502
All data	r.p	249	0.2068	0.2588	0.1964	0.02808
REAnalysis	r.p	249	0.2077			0.02999
All data	r.exp	80	0.2745	0.2464	0.07556	0.03073
REAnalysis	r.exp	80	0.272			0.03522

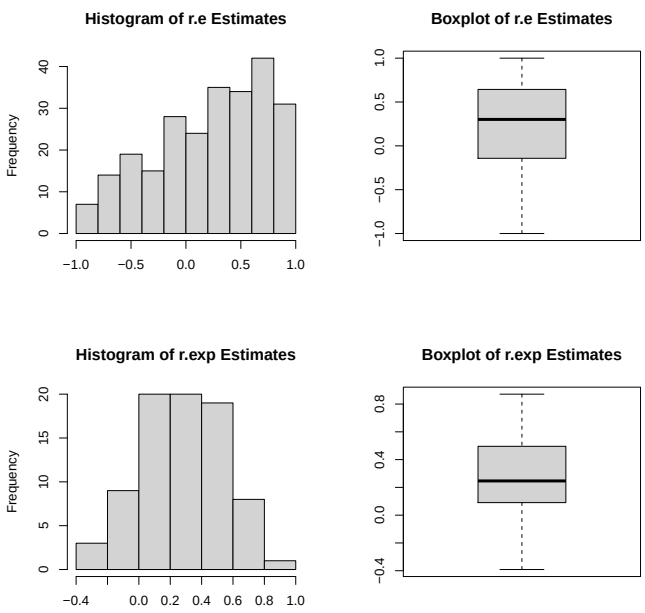


Fig. 3. Distributions of r_e and r_{exp} estimates

4.3.2 The Relationship Between Sample Size and r Estimates

Figure 4 shows the relationship between r estimates and sample size. The upper two panes show the distribution of r_e estimates. The scatter plot shows the r_e estimates plotted against sequence group size, while the box plots are constructed from the seven sequence group size categories

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2021.3070480, IEEE Transactions on Software Engineering

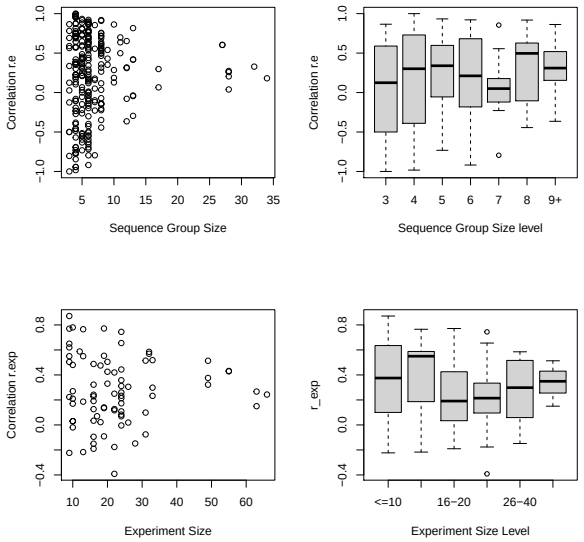


Fig. 4. Relationship between r_e and r_{exp} estimates and Size

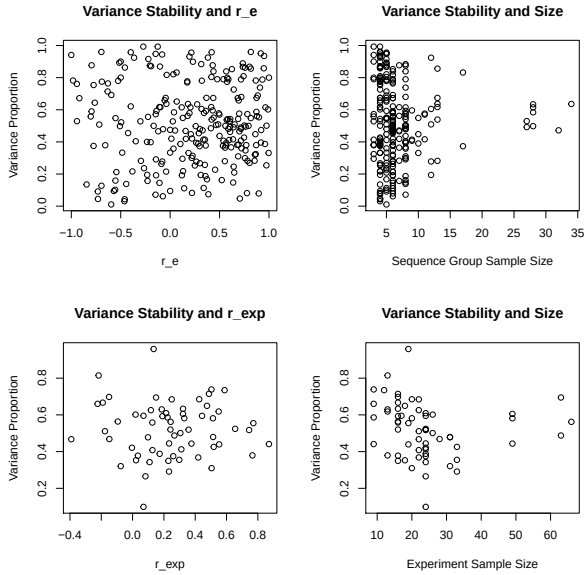


Fig. 5. The Relationship between r estimates, sample size and Variance Instability

