Further Details on the Experimental Design

J. Olsson et al.

1 Introduction

This manuscript details aspects of the experiment carried out by Olsson et al. (2020). Its purpose is to document instance-specific design decisions that had to be cut from the journal article due to size constraints.

2 Participants

Forty software practitioners participated in this study. They attended one 90-minute session each and did so between March 26 to April 17 (2019). The session was held in a conference room at their work office (four cities in Sweden).

The participants were employed by eleven companies and one government agency, which covered a diverse set of company sizes and business domains (e.g., automotive, finance, and renewable energy).

The participants were selected by the companies, which in turn had been selected through convenience sampling by using the researchers' and the university's professional networks. The companies were then asked to select participants at their own volition, as one of several actions to reduce the risk of discriminating roles or company types. Consequently, the companies contributed with as many participants as they wanted (range 1–7, $\mu = 3.33$, $\sigma = 1.87$). Similarly, the term *software practitioner* (as opposed to, e.g., *developer*) was used to describe eligible participants.

2.1 Participant Motivation

Voluntary participation was motivated by informed consent and anonymity, i.e., neither participants nor companies were offered monetary or similar benefits. In return, we made sure to limit any economic harm, e.g., by terminating sessions

that would continue past the allocated time. Hence, participant commitment was comparatively minor: A maximum of 90 minutes of their regular work time, approved by their company. Further, the participants were informed about their rights to decline to answer any questions or, at any time, withdraw from the study.

Informed consent was achieved by the participants reading and signing a confidentiality agreement (also available in this replication package) at the start of the session. It disclosed the study's purpose, promised anonymity, and declared that the participant's data would be destroyed upon their request.

Confidentiality was ensured through randomly generated identifiers. The identifier was linked to the participant through a single (physical) document, which would be used if required during the analyses, e.g., to explain potential outliers. The links were destroyed upon completion of the analysis phase.

In conclusion, the participants can be presumed to have taken part in this study willingly. They faced no coercion, and their employer authorized participation.

2.2 Allocation to Treatments

The repeated-measures experiment consisted of five scenarios (ScA–ScE) with two levels each (L and H). While the order of the interventions ScA–ScE was randomized in order to cancel out any learning effect, a genuinely random assignment (scenario permutations and random level) would result in an infeasible number of combinations. Hence, the levels were fixed to two *treatment patterns*: LHHLH and HLLHL. In other words, the n^{th} participant was randomly allocated to a permutation of the scenarios but would receive the treatment pattern LHHLH if n was odd and HLLHL if n was even.

Those particular (complimentary) treatment patterns were constructed to mitigate the risk that they influence responses. For instance, some patterns would obfuscate whether the explanatory variable caused response variable changes, e.g., LLHHH might confound the participant becoming bored or comfortable with the experiment. Similarly, some patterns could potentially allow participants to influence the data, e.g., the participant might recognize that LHLHL alternates the levels.

3 Setting

Each session included a singular participant. The researchers arrived at the room 30 minutes before the session to ensure ample time for preparations. Further, this time would act as a buffer if multiple participants from the same

company were scheduled in sequence and, e.g., arrived late.

4 Anchor Point

At the start of the measurement sitting, the participant was presented with a practice scenario (anchor point). The intention was to accustom the participant to the experiment, thereby lowering the impact of learning effects.

From the viewpoint of the participant, the anchor point should appear no different from the scenarios. Their SAM ratings for the anchor point were, however, not included in the analysis. Further, to act as a common baseline for all participants, the anchor point had just one treatment level (H).

5 Deacclimatization Periods

Evidence suggests that affective states may persist after stimulus removal (Gomez et al., 2009). Hence, the repeated-measures could be distorted if the participant carried over affects from one scenario to the next. To help the participant return to their natural affective state, short pauses (deacclimatization periods) were introduced between each scenario.

Surprisingly, a search of the literature revealed few studies discussing the topic of deacclimatization periods—none of those offered or referenced concrete guidelines, e.g., duration or appropriate activities. For instance, Cinaz et al. (2013) used relaxing documentary films but did insufficiently report on the procedure and the material to allow reproduction.

In the end, we employed 120 seconds of rest and no specific activity (silence or small talk).

