Lessons and Recommendations from a Research Audit for the Transparent Psi Project (TPP)

James E. Kennedy
Version of April 19, 2023

The lessons and recommendations described here were developed from a research audit (Kennedy, 2023a) of the Transparent Psi Project (Kekecs et al., 2023). The TPP was intended as a potential model and learning experience for optimal research practices. The TPP attempted to confirm one of the studies Daryl Bem reported in 2011 as claiming evidence for precognition or ESP. The TPP outcome was strong evidence that precognition did not occur in the TPP. To put the audit in context, some key points from the introduction in the audit report (Kennedy, 2023a) are quoted below:

> The purpose of this audit is to apply to the TPP my experience with research audits in regulated clinical research. I initiated the audit after reviewing the final reports of previous audits of TPP by an IT auditor and two research auditors, all with academic backgrounds. … The previous audits did not have the range of topics, details, emphasis on protocol deviations, and edginess consistent with my expectations for an audit. My hope is to provide an audit that demonstrates useful expectations based on experience outside of academic research.
>
> …
>
> The goals of a good academic research audit have two components: (a) verify that recognized good research practices were used, and (b) verify that the actions and practices specified in the preregistration and/or protocol were properly implemented. The topics described in this audit reflect my views about good research practices.
>
> An audit in regulated clinical research has another major component that unfortunately cannot be expected for academic research. Regulatory guidelines for clinical trials define an audit as:
>
> > A systematic and independent examination of trial-related activities and documents to determine whether the evaluated trial-related activities were conducted, and the data were recorded, analyzed, and accurately reported according to the protocol, sponsor's standard operating procedures (SOPs), good clinical practice (GCP), and the applicable regulatory requirement(s). (US FDA, 2018, page 3)
>
> Any organization that has a role in clinical data collection, data processing, or data analysis is expected to have written SOPs that implement good research practices (US FDA, 2007, 2018). SOPs cover everything that involves the data … Such SOPs are typically far outside the thought process and practices of academic

> researchers—but would be very valuable for implementing consistently good research practices. …
>
> Good research practices should make it very difficult or impossible for one person acting alone to unintentionally or intentionally (fraudulently) bias the data with little chance of detection. …
>
> When a regulatory auditor asks the carefully worded question "how do you know that the programmer did not change the data?" it does not matter whether a change was intentional or unintentional. Documented evidence that the data were not changed is expected.

The topics discussed here are among the potential lessons from the TPP. Some key findings from the audit are noted here, but the detailed findings and discussion are not repeated.

References

Kekecs et al. (2023). Raising the value of research studies in psychological science by increasing the credibility of research reports: The Transparent Psi Project. *Royal Society Open Science*, 10: 191375. https://doi.org/10.1098/rsos.191375

Kennedy, J. E. (2023a). Research Audit for the Transparent Psi Project (TPP). PsyArXiv. https://doi.org/10.31234/osf.io/mytgd. Also at https://jeksite.org/psi/tpp_audit2_jk.pdf

US FDA (2007). Guidance for Industry: *Computerized Systems Used in Clinical Investigations*. https://www.fda.gov/media/70970/download

US FDA (2018). Guidance for Industry: *E6(R2) Good Clinical Practice: Integrated Addendum to ICH E6(R1)*. https://www.fda.gov/media/93884/download

Lessons and Recommendations

1. Validation of data collection software is essential.

The initial data collection software for the TPP had a significant programming error that, among other things, would allow a computer-savvy subject to see the randomly selected target before making a response. This error was caught in the initial validation of the data collection software. A programming error like this is not unusual and does not indicate incompetence by the programmer. Such programming errors can be expected, particularly when software specifications are developed by nontechnical people such as psychologists.

As another example, a subtle data collection software programming error that produced false evidence for ESP was described by Watt and Brady (2002). The error was found in checking the software after the study was completed rather than before the study started. It seems likely that subtle errors like this will usually not be detected in most psychological research.

Formal software validation and user acceptance testing are well established necessary steps in software development. Formal validation of data collection software should be included in the timelines and budget for research with automated data collection processes.

References

Watt, C., & Brady, C. (2002). Experimenter effects and the remote facilitation of attention focusing: Two studies and the discovery of an artifact. *Journal of Parapsychology*, 66, 49–71.

2. Evaluation of power or operating characteristics (inferential errors) is essential, including for sequential Bayesian analyses.