specified in the first column of Table 5. The lower two panes show the distribution of r_{exp} estimates. The r_{exp} estimates are plotted against experiment size in the scatter plot. The box plot is based on experiment size categories specified in the first column of Table 6. Figure 4 confirms that small sample sizes are associated with large variation in the observed r estimates both at the sequence group and the experiment level and the variation decreases as size categories increase. In addition, the variation associated with r_{exp} values is less than the variation among r_e values. There does not appear to be any clear increasing or decreasing trend between median r values and the size categories.

A more detailed break down of the r_e and r_{exp} estimates descriptive statistics associated with specific sequence group categories are shown in Table 5 and Table 6, respectively. In addition, we include the mean values from random-effects analysis. The mean r values are all below 0.4, with the r_e means generally lower than the r_{exp} means. Both r_e and r_{exp} analyses suggest a decrease in variance with increasing sample size. Results of the analysis of the r_p values are shown in our Supplementary Materials and are similar to the results for the analysis of r_e.

4.3.3 The Incidence of Variance Instability

Our simulation studies revealed a high incidence of variance instability for small sample sizes, but no evidence that variance instability impacted r values. In this section, we review the stability of variances in our data sets.

Table 7 reports the variance proportion statistics for the sequence group and experiment level data. We also report the percentage of the variance proportion values less than 0.25 and greater than 0.75. Such values indicate a difference of 3:1 in the values of the two variances. For the sequence group data set, over a third of the variance ratios were 3:1 or larger. As would be expected from our simulation study, at the experiment level data, because sample sizes were larger, only 4.4% of values were anomalous. The VP data is based on only 68 correlations because the VP data could not be calculated for Study 15.

In Figure 5, the two left-hand panes show scatter plots of variance proportion against r_e and r_{exp} respectively. It seems that there is no strong relationship between the two variables. In particular, there is *no* evidence that r_e or r_{exp} estimates associated with homogeneous variances were:

1) Larger than estimates associated with heterogeneous variances.
2) Less variable than estimates associated with heterogeneous variances.

The two right-hand panes of Figure 5 show the relationship between variance stability and size. As would be expected, variance stability (shown by variance proportion values close to 0.5), increases as sample sizes increase. All these results are completely consistent with the results of our simulations.

4.3.4 Limitations

A major limitation of this study is that the data sets we analysed were not obtained from either a random sample of experiments nor from a full set of all crossover studies in software engineering. With the only exception of S11 [12], S14 [13] and S15 [4], the experiments considered in this study were all published by authors of the paper. The reason for this is the problem of finding published data sets. Wider adoption of reproducible research would be beneficial for empirical software engineering research [26]. Unfortunately, it is still the case that a few researchers publish their data sets and published data sets are not always maintained. For example, in a mapping study of families of experiment, Santos et al. [27] identified 39 papers, but reported that only six papers provided access to raw data, all of which are included in our analysis. Four were authored by Scanniello and/or Gravino, the other two papers are S11 [12] and S14 [13].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2021.3070480, IEEE Transactions on Software Engineering

TABLE 5
Descriptive Statistics of r_e Estimates for Different Group Sizes

Seq Group Size	Num r_e estimates	Mean	Median	Variance	StDev	SE	REA Mean
3	13	0.081	0.124	0.393	0.626	0.174	0.075
4	63	0.183	0.301	0.395	0.628	0.079	0.184
5	41	0.238	0.340	0.236	0.485	0.076	0.235
6	60	0.186	0.211	0.291	0.539	0.070	0.200
7	9	0.062	0.050	0.222	0.471	0.157	0.038
8	32	0.337	0.496	0.158	0.398	0.070	0.340
>8	31	0.309	0.310	0.089	0.298	0.054	0.305