6 Scenarios

As DTD is challenging to measure directly, software scenarios were constructed to act as proxies during the experiment. Special care was taken to create appropriate scenarios, here called *concrete representations* of DTD. First, to make the findings more relevant to industry, the scenarios should exemplify a problem that could realistically be encountered in practice. Second, because smells are not necessarily indicative of definite quality problems (Sharma and Spinellis, 2018), the design smells should be actual (rather than contingent) liabilities. Third,

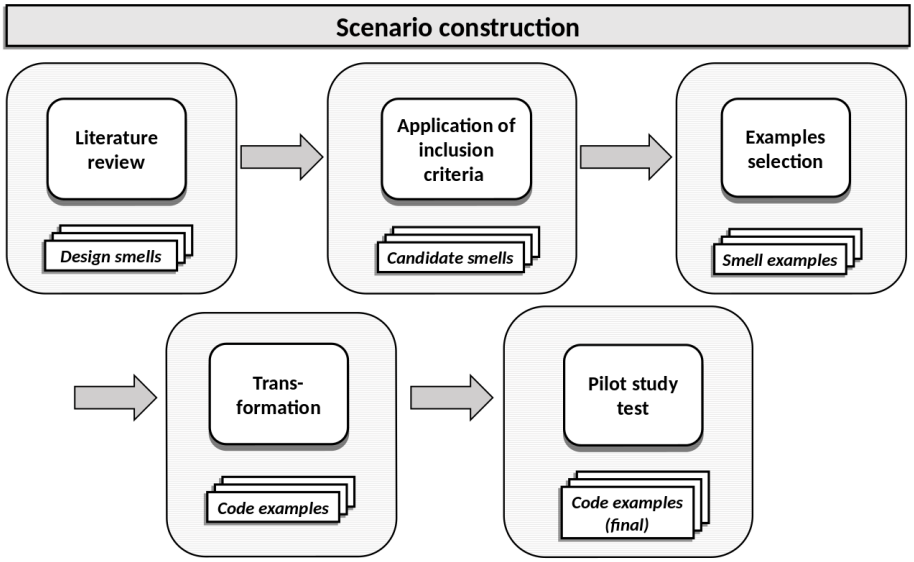

Figure 1: The process for constructing the scenarios.

because the proxies should be self-contained, it must be possible to present the TD item (and its resolution) in a reasonably sized piece of source code.

In other words, the advantages of the refactoring of these *concrete representations* of DTD should be pretty much evident from the examples alone (keep in mind, however, that the participants cannot compare the two versions). Unfortunately, concrete representations differ from the abstract representation of smells typically used in smell catalogs, see, e.g., (Garcia et al., 2009).

After reviewing the literature, the scenarios were constructed (see Figure 1) based on the work of Ganesh et al. (2013), and refined by Suryanarayana et al. (2014). Out of the 25 design smells, we selected (candidate smells) those that fulfilled the following inclusion criteria:

1) The smell had a *concrete representation* and;

2) The smell was listed as negatively impacting understandability and;

3) The essence of the smell could be understood in a short amount of time.

In instances where Suryanarayana et al. (2014) listed several examples (smell examples) of the candidate smell, we chose the one deemed most appropriate for the experiment's context.

Next, the smell examples were transformed into code examples. They were

harmonized into correct syntactic syntax highlighted, Java source code that conformed to a popular coding style guide. Some examples were deemed too domain-specific without explanatory texts. Those had their context and names modified.

Finally, these code examples were tested in pilot studies, and the six (anchor point plus five measurements) most suitable scenarios were used for the measurement sitting.

References

Cinaz B, Arnrich B, Marca R, Tröster G (2013) Monitoring of mental workload levels during an everyday life office-work scenario. Personal and ubiquitous computing 17(2):229–239

Ganesh S, Sharma T, Suryanarayana G (2013) Towards a principle-based classification of structural design smells. J Object Technol 12(2):1–1

Garcia J, Popescu D, Edwards G, Medvidovic N (2009) Toward a catalogue of architectural bad smells. In: International Conference on the Quality of Software Architectures, Springer, pp 146–162

Gomez P, Zimmermann PG, Guttormsen Schär S, Danuser B (2009) Valence lasts longer than arousal: Persistence of induced moods as assessed by psychophysiological measures. Journal of Psychophysiology 23(1):7–17

Olsson J, Risfelt E, Besker T, Martini A, Torkar R (2020) Measuring affective states from technical debt: A psychoempirical software engineering experiment. 2009.10660

Sharma T, Spinellis D (2018) A survey on software smells. Journal of Systems and Software 138:158–173

Suryanarayana G, Samarthyam G, Sharma T (2014) Refactoring for Software Design Smells: Managing Technical Debt. Morgan Kaufmann