The initial simulations of Bayesian analyses for TPP found unexpected and unwanted properties for the planned analysis. Simulations had a crucial role in designing a study with >.95 probability of detecting a null effect if the effect is actually null and >.95 probability of detecting a precognitive effect if precognition is true. The strong inferences from the study outcome were a result of the high power.

Simulations to generate the operating characteristics or power curve evaluate the performance (validity and reliability) of a planned statistical analysis, similar to evaluating the validity and reliability of a measurement instrument (Kennedy, 2023b).

Two different methods were used for the simulations of Bayesian analyses. One method samples from models for an assumed effect size that covers the range of possible effect sizes (US FDA, 2010). The other method samples from a prior probability distribution. The first method provides more useful information about the effect sizes that can be reliably investigated and is more objective because it is based on actual possible states of the world, not the researchers' ideas or assumptions about prior probability.

References

Kennedy, J.E. (2023b). Planning falsifiable confirmatory research. PsyArXiv. https://doi.org/10.31234/osf.io/pu2xy. Also at https://jeksite.org/psi/falsifiable_research.pdf

US FDA (2010). Guidance for Industry: *Guidance on the Use of Bayesian Statistics in Medical Device Clinical Trials.* https://www.fda.gov/media/71512/download

3. An online repository copy of data is very valuable, but easy-to-use, reliable, secure processes remain to be developed.

Generating two copies of the data during data collection or as soon after as possible is a basic step for secure research practices. The key principle is that one person acting alone should not be able to access and modify both copies of the data. This means that reliable, unchangeable version controls and audit trails should be implemented in a data repository, and/or different people should control each copy of the data.

Git-based websites were primarily intended for software development, not as data repositories. As discussed below for software monitoring, Git is very complex and appears not to have optimal tracking of file changes for preventing fraud. If Git-based websites are used as a secure data repository, the optimal practice would be to have Git experts with extensive knowledge of the options and inner workings of Git develop guidelines for using Git to prevent fraud.

Having a data collection system automatically email a copy of the data to a distant third party is an easy, reasonably reliable strategy for generating a second copy of the data that will prevent an experimenter from being able to make subsequent undetectable data alterations.

4. A good research audit is much more extensive than current peer review for publication, including Registered Reports.

Most academic researchers, particularly psychologists, have experience with peer review for publication, but do not have experience with a good, detailed research audit. In this situation, it can be expected that a researcher asked to audit a study will basically do something close to peer review for publication. The research audit reports for TPP by the two auditors with academic backgrounds were 3 and 4 pages. The audit report adapted from my experience with regulatory audits in clinical research is 34 pages (which includes significant added explanations).

An audit is a much more in-depth investigation of the details of the study conduct. The focus is on evidence and documentation about the actual conduct of the study, not just a review of what was planned or what an experimenter says happened.

The experience with TPP shows how this plays out. The TPP was a Registered Report that included extensive stage 1 review (study plans before data collection) and stage 2 review (the report after data collection) by four reviewers. The reviews and responses were handled as open science and are at https://royalsocietypublishing.org/action/downloadSupplement?doi=10.1098/rsos.191375&file=rsos191375_review_history.pdf.

One of the reviewers asked about the possibility that the software for data collection could be fraudulently altered. The experimenters responded that:

> The code cannot be modified without it being detectable. The server contents are synced with a GitLab repository, this way the code running on the server is verifiable at all times. Any modification to the code would be visible. (Appendix C, Reviewer 1, Response 19)

The reviewer accepted this response and the use of the synced GitLab repository was included in the preregistration and in the stage 2 paper.

As part of this audit of TPP, the PI was asked about the fact that the IT auditor mentioned in his final report that a file had been changed on the data collection server without being tracked in the Git repository. The report had downplayed this case and it was easy to miss. Upon inquiry with the programmer, the PI discovered that there had

been miscommunication and the GitLab repository had not been synced with the server as had been planned, preregistered, and reported. This was a significant protocol deviation.

When combined with a failure to log user accesses to the server (another significant protocol deviation) and improper system access granted to the programmers, the net result was seriously deficient management of the security of the data collection server. As discussed in the audit, if the study would have obtained evidence for ESP, these deficiencies would have been a significant concern.

This case demonstrates that even extensive peer review for a Registered Report cannot be expected to identify significant methodological deficiencies in research conduct. A detailed research audit is needed.

