SN Computer Science (2024) 5:1097
https://doi.org/10.1007/s42979-024-03471-3

ORIGINAL RESEARCH

# Metrics for Evaluating and Improving Transparency in Software Engineering: An Empirical Study and Improvement Model

Paulinus Ofem[1] · Bassey Isong[1] · Francis Lugayizi[1]

Received: 19 April 2024 / Accepted: 27 October 2024



## Abstract

Transparency remains an emerging non-functional requirement that stakeholders in the software engineering process (SEP) must consider. Existing studies provide evidence of the usefulness of transparency and offer hindsight on how transparency may be evaluated in different contexts. Considering communication, software maintainability, and stakeholders' productivity, SEP has not thoroughly explored the benefits of transparency. Currently, there is no appropriate measurement of transparency in SEP, and the consequence of this is that theoretically and empirically, the factors that characterise transparency and their interrelationships are unknown. This study empirically validates measures and metrics for evaluating and improving software-development-life-cycle-product transparency. It investigates the correlations between transparency factors and their corresponding metrics. Following the Goal Question Metric paradigm, it proposes a transparency evaluation and improvement model based on the transparency factors, measures, and metrics. The usefulness of the metrics was demonstrated via a controlled experiment to investigate the impact of transparency on the maintainability of software requirements specifications (SRS) and its support for good communication and stakeholders' productivity. Results suggest that object-oriented SRS are easier to maintain and offer better communication and more productivity. The practical implication of our findings is that adopting use case modelling in developing SRS benefits from transparency implicitly. Secondly, a transparency evaluation team can use the improvement model to evaluate SRS transparency explicitly and and improve SRS based on the evaluation outcome. Our proposals can be applied early in the requirements engineering and design phases of SEP. The model can be used to diagnose and deal with maintainability and communication problems with SRSs and design documents.


**Keywords** Transparency · Requirements engineering · Software engineering · Metrics

## Introduction

Transparency remains an emerging non-functional quality factor in computing. It has been conceptualised and applied within information systems (IS) and software engineering (SE) sub-disciplines. The notion of transparency is typically interpreted as the act of "being open." In the context of IS, transparency is defined in terms of business process and information disclosure [1–3]. In SE, transparency refers to how software stakeholders can address their concerns with the software [4, 5]. It is also considered as the full disclosure of the functions of a software system [6]. Therefore, transparency is considered a non-functional requirement (NFR) of computing software stakeholders and business processes. Because transparency of processes is also vital, in [7], we defined transparency as "the degree to which information about software products and processes is open to the stakeholders of a software project."

The concern for stakeholders' transparency has become more evident globally. This is due to incidents like the VolksWagen's carbon emission scandal [8] and Boeing's 737-Max plane crashes [9]. Leite and Cappelli [1] categorised these concerns into three levels: a) citizens' general concern for information and process transparency, b) users' demand for transparency in terms of the mediator (product or software) of information and process transparency, and

✉ Paulinus Ofem
25831097@student.g.nwu.ac.za

Bassey Isong
bassey.isong@nwu.ac.za

Francis Lugayizi
francis.lugayizi@nwu.ac.za

1 School of Computer Science and Information Systems, North-West University, Building A12, Mahikeng Campus, PO Box X2046, Mmabatho 2735, South Africa

Published online: 27 November 2024

c) stakeholders' demand for transparency of software and the software process. Therefore, stakeholders, especially requirements engineers and developers, need to know how to deal with transparency requirements during the software engineering process (SEP). This is even more so as transparency implementation is considered a requirements engineering activity that should be dealt with in the early phase of the SEP. The current study focuses on the transparency of software life development cycle (SDLC) products.

Transparency research has been predominantly witnessed in IS domain, where several conceptualisations and applications have been reported. The IS domain conceptualised transparency in the context of functional software, data processed, and the processes supported. This context is a process transparency-driven context (PT) because SEP is not considered. The SE discipline has witnessed minimal research in transparency. Existing studies [4, 5] in SE have considered transparency in the context of SEP. We consider this to be an information transparency (IT) context since the SDLC encompasses various phases and their corresponding artefacts that capture the functional requirements of a software system. Despite the remarkable achievement in transparency research, in SE, there appears to be no measurement of transparency nor a framework for evaluating and improving transparency in this context. This is evidenced by the results in [10]. Research in this regard has been called for by the authors in [4, 5, 11]. In this vein, the current study aims to fill this significant gap by empirically validating transparency measures and metrics we have previously proposed.

Measurement is an everyday life activity vital to scientific and engineering disciplines. Establishing measures facilitates the acquisition of information that can be applied to developing theories and models [12]. To this end, several measurements or metric models have been proposed in the literature. Existing models do not always cater to varying measurement needs and contexts. This necessitates developing targeted measurement models that fulfil an organisation's interest [12, 13]. To define a suitable transparency measurement model, we combined and adopted Gilb and Finzi's "define-your-own model" [13] approach and the Goal Question Metric (GQM) methodology [14] to develop transparency measures and metrics. The paper that reports the complete definition of the metrics and their role in modelling transparency has been submitted, passed preliminary review, and is now being peer-reviewed by the International Conference on Software Technology and Engineering (ICSTE 2024). The conference was held in August 2024.

Therefore, the overarching goal of this study is to empirically investigate whether the proposed transparency factors and metrics assess what they purport to assess by testing whether transparency contributes to SRS' maintainability, effective and efficient communication, and stakeholders' productivity. Utilising the transparency factors and metrics, it proposes a transparency-improving model.

## Related Work

Transparency is a multifaceted concept that remains theoretically and empirically under-researched. This observation is supported by the findings from a systematic literature review (SLR) we performed in the early phase of our transparency research programme. The full report of the SLR outcome is provided in [10]. A similar observation is noted in [15]. Consequently, transparency has served as a notion for describing heuristics (i.e., mental shortcuts or processes) and anomalies, therefore lacking in theoretical and empirical underpinnings. This has also resulted in a poor understanding of transparency by stakeholders. Stakeholders generally do not know how to deal with transparency in their various domains, especially in SEP. It has also resulted in limited research and scarce publications. We summarise some of the SLR findings in this section as follows.