TABLE 6
Descriptive Statistics of r_{exp} Estimates for Different Group Sizes

Experiment Size	Num r_{exp} estimates	Mean	Median	Variance	StDev	SE	REA Mean
<=10	16	0.370	0.375	0.108	0.328	0.082	0.367
11-15	5	0.374	0.550	0.154	0.392	0.175	0.400
16-20	21	0.216	0.191	0.066	0.257	0.056	0.206
21-25	19	0.210	0.214	0.069	0.262	0.060	0.229
26-40	11	0.265	0.298	0.070	0.265	0.080	0.255
41+	8	0.342	0.349	0.014	0.119	0.042	0.334

TABLE 7
Variance Proportion Descriptive Statistics

Source	N	Mean	Median	Variance	SE	LowerBound	UpperBound	PercentUnstable
Seq Group	249	0.52	0.50	0.06	0.02	0.48	0.55	39.36
Experiment	68	0.52	0.52	0.02	0.02	0.48	0.55	4.41

Another important limitation is that the number of studies with larger sample sizes is small, which casts some doubts on the robustness of our empirical evidence concerning the relationship between r and sample size. However, our simulation studies provide additional support for our empirical results.

A final limitation is that we used the raw data to investigate the distribution of r values although some of the r values were repeated values based on different metrics measures on the same participants. Our random-effects analysis results confirm that the impact on mean values was small, but variance estimates on the raw data are biased towards underestimates. The raw data is essential for visualising the r values distribution, but it slightly underestimate the true variability of the data.

5 Discussion

Our data sets exhibited extremely varied estimates of r and considerable variance heterogeneity at the sequence group level that appeared to be due to the small sample sizes. At the experiment level, r estimates were less variable, but it seemed that estimates of r were affected by sample size with r estimates being inflated for relatively small sample experiments. However, both our analyses and our simulation results provide broadly consistent evidence that the underlying value of r across our set of 35 experiments is between 0.2 and 0.3.

As Senn [10] pointed out, small (or even negative) values of r do not undermine the theoretical power advantage of crossover experiments, so crossover studies are still useful in the context of medical studies. An additional analysis complication with negative r estimates (which was not mentioned by Senn) is that standard analysis tools may behave differently. We provide an example of this problem in the Supplementary Material [8].

However, we believe that small or negative r estimates cast some doubt on the validity of crossover experiments in the context of software engineering studies. The impact of skill differences is built into software engineering management theory and conforms with the industry experience and expectations. So we must ask why the impact of skill seems to be small in our software engineering experiments. Small values of r have a number of possible explanations:

1) Skill may not be an issue for using the control or the treatment method. This is unlikely, since it is contrary to existing research emphasizing the importance of individual skills. However, in the special case of SE experiments, participants' sample may have been too homogeneous for skill differences to be discernible. This may be possible with student participants that have all had the same training, particularly if participation is voluntary. Voluntary participants are likely to be the most skilled and motivated students [28]. Another issue that could reduce skill differences is that tasks suitable for a laboratory experiment could be too simple for skill to have a major impact on observed performance. However, the possibility of no significant skill dif-

ferences is not supported by the experiments we investigated. Five studies reported the presence of effects due to skill difference among participants although these observations usually related to participant types (e.g., undergraduates, postgraduates, or practitioners) rather than individual participants (see [16], [17], [18], [20], [22]).