Important questions include: (a) how to find or develop the expertise needed for a good detailed research audit, (b) whether the substantial effort needed for a detailed audit can be expected to be voluntary or whether compensation is appropriate, and (c) how to establish independence if an auditor is paid by those being audited?

5. Quality control (QC) comes before auditing.

Quality control (QC) practices are active efforts to establish the integrity of research processes, particularly data collection, data management, data analyses, and data reporting. QC practices are based on double-checking and monitoring of all key research activities. Examples of common QC practices include formal software validation, double-checking or double-entry of data, keeping electronic and/or paper records of key research processes with date-time stamped audit trails, verifying that these records are being properly kept, and frequent checking of unexpected event logs for a server and software system.

QC practices are done as the study is being conducted by people who have responsibility for maintaining and documenting the integrity of the research. In regulated clinical trials, this includes “monitors” who periodically “visit” (inspect) the sites collecting data to identify and correct any research activities that are not being properly conducted and documented (US FDA, 2018, pages 33-37). The monitors prepare reports about these site visits.

An audit is typically done when a research project is complete to evaluate the integrity of the final result. An auditor is an independent person who verifies that research activities were adequately conducted and documented, but does not have responsibility for implementing or overseeing the QC measures during the research. An audit is often too late to correct problems that are found.

In the TPP, the “IT auditor” was actually responsible for quality control more than auditing. The IT auditor was responsible for conducting pre-study software validation and “will oversee data integrity throughout the study.” Deficiencies in managing the data collection server to a large extent reflect the IT auditor not fulfilling the QC role. However,

the duties were and will be ambiguous if the different expectations and associated titles for QC and auditing are not clearly distinguished.

Psychological research needs QC people more than auditors. QC people have significant responsibility for double-checking work and establishing the integrity of the research. The primary value of an audit is to evaluate the adequacy of quality control. Quality control practices detect and correct unintentional errors and oversights, and also possible intentional errors or fraud.

References

US FDA (2018). Guidance for Industry: *E6(R2) Good Clinical Practice: Integrated Addendum to ICH E6(R1)*. https://www.fda.gov/media/93884/download

6. Good research practice guidelines and/or standard operating procedures (SOPs) for general use would be very valuable for confirmatory psychological research.

Psychologists increasingly utilize technology without technical training or experience in understanding the implications of the details of the technology. The vulnerabilities and pitfalls in managing the technology are often not recognized.

Practical, usable guidelines for all aspects of research would be valuable, including (a) setting up workstations and servers with limited access permissions for secure research, and (b) software validation and documentation. The guidelines would be useful for both psychologists and those providing technical support for psychological research.

A website could be established for model methodological guidelines and SOPs for psychological research, including for data collection and management for different types of studies. Guidelines and SOPs for regulated clinical trials (US FDA, 2007, 2018, 2023) may be a useful starting point and a step beyond the usual limited technical experience of those working in academic psychology.

The Society for Clinical Data Management (https://scdm.org/) may be a useful model of an organization that could be established to promote good research practices in psychology.

References

US FDA (2007). Guidance for Industry: *Computerized Systems Used in Clinical Investigations*. https://www.fda.gov/media/70970/download

US FDA (2018). Guidance for Industry: *E6(R2) Good Clinical Practice: Integrated Addendum to ICH E6(R1)*. https://www.fda.gov/media/93884/download

US FDA (2023). Draft Guidance for Industry: *Electronic Systems, Electronic Records, and Electronic Signatures in Clinical Investigations: Questions and Answers.* https://www.fda.gov/media/166215/download (Use the final version when it become available.)

7. It is time to implement routine measures to prevent experimenter fraud.

The problem of experimenter fraud has been clearly established (Strobe, Postmes, & Spears, 2012). Strobe et al. and other investigators have also established that replication and peer review are generally not effective at detecting or deterring experimenter fraud,

As noted in the TPP audit, fraudulent data collection programming to produce a false ESP effect would be easy and tempting in an automated precognition study like this, with very little chance of detection if the programming is not monitored. The fraud would be impossible to detect from patterns or artifacts in the data.

Without transmitting a copy of the data to a third party or to a controlled repository, the only copy of the data would be on a workstation or server, often in a university setting without good control of restricted access. This adds many more possibilities for compromise. Minimally sophisticated fraud that includes changing the date-time for the last modification of a file would not be needed because no one would be paying attention to those details.