Transparency has been majorly conceptualised in the information systems (IS) domain. These conceptualisations [1–3, 16–23] are from the view of deployed software, the data it deals with, and the supported business processes. For instance, the authors in [1, 16, 17] conceptualised transparency from an information and business process perspective. Leite and Cappelli [17] proposed a transparency ladder with attributes that help transparency. We observed that the ladder indicates a hierarchy of transparency achievement and the attributes of transparency that can be achieved. However, it does not provide a measurement nor a programme for achieving transparency.

Furthermore, SE studies [4, 5, 11, 24] did not provide a well-defined measurement for transparency. However, they offer good hindsight and background that supports further research on how SEP can achieve transparency. The studies have not equally provided transparency requirements that developers and end-users may use to assess the degree of transparency achievement. By adopting a formal approach, the outcome of our study can shed more light on transparency measurement.

However, we did not encounter any empirical investigation of the correlations between transparency-describing factors. To address these gaps, we proposed a conceptual framework for understanding, evaluating, and improving transparency within SEP in [25]. The framework consists of five components, namely, (1) stakeholders, (2) stakeholders' transparency requirements, (3) transparency factors, (5) transparency metrics, and (6) improvement scheme. We must identify stakeholders with transparency concerns to initiate a transparency evaluation and improvement exercise. We next establish these transparency concerns as requirements. These requirements should be achieved via well-defined factors and metrics that measure the fulfilment of each requirement. Based on the factors and metrics, a transparency improvement scheme can be designed to improve transparency. Transparency factors have previously been identified and populated in [25].

Because stakeholders and transparency requirements are at the heart of our transparency improvement framework, in [7], we populated the stakeholders' transparency component. We elicited an initial set of transparency requirements that may be evaluated and improved during SEP. This study argues that the conceptualisation of transparency, especially in SEP, should investigate the interrelationships between the factors that explain transparency. This is a requirement for theory formulation and scientific explanation of concepts [15, 26, 27]. The interrelationships we observed in the literature are products of decompositions and aggregations based on expert judgement and goal modelling. Tu [11] and Tu et al. [5] called for fulfilling this theorisation requirement in SEP. This is in addition to augmenting transparency factors together with their operationalisations. This approach also considers the peculiarities of the context in which it is applied. This paper aims to fulfil the requirement for theory building and provide an approach to improving transparency in SEP.

According to [1], transparency is a quality that is orthogonal (i.e., not required to function) to the functionality of a software system. The implication is that software systems do not require the fulfilment of transparency to function. The authors also opine that transparency may not be approached like a functional characteristic of software. Again, the consequence is that transparency may defy measurement in the typical or traditional sense of measurement. This implies that achieving transparency may also depend on less formal measurement approaches. Despite these submissions, we think that in SEP context, it is possible to achieve objective measurements. This outcome will enable informed decisions on software artefact transparency and the choice of development paradigms that give birth to them.

## Methodology

This section presents the research motivation and objectives the study intends to address, the context of the study, and the research approach we adopted.

### Motivation

Transparency remains an emerging concept researchers do not often study from a theoretical standpoint [10, 15]. Despite the achievements recorded in transparency research, there is no framework or measurement of transparency in SEP. The need for transparency measurement and further research in this context has been explicitly stated in [5, 11]. The current research is motivated by the need to measure transparency in SEP. It also aims to provide scientific and practical evidence supporting the usefulness of transparency. These outcomes make transparency a viable additional non-functional quality requirement in SEP. This evidence may equally support the application of transparency to achieve effective communication, software maintainability and stakeholder productivity in SEP.

### Research Objectives

In this study, we focus on the problem of realising transparency in software artefacts such as SRS. The research objectives (RO) are as follows:

- RO1 —To evaluate the impact of transparency on the maintainability of SRSs.
- RO2 —To evaluate the impact of transparency on effective and efficient communication of SRSs.
- RO3 —To evaluate the impact of transparency on stakeholders' productivity using different SRSs.
- RO4 —To investigate the correlations between transparency factors and their metrics.
- RO5 —To investigate the perceptions towards different SRSs.
- RO6 —To propose a transparency-improving model.

RO1-RO3 are motivated to empirically demonstrate the usefulness of the proposed metrics by assessing the maintainability and communication effectiveness of SRS and stakeholders' productivity. RO4 is motivated to provide the metrics' scientific and practical significance. The metrics should correlate with the quality factors they purport to assess. RO5 is motivated to investigate stakeholders' perceptions using various SRS. This investigation is achieved via a debriefing questionnaire, which could also explain some anomalies in results. Based on the proposed transparency measures and metrics, RO6 is motivated to provide a general model that can be followed to improve transparency in SEP generally.

### Research Approach

This study follows a quantitative research design [28]. The quantitative methods adopted for this study include descriptive, correlational, causal, and experimentation. The reason for adopting a quantitative approach is that it suits the problems the research addresses. The selected design enabled us to experiment, collect, and statistically analyse quantitative data about transparency. To define and perform the experiment, we followed the guidelines for conducting SE experiments provided in [29]. The previously proposed conceptual framework [25] for evaluating and improving transparency can be considered our research approach to realising transparency as it embodies the adopted research design.

In implementing the quantitative method, we adopted the Goal Question Metric (GQM) [14] paradigm to develop a transparency-improving model. This model includes transparency measures and metrics. The GQM is a framework for developing software metrics. The adoption of GQM is because, in software measurement research, GQM has long established itself as the most cited approach among researchers. The *goal* in GQM defines a project's primary target. The *questions* define the way an assessment of a goal is performed. The *metric* is data associated with each question that quantitatively answers the question. The GQM assumes that purposeful measurement by organisations can be achieved when goals are specified. These goals should be traceable to data that defines the goals. Additionally, a framework must be provided to interpret the data concerning established goals. Apart from utilising GQM in developing the proposed transparency-improving model, GQM also enabled us to propose measures and metrics for transparency in [30], which are validated in this paper.

## Description of Experiment

This experimental study was conducted to investigate several different effects and relationships concerning transparency. This section presents the experiment's goals, the study's object, and the experimental hypotheses. In performing and reporting the experiment, we adopted the approach provided by Wohlin et al. [29].