2) The treatment method interacts with the skill of the participants. Correlations would be lowered if the alternative method improves the performance of less skilled participants but reduces skilled participants' performance. However, the five studies reporting skill differences mentioned above, all reported that the alternative method increased the performance of more skilled participants with a possibly negative impact on the less skilled participants. Although it appears that interactions are possible, it is not clear how much an effect they would have on the correlations. If highly skilled participants scored well using both the control and alternative method and less skilled participants performed poorly in both conditions, the performance of specific participants should still be relatively consistent, leading to a reasonably large r value.
3) The treatment method interacts with the system being used. The basic crossover design is intended to cater for systematic differences due to using different software application materials when performing SE tasks. The 4-group design is intended to cater for systematic differences due to using a specific set of materials in the first time period. In fact, Section 6 in the Supplementary Materials confirms that the software applications used in each of the studies, with the exception of Study 15 [4] which used materials from the host company, were straightforward IT applications that would be unlikely to exhibit major differences in complexity. It should also be noted that our simulations confirmed large variance instability for small sample sizes. Thus, we would expect to see a fairly high proportion spurious interactions as a result of small sample sizes.
4) The training provided was insufficient for skill differences among participants to affect the outcomes. To fit into time restraints, training available to experiment participants is certain to be limited. It may be that participants were simply not given enough time to practice the new methods before their performance was assessed.
5) Training participants in two different methods could introduce an interaction between method and time period. In medical crossover studies, an interaction between method and time period is a physiological factor caused by two different drugs both being in a patient's body at the same time. Hence, the medical statisticians recommend a *wash-out* period[5] both prior to the experiment, and between the first and second phases of the crossover to minimise any potential interactions between drug and time period. In SE experiments, interactions between method and time period are likely to be a psychological factor, that is, whether learning one method of performing a task helps or hinders learning another method, or whether the teaching process adopted for one method is more effective than the teaching process adopted for the other. Furthermore, if we have taught a method well, we do not expect it to be quickly forgotten, so if learning one method of performing an SE task makes it more difficult (or easier) to learn another method, then the better we train our participants in the method they use first, the more likely we are to introduce a method by time period interaction when they attempt to learn the second method.

Whatever the reason, low values of r cast doubts on the validity of a crossover experiment in SE. Thus, it is important that values of r are reported, and the impact of low values of r is discussed.

Furthermore, it is critical that we investigate causes of low r values, because if inadequate training is a major factor, this affects *all* empirical software engineering experiments, not just crossover experiments. Reverting to between groups designs with strategies such as balancing the skill levels between groups will not make problems associated with training and available practice time disappear. We will just deny ourselves any observable indicators of potential problems. Unless we undertake longitudinal studies that allow us to track improvements in performance over time, we cannot be sure that participants have been given sufficient training and practice time to become competent in a specific technique. In addition, if further studies confirm that the problem is a result of inadequate training and/or practice time, it raises an important ethical issue, because we need to ensure that experiments involving student participants do not adversely affect their educational experience.

6 Conclusions and Recommendations

In summary, r values in SE crossover studies can be quite low. Our data and simulations make it clear that small sample sizes lead to large variations in the observed r-values. However, our results *do not* suggest that sample size is the cause of low r values, because even for larger sample sizes r-values remain low.

In the context of software engineering low r values are difficult to understand. Like most software practitioners and educators, we expect skilled software engineers to outperform less skilled engineers. Most software engineering experiments involve students rather than practitioners, but we have no reason to believe that skill differences are nonexistent among students.

A particular problem is that a low value of r could be due to insufficient training, in one or both techniques being compared, for the effect of the different methods to be properly evaluated. In addition, crossover methods require participants to use both techniques in sequence. However, learning one technique may help or hinder the ability to use another. Any interaction between sequence order and technique would lower values of r.

5. A washout period is time period in which the patients do not use any drug. This means that the effects of any drug they used previously are removed, and the patients return to their baseline condition.

We do not claim that any of these issues actually caused the low values, only that the low values exist and need to be explained before we can be sure that crossover designs are suitable for SE experiments. We recommend that researchers currently analysing crossover design experiments (or, indeed any other repeated measures design) report observed values of r. If the observed estimate is low or negative (i.e., < 0.3) researchers should discuss why this has happened, and the impact of the small value of r on the reliability of their results.