Ignoring well-established methodological issues such as experimenter fraud and software validation appears to be an accepted part of the current research culture in psychology.

Most confirmatory research can and should be designed to make fraud by one person acting alone difficult, not easy and tempting with little chance of getting caught. The research culture should expect duplicate copies of the data and routine quality control practices such as software validation and double-checking and monitoring data collection and data management.

References

Strobe, W., Postmes, T., & Spears, R. (2012). Scientific misconduct and the myth of self-correction in science. *Perspectives on Psychological Science*, 7, 670–688. https://journals.sagepub.com/doi/pdf/10.1177/1745691612460687

8. Git appears to be not optimal for software tracking to prevent programming fraud.

Git is intended to facilitate software development and appears to be designed to give the programmers using Git control over the tracking of file changes—which is not an optimal practice for preventing fraud. Git does not monitor folders and automatically track file changes. Such automatic tracking would be counterproductive for the frequent file changes when creating and editing programming code. Git is based on a programmer entering commands when the programmer wants Git to recognize and track a file change. Files can be directly changed outside Git processes and the changes will not be identified by the Git system until a Git command such as Git Status is run, as occurred with the TPP.

Git is very complicated and includes several options for temporarily or permanently excluding a file from Git tracking. Updating a repository from another repository is a particularly complicated process. Extensive expertise in the many options, interactions of the options, and inner workings of Git would be required to confidently use Git to prevent programming fraud—or to know whether Git can be used reliably for that purpose. The possibilities for circumventing the tracking of file changes would need to be identified and neutralized.

Development of a detailed guidance document by Git experts would be useful if Git is used to prevent programming fraud. Other less complicated, more focused methods for tracking software changes could be used with more confidence and less overhead.

9. Simple dedicated practices that are independent of the complex details of the operating system and Git can provide greater confidence for monitoring software changes.

Key conceptual points for optimal tracking of software changes are:

- The software tracking system should be set up and controlled by an independent quality control (QC) person, not one of the programmers being monitored. The QC person should have read-only access to the application server. Giving the QC person full access to the server adds to the vulnerability of the server, rather than adding to security.
- The software for identifying changes to the application programming code should be run from a different computer that is not accessible to the application programmers being monitored. Relying on tracking processes run from the application folder on the application server will be much more vulnerable to compromise.
- The tracking system should be sufficiently simple that a nontechnical person can understand it and monitor it. Complex systems that require extensive specialized technical expertise will take much more effort and often be less reliable (but will sometimes be recommended because it provides income and job security for technical persons).
- The programmers being monitored should not have access permissions to the server that would allow them to alter operating system functions such as tracking accesses to the server or altering date-time records for file changes and audit trails.

A simple process would be to assign a QC person monitoring software integrity read-only access to the application files on the server. The QC person could be a computer professional, a PI, or an independent third party. As a last step in pre-study software validation, a copy of the application files and folders would be copied to a computer under the control of the QC person (and not accessible to the application programmers). The QC person would have a program that compares this original copy of the files with the files currently on the application server. Certain files could be excluded. At a

minimum the program could be scheduled to run several times a day or even every 15 or 30 minutes to verify that the file names, sizes, and modification date-time match. Optimally, about once every day or so, the comparison would compare every byte of every file. Logs would be produced and an email sent to notify the QC person if a serious unexpected discrepancy was found. Ideally, an open-source dedicated program would be developed that could be easily and widely used for this purpose, including copying the files and folders. This essentially would do what is wanted from Git to prevent fraud, but without the complexity and resulting uncertainty and overhead of Git.

Once the program was developed, the monitoring to detect changes could be done by a person with limited technical knowledge of programming and operating systems.

The main obstacle to this process is that most programmers and IT professionals, particularly in academic settings, do not have experience working with high security like this. When a person with a nontechnical background, such as a psychologist, talks to them about security, the programmers and IT people will propose what they know about, which is complex processes like the operating system and Git. The complications and vulnerabilities that were well demonstrated with TPP do not come to mind. A nontechnical person will defer to them and learn the hard way that the types of weaknesses that occurred in TPP are common in such situations. The underlying problem is the learning curve in developing more secure programming processes. Software tracking for research is relatively simple with little overhead once a person recognizes the need to look outside the complex systems. Guidelines and SOPs as noted above would be very valuable.

Copyright Notice