### Goal Definition

The goals of this experimental study are as follows:

- G1: To analyse the impact of transparency on the maintainability of SRSs
- G2: To analyse the impact of transparency on the efficient and effective communication of SRSs.
- G3: To analyse the impact of transparency on stakeholders' productivity.
- G4: To analyse the correlations between the quality metrics and constructs of transparency (i.e., whether metrics assess what they purport to measure).
- G5: To analyse stakeholders' perceptions using different types of SRSs.

### Experimental Hypotheses

In order to derive our hypotheses, we posed the following questions:

- Q1: Do object-oriented SRS (OO-SRS) offer more maintainability based on their degree of transparency than a non-object-oriented SRS (NOO-SRS)?
- Q2: Do OO-SRS offer more effective communication based on their degree of transparency than a NOO-SRS?
- Q3: Do OO-SRS offer stakeholders increased productivity based on their transparency than a NOO-SRS?
- Q4: What are the correlations between the quality constructs of transparency?
- Q5: Are there any differences in stakeholders' general perceptions towards using OO-SRS and NOO-SRS?

As previously stated, this paper primarily aims to empirically validate transparency constructs and metrics that we defined previously in [30]. Table 1 presents a summary of the parameters considered in this investigation. These questions will enable us to address the operationalisation and validation of transparency in the contexts of maintainability (Theme A, 14 parameters), communication (Theme B,

**Table 1** Constructs and metrics considered for experimental hypotheses [30]

| Themes | Constructs | Metrics |
|---|---|---|
| Theme A: Maintainability | Modifiability | MT, MEffic, MEffec, MComp, SoI, Mod |
| | Understandability | UnT, UnEffec, UnEffic, UnNotion, RNotion, ConI, RoI, Und |
| Theme B: Communication | Accessibility | PoI, ComI, EFI, Acc |
| | Usability | UsT, UsEffect, UsEffic, EoU, PoU, PU, DIC, SU, Usa |
| | Reusability | CoI, GD, EoR, ARP, Reu |
| | Modifiability | MT, MEffic, MEffec, MComp, SoI, Mod |
| | Understandability | UnT, UnEffec, UnEffic, UnNotion, RNotion, ConI, RoI, Und |
| *Theme C: Productivity | - | UsT, UnT, MT, UsEffec, UsEffic, MEffec, MEffic, UnEffec, UnEffic |
| Theme D: Correlations | All | All |
| **Theme E: Perceptions | - | Mot, Conf, UnTask, CorrUnTask, CorrUsTask, CorrIA, AccIA |

* Metrics under this theme are used directly without their constructs

** Metrics under this theme are not transparency metrics but variables for measuring participants perceptions

32 paramters), and productivity (Theme C, 9 parameters). We also want to determine the correlations (Theme D, 32 parameters) between transparency constructs and metrics and users' perceptions (Theme E, 7 parameters) towards alternative SRSs. Because of the inherent transparency complexity and challenge with its measurement, as we uncovered in our previous effort, these transparency constructs and metrics were carefully elicited and well-defined according to well-known standards in [30]. Therefore, this study is only concerned with empirically validating the pre-selected transparency constructs and metrics to ensure they measure what they purport to measure. Next, we elucidate these questions further by considering them separately under various themes and providing more specific high and low-level hypotheses.

- Theme A: *Comparison of the maintainability of OO-SRS and NOO-SRS based on transparency (RO1).*

Theme A investigates whether different SRSs (i.e., OO-SRS, NOO-SRS) offer better transparency because of the method of their development and presentation. Secondly, as we have argued in our research programme and as highlighted by related studies, the inherent complexity of transparency makes it challenging to measure directly with certainty, hence the need for constructs and metrics. This measurement challenge and the peculiarity of transparency also necessitate using "more" and "less" qualifiers to evaluate the extent of transparency achievement, as seen in the following hypotheses. To help the investigation, we consider question *Q1* and goal *G1*, and state the following high-level hypotheses. We deem them high-level because they cannot be directly assessed by transparency constructs and metrics.

(i) General Null Hypothesis, $\underline{GNH_{i0}}$: A "more transparent" OO-SRS does not offer more maintainability than a "less transparent" NOO-SRS.
(ii) General Alternative Hypothesis, $\underline{GAH_{ii1}}$: A "more-transparent" OO-SRS does offer more maintainability than a "less-transparent" NOO-SRS.
Note: From the general null hypothesis viewpoint, we assume that an OO-SRS is more transparent than a NOO-SRS. This assumption would be rejected if experimental outcomes indicate otherwise or favour the alternative hypothesis.

These high-level hypotheses will enable us to propose the following specific low-level hypotheses that transparency constructs and metrics can directly assess.

(a) Null hypothesis, $\underline{H_{a0}}$: there is no significant difference between the modifiability of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{a1}}$: there is a significant difference between the modifiability of OO-SRS and NOO-SRS.
(b) Null hypothesis, $\underline{H_{b0}}$: there is no significant difference between the understandability of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{b1}}$: there is a significant difference between the understandability of OO-SRS and NOO-SRS. **Note:** From the viewpoint of the null hypothesis, we assume that OO-SRS and NOO-SS have different modifiability and understandability levels (i.e., "more" or "less"). However, we cannot determine which SRS offers more modifiability and understandability at this point. We argue that an SRS with higher modifiability and understandability offers more maintainability and, therefore, more transparency. This argument is in line with the general ($GAH_{ii1}$) and specific alternative hypotheses ($H_{a1}, H_{b1}$) and could lead to a rejection of their corresponding null hypotheses ($GNH_{i0}, H_{a0}, H_{b0}$) if the experimental outcomes suggest otherwise. Table 1 provides the parameters considered for Theme A hypotheses.

Furthermore, the reasoning behind these hypotheses, as we have previously shown, is that we can determine the maintainability of different software artefacts like SRS with the transparency constructs of *understandability* and *modifiability*. According to [31], understandability and modifiability are maintainability constructs. They have been used to assess maintainability in [32, 33]. We argue that an SRS could communicate effectively with stakeholders when functional requirements are captured using an object-oriented technique like use-case modelling, as reported in [5]. It can be more maintainable owing to the benefit object orientation carries with it, as OMG advocates. In light of this, we hypothesise that a "more-transparent" OO-SRS offers more maintainability than a "less-transparent" NOO-SRS.