For future studies, researchers in SE need to increase sample sizes. This is a familiar request, but it remains an important issue. Without increased sample sizes we cannot reduce the likelihood that we will observe spurious interactions between technique, participant skill and sequence group that make crossover designs difficult to interpret. Increased sample sizes can be addressed by designing distributed experiments and families of experiments (see, e.g., [29]), but our simulation results suggest that estimates of r estimates and variance estimate do not begin to stabilise until participant numbers reach at least 60. The analysis of the power of two-group crossover designs reported in Section 2.2 suggests that sequence group sizes of approximately 32 participants (for a medium effect size) are equivalent to a between groups study with 64 participants even if $r = 0$. Thus, we assume that two-group crossover designs should aim for a minimum of 30 participants per sequence group. However, without further simulation studies, we cannot be sure of appropriate numbers of participants per sequence group for four-group crossover designs.

In addition, although crossover studies were designed to cater for individual differences, we cannot be confident that crossovers are working as expected unless we collect data about the differences among participants. Such data can be used to investigate, both the validity of crossover design in SE and more detailed hypotheses about the impact of a new SE technique or method.

For studies that investigate difference between competing SE methods (e.g., test-before versus test-after), we strongly advise researchers to give participants time to become familiar with new methods. It would be worthwhile tracking the results of participants over several different practice sessions, which will allow the existence of any individual differences to be identified empirically. Formal hypothesis tests should only be applied once r values obtained from different practice sessions start to stabilise.

For experiments that aim to investigate different working conditions, such as the impact of background noise, or variations in component documentation, the method of performing the software engineering task is the same for all conditions. In such cases, a crossover design with an appropriate sample size is much less risky than a crossover experiment aimed at evaluating competing software engineering technologies. In such cases, the power benefits of replicated experiments is likely to be substantial compared with simple between group experiments, and the risk of significant and genuine interactions complicating analysis and interpretation of results is likely to be substantially reduced.

Finally, we reiterate that if the low r-values are due to insufficient training, this is a problem for all human-participant-based SE experiments that aim to compare different SE techniques or methods, not just crossover experiments.

REFERENCES

[1] S. Vegas, C. Apa, and N. Juristo, "Crossover Designs in Software Engineering Experiments: Benefits and Perils," *IEEE Transactions on Software Engineering*, vol. 42, no. 2, pp. 120–135, 2016.

[2] L. Madeyski and B. Kitchenham, "Effect sizes and their variance for AB/BA crossover design studies," *Empirical Software Engineering*, vol. 23, no. 4, pp. 1982–2017, 2018. [Online]. Available: https://doi.org/10.1007/s10664-017-9574-5

[3] B. Kitchenham, L. Madeyski, and P. Brereton, "Problems with Statistical Practice in Human-Centric Software Engineering Experiments," in *Proceedings of the Evaluation and Assessment on Software Engineering*, ser. EASE '19. New York, NY, USA: ACM, 2019, pp. 134–143. [Online]. Available: http://doi.org/10.1145/3319008.3319009

[4] O. Laitenberger, K. E. Emam, and T. G. Harbich, "An Internally Replicated Quasi-Experimental Comparison of Checklist and Perspective-Based Reading of Code Documents," *IEEE Transactions on Software Engineering*, vol. 27, no. 5, pp. 387–421, 2001.

[5] W. P. Dunlap, J. M. Cortina, J. B. Vaslow, and M. J. Burke, "Meta-analysis of Experiments with Matched Groups or Repeated Measures Designs," *Psychological Methods*, vol. 1, no. 2, pp. 170–177, 1996.

[6] B. Kitchenham, L. Madeyski, and P. Brereton, "Meta-analysis for families of experiments in software engineering: A systematic review and reproducibility and validity assessment," *Empirical Software Engineering*, vol. 25, no. 1, pp. 353–401, 2020. [Online]. Available: https://doi.org/10.1007/s10664-019-09747-0

[7] B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, *Software Cost Estimation with COCOMO II*. Prentice-Hall Inc., 2000.