- Theme B: *Comparison of the effectiveness of communication between OO-SRS and NOO-SRS based on transparency (RO2)*

Poor communication is one of the reasons why software projects fail [34, 35]. A communication model [2, 11] consists of the information sender, the channel, and the receiver of the information. In the context of SEP, we consider the channel to represent the medium (e.g., requirements documents and design documents, code) through which information is conveyed to the receiver (e.g. requirements engineers, developers) of the information. Based on the communication model, we consider effective communication as the successful delivery of information by the sender and its understanding by the receiver. Achieving effective communication results in a "shared understanding" of the information shared during SEP [36]. Software artefacts such as SRS represent communication channels via which information is shared. We consider their effectiveness in providing information based on the proposed transparency constructs and their corresponding metrics. This is part of our conceptual argumentation of transparency and its operationalisation in SEP.

Similarly, Theme B seeks to investigate if alternative SRSs (i.e., OO-SRS or NOO-SRS) provide better communication using transparency. To aid this investigation, we put forward the following hypotheses considering question *Q2* and goal *G2* above.

(i) General Null Hypothesis, $\underline{GNH_{i0}}$: A "more-transparent" OO-SRS does not communicate more effectively than a"less-transparent" NOO-SRS.

(ii) General Alternative Hypothesis, $\underline{GNH_{ii1}}$: A "more-transparent" OO-SRS does not offer more effective communication than a "less-transparent" NOO-SRS.

**Note:** From the general null hypothesis ($GNH_{i0}$) perspective, our study assumes that a more transparent OO-SRS does not provide better communication than a NOO-SRS. Again, depending on experimental results, this assumption could be rejected in favour of the general alternative hypothesis ($GNH_{ii1}$).

Following these high-level or general hypotheses, we can easily derive the following specific low-level hypotheses that can be directly assessed by all transparency constructs and their corresponding metrics.

(a) Null hypothesis, $\underline{H_{a0}}$: There is no significant difference between the modifiability of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{a1}}$: There is a significant difference between the modifiability of OO-SRS and NOO-SRS.

(b) Null hypothesis, $\underline{H_{b0}}$: There is no significant difference between the understandability of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{b1}}$: There is a significant difference between the understandability of OO-SRS and NOO-SRS.

(c) Null hypothesis, $\underline{H_{c0}}$: There is no significant difference between the usability of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{c1}}$: There is a significant difference between the usability of OO-SRS and NOO-SRS.

(d) Null hypothesis $\underline{H_{d0}}$: There is no significant difference between the reusability of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{d1}}$: There is a significant difference between the reusability ofOO-SRS and NOO-SRS.

(e) Null hypothesis $\underline{H_{e0}}$: There is no significant difference between the accessibility of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{e1}}$: There is a significant difference between the accessibility of OO-SRS and NOO-SRS.

**Note:** Again, from the perspective of all the null hypotheses and considering all transparency quality constructs, we assume that OO-SRS and NOO-SS have different levels (i.e., "more" or "less") of accessibility, understandability, reusability, modifiability, and understandability. Again, we cannot determine, at this juncture, which SRS offers more of these qualities that we deem to be helpful to communication. We argue that an SRS with a higher rate of these qualities offers better communication and, hence, more transparency. This aligns with the general ($GNH_{ii1}$) and specific alternative hypotheses ($H_{a1}$, $H_{b1}$, $H_{c1}$, $H_{d1}$, $H_{e1}$) and could lead us to reject their respective null hypotheses ($GNH_{ii0}$, $H_{a0}$, $H_{b0}$, $H_{c0}$, $H_{d0}$, $H_{e0}$) if the experimental outcomes suggest otherwise. Parameters considered for these hypotheses are provided in Table 1.

Whereas *understandability* and *modifiability* can predict maintainability, they also help achieve effective communication. They can predict stakeholders' productivity as they use the artefact. In generality, given the reasoning we advanced in *Theme A*, we hypothesise that a "more-transparent" OO-SRS offers more effective communication than a "less-transparent" NOO-SRS. This result addresses goal *G2* and question *Q2*.

The second perspective is practical. We analyse which representation and design technique offers more transparency regarding artefact maintainability, reusability, effective communication, and stakeholder productivity. Do SEP and stakeholders stand a better chance of significantly gaining from introducing an object-oriented technique in developing SRS than from not introducing it?

It is generally accepted that SRSs be produced in natural language text for the understanding of all stakeholders. There have been suggestions for the use of object-oriented techniques in SRS preparation. However, we found no empirical evidence to support or refute the perceived benefits of its use apart from the study in [5]. We hope our study will contribute significantly to the discourse, at least from the perspective of transparency.

Evidence shows that tools have been developed to check the quality of SRS based on several properties of natural language-based SRS. This check will be illusory to conduct manually. These tools have not been generally accepted because of the perception of requirements engineers in terms of job security. However, there is evidence that checking the quality of SRS is still a requirement of concern for academia and the industry. This problem motivated us to empirically explore the perceived benefit of introducing object-oriented techniques in developing SRS.

- Theme C: *Comparison of stakeholders' productivity based on the use of OO-SRS and NOO-SRS (RO3)*

This theme investigates the difference between stakeholders' productivity when using OO-SRS and NOO-SRS. We argue that a "more transparent" SRS offers stakeholders more productivity than a "less transparent" SRS. To enable this investigation, we generate the following hypotheses based on goal *G3* and question *Q3*. We rely on various transparency metrics, including task completion time (TCT), task efficiency time (TET), and task effectiveness (TE), to assess productivity.

(a) Null hypothesis, $\underline{H_{a0}}$: There is no significant difference in task completion time between participants who used OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{a1}}$: There is a significant difference in task completion time between participants who used OO-SRS and NOO-SRS.
(b) Null hypothesis, $\underline{H_{b0}}$: There is no significant difference in task efficiency time between participants who used OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{b1}}$: There is a significant difference in task efficiency time between participants who used OO-SRS and NOO-SRS.
(c) Null hypothesis, $\underline{H_{c0}}$: There is no significant difference in task effectiveness between participants who used OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{c1}}$: There is a significant difference in task effectiveness between participants who used OO-SRS and NOO-SRS. **Note:** From the standpoint of the null hypothesis, we assume that OO-SRS and NOO-SS have different productivity rates measured in terms of TCT, TET, and TE. Again, at this point, we cannot determine which SRS offers more productivity. We also argue that an SRS with better TCT, TET, and TE offers more productivity and, therefore, more transparency. The null hypotheses ($H_{a0}$, $H_{b0}$, $H_{c0}$) would be rejected in favour of the alternative hypothesis ($H_{a1}$, $H_{b1}$, $H_{c1}$) or retained depending on experimental outcomes. Table 1 presents the transparency parameters we considered for Theme C hypotheses. The peculiarity of productivity measurement implies we could only measure it at the transparency metric level instead of the construct level.

- Theme D: *Correlations between quality constructs of transparency (RO4)*

One of our research goals is to determine the correlations among transparency constructs and metrics. Proposed transparency constructs and metrics should correlate to enhance our understanding and measurement of transparency achievement. This is in addition to measuring what they purport to measure, which is the essence of empirical validation. To help our investigation of correlations, we derive the following hypotheses based on question*Q4* and *G4*.

(a) Null hypothesis, $\underline{H_{a0}}$: there is no significant relationship between accessibility and usability. Alternative hypothesis, $\underline{H_{a1}}$: there is a significant relationship between accessibility and usability.
(b) Null hypothesis, $\underline{H_{b0}}$: there is no significant relationship between accessibility and understandability. Alternative hypothesis, $\underline{H_{b1}}$: there is a significant relationship between accessibility and understandability.
(c) Null hypothesis, $\underline{H_{c0}}$: there is no significant relationship between accessibility and modifiability. lternative hypothesis, $\underline{H_{c1}}$: there is a significant relationship between accessibility and modifiability.
(d) Null hypothesis, $\underline{H_{d0}}$: there is no significant relationship between accessibility and reusability. Alternative hypothesis, $\underline{H_{d1}}$: there is a significant relationship between accessibility and reusability.
(e) Alternative hypothesis, $\underline{H_{e0}}$: there is a significant relationship between usability and understandability. Alternative hypothesis, $\underline{H_{e1}}$: there is a significant relationship between usability and understandability.
(f) Null hypothesis, $\underline{H_{f0}}$: there is no significant relationship between usability and modifiability. Alternative hypothesis, $\underline{H_{f1}}$: there is a significant relationship between usability and modifiability.
(g) Null hypothesis, $\underline{H_{g0}}$: there is no significant relationship between usability and reusability. Alternative hypothesis, $\underline{H_{g1}}$: there is a significant relationship between usability and reusability.
(h) Null hypothesis, $\underline{H_{h0}}$: there is no significant relationship between modifiability and reusability. Alternative hypothesis, $\underline{H_{h1}}$: there is a significant relationship between modifiability and reusability.
(i) Null hypothesis, $\underline{H_{i0}}$: there is no significant relationship between understandability and modifiability.Alternative hypothesis, $\underline{H_{i1}}$: there is a significant relationship between understandability and modifiability.
(j) Null hypothesis, $\underline{H_{j0}}$: there is no significant relationship between understandability and reusability.Alternative hypothesis, $\underline{H_{j1}}$: there is a significant relationship

between understandability and reusability. **Note:** From the perspective of null hypotheses $H_{a0}$-$H_{j0}$, we assume that there are no correlations among all transparency constructs and metrics. Depending on experimental results, these null hypotheses are retained or rejected in favour of the alternative hypotheses $H_{a1}$-$H_{j1}$. All transparency metrics and constructs considered for these hypotheses are presented in Table 1.

These hypotheses were motivated by the need to know how the quality constructs of transparency interrelate. Existing studies [5, 11] have not provided any empirical evidence for interrelationships between transparency constructs. Each of the SRS will be subjected to the above hypotheses. The idea is to understand how both perform under the same model.

The key objective of our study is to provide the theoretical and empirical underpinnings for transparency as an NFR in SE. The above hypotheses are, therefore, driven by the need to theoretically and empirically understand the interrelationships between the constructs of transparency. Consequently, we do not rely on "softgoals" like in [1] to support existing relationships. Additionally, we proposed several measures and metrics for assessing each quality factor of transparency. It is, therefore, vital to determine if the measures measure each quality factor they purported to measure.

- Theme E: *Comparison of perceptions towards the use of OO-SRS and NOO-SRS (RO5)*

Theme E seeks to investigate stakeholders' perceptions about different SRSs (i.e., OO-SRS and NOO-SRS). We argue that "more transparent" SRS evokes more positive perceptions than "less transparent" SRS. Perceptions are measured in terms of their motivations, confidence and understanding of alternative SRSs. We need hypotheses to embark on this investigation, and we derive them from goal *G5* and question *Q5* as follows.

(a) Null hypothesis, $\underline{H_{a0}}$: there is no significant difference in motivation between the use of OO-SRS and NOO-SRS.Alternative hypothesis, $\underline{H_{a1}}$: there is a significant difference in motivation between the use of OO-SRS and NOO-SRS.
(b) Null hypothesis, $\underline{H_{b0}}$: there is no significant difference in confidence between the use of OO-SRS and NOO-SRS.Alternative hypothesis, $\underline{H_{b1}}$: there is a significant difference in confidence between the use of OO-SRS and NOO-SRS.
(c) Null hypothesis, $\underline{H_{c0}}$: there is no significant difference in understanding between the use of OO-SRS and NOO-SRS. Alternative hypothesis, $\underline{H_{c1}}$: there is a significant difference in understanding between the use of OO-SRS and NOO-SRS. **Note:** From the stance of the null hypotheses ($H_{a0}$, $H_{b0}$, $H_{c0}$), we assume that there are differences in perceptions for OO-SRS and NOO-SRS, but we cannot yet determine which SRS accounts for this differences. We also assume that a more transparent SRS offers stakeholders more motivation, confidence and understanding than a less transparent SRS. Based on the result of experiments, these null hypotheses could be retained or rejected in favour of the alternative hypotheses ($H_{a1}$, $H_{b1}$, $H_{c1}$). The parameters taken into account for these hypotheses are provided in Table 1. However, they are not part of the transparency metrics we proposed and aim to validate in this paper, but they enable us to study stakeholders' general perceptions towards different SRSs as part of their debriefing during on completion of the experiment.