[8] B. Kitchenham, L. Madeyski, G. Scanniello, and C. Gravino. Supplementary material to the paper "The Importance of the Correlation in Crossover Experiments". Alternative download: https://madeyski.e-informatyka.pl/download/KitchenhamMadeyskiScannielloGravinoTSEappendix.pdf. [Online]. Available: https://doi.org/10.5281/zenodo.4475865

[9] J. Cohen, "A power primer," *Psychological Bulletin*, vol. 46, no. 1, pp. 155–159], 2020.

[10] S. Senn, *Cross-over Trials in Clinical Research*, 2nd ed. Indianapolis, Indiana, USA: John Wiley and Sons, Ltd., 2002.

[11] L. Madeyski, B. Kitchenham, and T. Lewowski, *reproducer: Reproduce Statistical Analyses and Meta-Analyses*, 2020, R package version 0.4.0. [Online]. Available: http://CRAN.R-project.org/package=reproducer

[12] F. Ricca, M. D. Penta, M. Torchiano, P. Tonella, and M. Ceccato, "How developers' experience and ability influence web application comprehension tasks supported by uml stereotypes: A series of four experiments," *IEEE Transactions on Software Engineering*, vol. 36, no. 1, pp. 96—118, 2010.

[13] A. Fernandez, S. Abrahão, and E. Insfran, "Empirical validation of a usability inspection method for model-driven web development," *The Journal of Systems and Software*, vol. 86, no. 5, 2013.

[14] G. Scanniello and U. Erra, "Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments," *Journal of Visual Languages and Computing*, vol. 25, no. 4, pp. 494–517, 2014.

[15] G. Scanniello, A. Marcus, and D. Pascale, "Link analysis algorithms for static concept location: an empirical assessment," *Empirical Software Engineering*, vol. 20, no. 6, pp. 1666–1720, 2015.

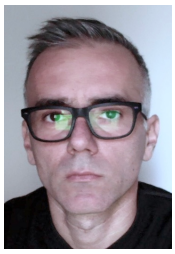
[16] G. Scanniello, C. Gravino, M. Genero, J. A. Cruz-Lemus, and G. Tortora, "On the Impact of UML Analysis Models on Source-code Comprehensibility and Modifiability," *ACM Transactions on Software Engineering and Methodology*, vol. 23, no. 2, pp. 13:1–13:26, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2491912

[17] S. Abrahao, C. Gravino, E. Insfran Pelozo, G. Scanniello, and G. Tortora, "Assessing the effectiveness of sequence diagrams in the comprehension of functional requirements: Results from a family of five experiments," *IEEE Transactions on Software Engineering*, vol. 39, no. 3, pp. 327–342, March 2013.