## Subjects

Because the research involves human subjects, ethics approval was requested and granted by the research and ethical committee of the researchers' university. Recruiting industry professionals is usually encouraged when performing controlled experiments in SE. However, owing to the difficulty usually experienced in recruiting professionals, our primary target was to recruit final-year computing undergraduate and graduate students. In this study, the experimental subjects were mainly information and communication technology (ICT) professionals and a small group of final and third-year students of the first author's institution. Participants were recruited both online and offline.

We assumed that ICT professionals are generally familiar with different types of SRSs and can perform various tasks, including impact analysis. The targeted student category is already familiar with reading and analysing different requirements specifications because of the software-related courses they previously studied. The students were introduced to impact analysis for the experiment to accomplish the modifiability task. Recruiting student subjects offers some incentives for getting a considerable sample size. The experimental tasks at this stage of RE do not necessarily require a high level of industry expertise. However, following Falessi et al. [37] recommendations, an experiment with students is justified.

We aimed to have subjects participate in the experiment online and offline. Conducting it online makes the subject recruitment global and not limited. The study also aimed to select subjects based on a convenience sampling method. This approach allows us to recruit subjects who are available and ready to participate in the experiment.

## Experimental Materials

The objects of study are software artefacts represented in different formats and developed using different development methodologies and guidelines. Precisely, the objects of the study are (1) SRS, which captures functional requirements represented in natural language texts (denoted as NL-SRS), and (2) SRS, which captures functional requirements in object-oriented, use case model (designated as UCM-SRS). We adopted these SRS from previous research [11]. Adopting published software artefacts is advantageous because they would have already been assessed for validity and reliability. They can also be a baseline for comparing new and previous results [38]. The NL-SRS contains actual functional requirements description of a university accommodation system, totalling 39 pages. The UCM-SRS was designed based on the NL-SRS and totalled 17 pages.

Furthermore, we developed a questionnaire that enabled data collection from participants. The questionnaire did not include objective metrics, such as conciseness and readability. The authors provided the values for this set of metrics. Before administering the questionnaire, the experiment was subjected to expert validation to ascertain its suitability and reliability. The validation outcome was used to improve the questionnaire, making it appropriate for potential participants. The SRS and the questionnaire are provided under Appendices.

## Experimental Tasks

Our experiment followed a between-subject design. Therefore, participants were expected to perform different document review tasks based on a randomly assigned SRS (NL-SRS or UCM-SRS). The first task was usability, which required participants to respond to questions that measured usability after reading SRS. The second task involved participants responding to questions that tested their perceptions about SRS's potential reuse. Like the usability task, the third task required participants to respond to another set of questions that measured their understanding of SRS.

The fourth task asked participants to perform an impact analysis. This task involves marking all places within SRS that require changes without implementing the changes. A fictitious customer requested two changes to the functional requirements. The number of change locations to be identified by participants was four (4) for NL-SRS and six (6) for UCM-SRS. In a study of this nature, it is often challenging to balance the objects of study. As in this case, UCM-SRS had two more places requiring change than NL-SRS. These additional locations were necessitated by adherence to use case modelling standards, which required functional requirements to be defined explicitly. Therefore, requirements are not lumped up for clarity as opposed to the presentation offered by NL-SRS. We ignored these additional locations to reduce variation and avoid bias even though participants correctly identified them. This decision resulted in an equal number of change locations for both treatments. The same rationale was used to judge the number of places where change was found.

On completing the tasks, participants were required to fill out a debriefing questionnaire. The questionnaire captured information including a) participant motivation, b) perceptions about their performance in terms of accuracy and correctness of tasks performed, c) the approach adopted in conducting the tasks, and the difficulties experienced. We have provided an online link to the questionnaire in Appendix A.

## Procedures

We prepared participant consent forms, a participant information sheet (PIS), and a poster to advertise the experiment. The 4th year Computer Science students of the Federal University of Lafia and North-West University were recruited voluntarily.

The recruitment of ICT professionals followed a rigorous approach. We advertised the experiment on various platforms, including ICT-related groups like LinkedIn and Facebook. The response rate improved when group members were privately invited to volunteer for the experiment and followed up with reminder messages. The experiment was also shared with several affiliated social and business networking groups for academics who are in computing.

A web-based questionnaire was created using Google Forms. A link to the questionnaire was provided to participants in the invitation email. The two SRSs were placed in different Google Drive folders. Participants were equally provided a link depending on which SRS they were randomly assigned. After familiarising themselves with information about the experiment, participants were expected to consent by clicking "Yes" or "No."

## Design

The general experimental design type employed is a between-subjects design. The two independent variables are (1) representation techniques (object-oriented or non-object-oriented) used in developing SRS and (2) their assumed degree of transparency ("more-transparent" and "less-transparent"). The choice of the between-subjects approach is to avoid order and carry-over effects such as practice and fatigue. The choice also stems from the fact that one treatment is developed based on the other. The hypotheses we subsequently present do not necessarily require us to select the standard design type in some cases. This is because both treatments will be subjected to the model evaluation.

## Variable Selection and Collection Procedures

Our previous work [30] proposed and described several metrics for explaining and assessing transparency. The metrics aim to measure the five quality constructs of transparency that we equally proposed. Therefore, the variables for the experiment are the quality constructs of transparency and their corresponding metrics. During the experiment, variables (i.e., constructs and metrics) were collected using a questionnaire. Given the goals of this study, the peculiarity of the correlations we seek to analyse and determine means that a variable could be a dependent variable in one instance. It could also be an independent variable in another instance.

## Data Analysis Procedure

The experimental runs resulted in the collection of 273 responses. The responses were screened, and 21 responses were expunged. These responses were either incomplete or invalid in several cases. Therefore, 252 data points were available for analysis. The experiment design is between-subjects; therefore, we considered non-repeated measures analysis. Some of the proposed metrics are subjective and taken on the Likert scale, which is ordinal. This data category was transformed into a numeric form and computed as sums. We then went on to use the transformed values in data analysis. Transformed Likert scale data could be used for parametric statistical tests (e.g., t-test) without arriving at a wrong conclusion [39]. The result of our analysis lends credence to this, as we will discuss in a subsequent related section.

In order to avoid experimental risks and to follow conventional wisdom, we considered power analysis. Power analysis increases the likelihood that an effect will be found for the phenomenon under empirical study [40]. Using a high power level supports the idea that a statistical test will report a significant result with a high probability that the effect exists. In this vein, a Type II error is unlikely to be committed. Assessing effect sizes in addition to statistical significance is necessary to avoid arriving at the wrong or imprecise conclusions during hypothesis testing. The reliance on *p*-values is deemed insufficient since it only indicates statistical significance. In contrast, effect size indicates the practical significance and meaningfulness of the result.

Going forward, we preset and present the statistical power and the effect sizes we aim to achieve in this study. We also preset the significance level, i.e., the *p*-value for the current study. Several factors need to be considered to preset the *p*-value. Firstly, we assess the implication of committing a Type I error, i.e., wrongly rejecting the true null hypothesis. In the context of our research, incorrectly rejecting a null hypothesis would mean the cost of adopting non-transparent models, e.g., poor communication among stakeholders and low productivity. Secondly, we consider the goals of the current study. We discuss this from the following two standpoints:

*A practical standpoint:* Here, we ask whether adopting object-oriented standards such as use-case modelling is more likely to be beneficial to achieving transparency than otherwise. Benefits are considered regarding SRS maintainability, effective communication, and productivity.

*A scientific standpoint:* To identify cause-effect relationships between SRS standards and the maintainability and communication effectiveness of the SRSs produced.

A scientific standpoint favours a *p*-value of 0.05 or 0.01. Applying a stringent *p*-value may not be practical, depending on the cost of adopting object-oriented development standards. What is rather vital is the minimisation of risk and possible loss. In this circumstance, a *p*-value in the range of 0.2 could be deemed sufficient to render an informed decision. This outcome is helpful even without empirical evidence that strongly supports a scientific statement of high import [32]. The empirical study aims to identify potentially interesting and statistically significant trends in transparency application to focus future studies. It aims to enable us to garner insights into the consequences of adopting object-oriented standards to achieve transparency in SEP.

The third factor for consideration is the possibility of obtaining a statistically non-significant result. The result could be explained by a) the non-existence of an effect, leading to the retention of the null hypothesis, or b) the experiment's statistical power is insufficient to detect the phenomenon being investigated. As earlier mentioned, performing power analysis reduces the probability of incorrectly retaining a false null hypothesis, i.e., a Type II error. The power of a statistical test is based on three different factors: the number of subjects ($N$), the size of the effect being investigated ($\gamma$), and the probability of error of Type I ($\alpha$). One approach to power analysis is to rely on previous empirical studies in the area that have reported effect size. This is often considered a problematic approach to adopt owing to limited studies. An alternative approach is the judgment approach, where the researcher decides the effect size he aims to achieve. A previous related study [5] did not fully report on power analysis. However, the study showed that the effect size for the t-test was computed during statistical analysis. For a sample size of 58 and a *p*-value of 0.05, effect sizes for three variables (time, correctness, and confidence) were given as 0.76, 0.89, and 1.07, respectively. We performed a post hoc statistical power analysis, which showed that the previous experiment generally had sufficient power in the recommended threshold of 0.80.

In this study, the preset *p*-value and statistical power we worked with are SE recommended values of 0.05 and 0.8, respectively. It is worth noting that a small effect size implies there is an effect, and the outcome is not trivial or by chance [40]. In our research, it is reasonable to consider small effect sizes. This scenario is despite the threshold we have specified because the concept of transparency is such that its non-realisation does not impact software functioning. It is, therefore, an orthogonal non-functional requirement. In this vein, a stringent *p*-value and effect size may not be necessary. To determine sample sizes, we computed size for multiple regression analysis and model fitness using Cohen's [40] recommended medium (0.35) and high (0.5) effect sizes in each case. For regression analysis and 26 variables under consideration, a minimum sample size of 88 was required. A minimum sample size of 113 was required for model fitness, and a minimum of 38 was required to detect an effect. We did not need to define anticipated sample sizes to perform a t-test possibly. Realising the minimum sample sizes in the above two instances would be sufficient to compute the t-test. The minimum sample size we aimed to achieve was 226 subjects for both objects of study.

Because statistical power can influence the statistical significance of results, low power will be considered in the case of non-significant results. This scenario requires consideration of practical significance. Practical significance is considered when determining whether the phenomenon studied practically and meaningfully impacts the dependent constructs [40–42]. In other words, the phenomenon is large enough to interest stakeholders. In this study, we will compute the observed effect size ($\gamma$) detected for each dependent variable for each hypothesis. A measure of effect size is expressed as the mean difference of two samples divided by the root mean square of the variances of the two given samples. The study aims to discuss both statistically and practically significant results. Furthermore, the current study will consider effects where $\gamma \geq 0.2$, i.e., a small to large effect. This decision is supported by the effect size indices provided by Cohen.

## Experimental Results

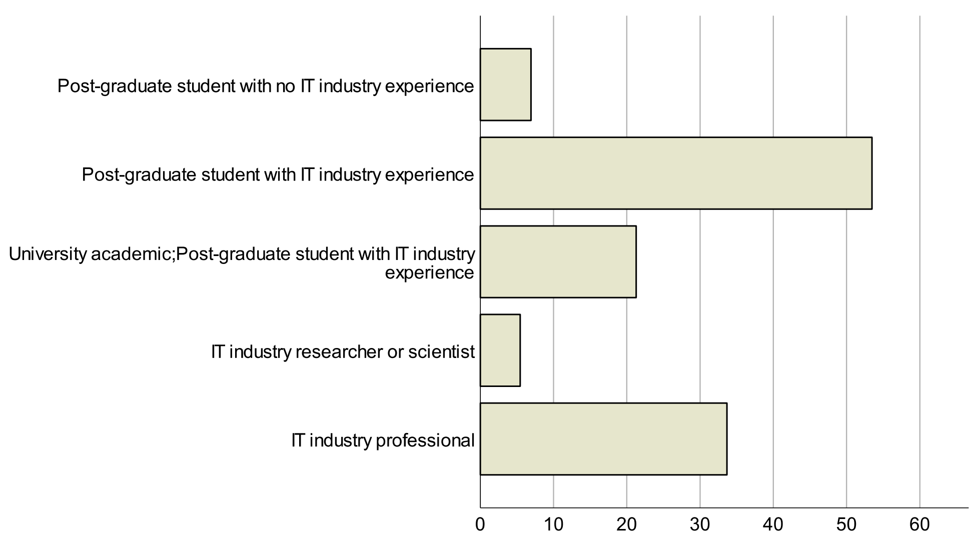
This section presents the experiment's outcome following the experimental goals and hypotheses we outlined in the previous chapter. Data were analysed using IBM SPSS version 28.0.1(15)