[18] M. Torchiano, G. Scanniello, F. Ricca, G. Reggio, and M. Leotta, "Do uml object diagrams affect design comprehensibility? results from a family of four controlled experiments." *Journal of Visual Languages and Computing*, vol. 41, pp. 10–21, 2017.
[19] G. Scanniello, M. Staron, H. Burden, and R. Heldal, "On the effect of using SysML requirement diagrams to comprehend requirements: results from two controlled experiments," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE. ACM, 2014.
[20] C. Gravino, G. Scanniello, and G. Tortora, "Source-code comprehension tasks supported by uml design models: Results from a controlled experiment and a differentiated replication," *Journal of Visual Languages and Computing*, vol. 28, pp. 23–38, 2015.
[21] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano, "Assessing the effect of screen mockups on the comprehension of functional requirements," *ACM Transactions on Software Engineering and Methodology*, vol. 24, no. 1, pp. 1:1–1:38, Oct. 2014.
[22] G. Reggio, F. Ricca, G. Scanniello, F. D. Cerbo, and G. Dodero, "On the comprehension of workflows modeled with a precise style: results from a family of controlled experiments." *Software and Systems Modeling*, vol. 14, pp. 1481–1504, 2015.
[23] L. Madeyski, *Test-Driven Development: An Empirical Evaluation of Agile Practice*. (Heidelberg, London, New York): Springer, 2010, foreword by Prof. Claes Wohlin. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04288-1
[24] G. Scanniello, M. Risi, P. Tramontana, and S. Romano, "Fixing faults in c and java source code: Abbreviated vs. full-word identifier names," *ACM Trans. Softw. Eng. Methodol.*, vol. 26, no. 2, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3104029
[25] S. Romano, G. Scanniello, D. Fucci, N. Juristo, and B. Turhan, "The effect of noise on software engineers' performance," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM'18, 2018.
[26] L. Madeyski and B. Kitchenham, "Would wider adoption of reproducible research be beneficial for empirical software engineering research?" *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 2, pp. 1509–1521, 2017.
[27] A. Santos, O. Gomez, and N. Juristo, "Analyzing families of experiments in se: A systematic mapping study," *IEEE Transactions on Software Engineering*, vol. 46, no. 5, 2020.
[28] R. L. Rosnow and R. Rosenthal, *People Studying People Artifacts and Ethics in Behavioural Research*. New York: W.H. Freeman & Co., 1997.
[29] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs, and A. Pohthong, "Robust Statistical Methods for Empirical Software Engineering," *Empirical Software Engineering*, vol. 22, no. 2, pp. 579–630, 2017.

Barbara Kitchenham is an Emeritus Professor in the School of Computing and Mathematics at Keele University in the UK. She has worked in software engineering for over 40 years both in industry and academia. She has published over 150 software engineering journal and conference papers. Her most recent research has focused on the application of evidence-based practice to software engineering. In 2019, she was awarded the IEEE Technical Committee Distinguished Women in Science & Engineering (WISE) Leadership Award.

Lech Madeyski is an Associate Professor and Deputy Head for Research at Department of Applied Informatics, Wroclaw University of Science and Technology, Poland. He has been a Visiting Researcher at Keele University, Brunel University London, and a Visiting Professor at Blekinge Institute of Technology. His research focus is on empirical (evidence-based) software engineering, data science in software engineering, robust statistical methods, reproducible research, software quality, mutation testing, agile methods. He is a co-founder of *e-Informatica Software Engineering Journal*, published, e.g., in *IEEE Transactions on Software Engineering*, *Empirical Software Engineering*, *Information and Software Technology*, and authored "Test-Driven Development: An Empirical Evaluation of Agile Practice" book by Springer. He serves as steering committee member, program co-chair, workshops/special sessions/track co-chair, and PC member of international conferences in software engineering.



Giuseppe Scanniello received his Laurea and Ph.D. degrees, both in Computer Science, from the University of Salerno, Italy, in 2001 and 2003, respectively. In 2006, he joined, as an Assistant Professor, the Department of Mathematics and Computer Science at the University of Basilicata, Potenza, Italy. In 2015, he became an Associate Professor at the same university. He has published more than 170 referred papers in journals, books, and conference proceedings. He serves on the organizing of major international conferences and workshops in the field of software engineering. Giuseppe Scanniello leads both the group and the laboratory of software engineering at the University of Basilicata (BASELab).

Carmine Gravino is an Associate Professor at the Department of Computer Science of University of Salerno. He is the co-director of the Software Quality and Measurement (SQM)/Web Engineering Laboratory and his research interests include software project management, software measurement and functional size measurement methods, predictive modeling for software engineering, software maintenance and evolution, software technology evaluation through experimental means. He has published more than 100 papers in international journals, books, and conference proceedings. He has served as organizing and program committee member of several international conferences in the field of software engineering and is in the editorial boards of international journals. He has also served as reviewer of several software engineering journals.