### Demographics

*Dataset overview*: The dataset used in this study was collected solely for this research and is publicly accessible on IEEE's Dataport [43]. It captures empirical data about the transparency of different SRSs based on previously proposed transparency constructs and metrics [30]. The data enabled our investigation of transparency and support for software maintainability, effective and efficient communication, and stakeholders' productivity. A questionnaire was used to obtain the data during an in-person lab and online experiment with SRSs. Experimental participants (including students-professionals and industry professionals) were chosen via convenient sampling, and experimental materials (SRSs) were assigned randomly. More details on demographics, subjects, and procedures, among others are provided under sections Subjects, Procedures, Data Analysis Procedure, Variable Selection and Collection Procedures, Anomalies in the Data. The dataset comprises numerical, Likert-scale, and qualitative data types. The data was pre-processed before importing into SPSS software for further analysis.

Of the original 273 participants recruited, 252 were retained after data screening. Of the 252 participants, 210 were considered professionals (experts) with academic and industry backgrounds. It also includes postgraduate students who may or may not have industry experience. Figures 1, 2, and 3 indicate their various occupations and specific roles

**Fig. 1** Number of experimental participants and their occupation



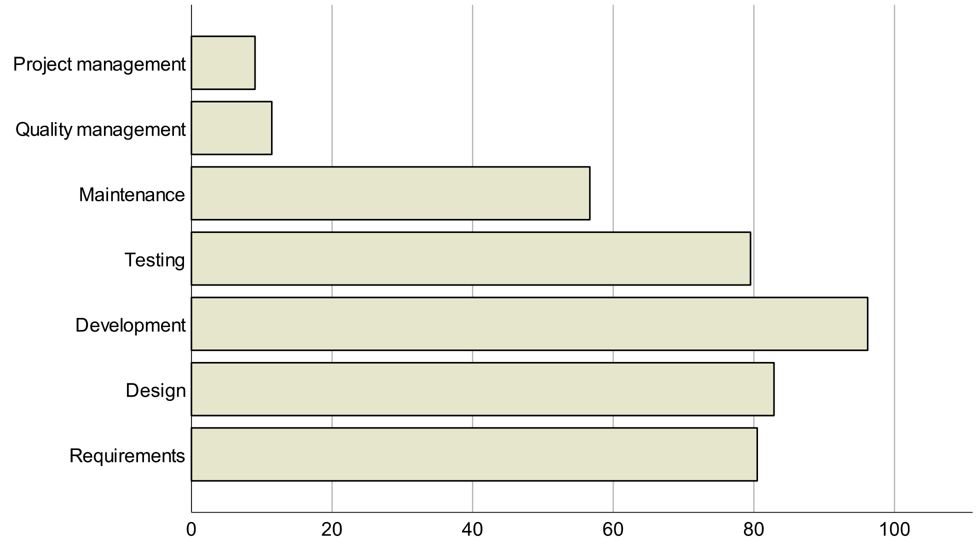



**Fig. 2** Number of experimental participants and their roles in software engineering process

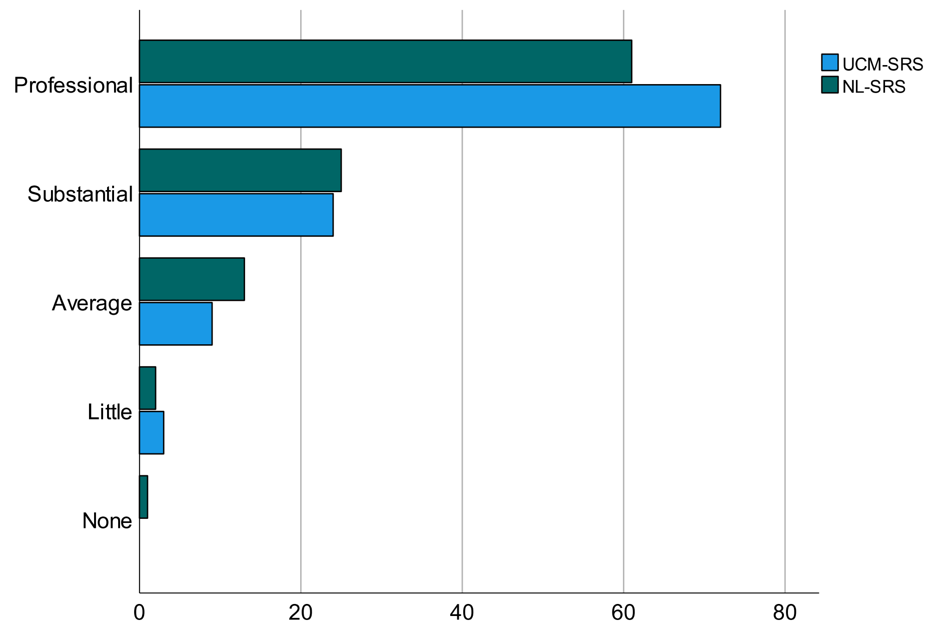



**Fig. 3** Number of experimental participants and their experience in software engineering practice

in the industry and experience, respectively. A significant percentage (48.1) of participants have 0–4 years of experience, and 38.6% have 5–9 years of experience. Also, 12.9% have 10–15 years of experience, and 0.5% have 15–19 years of experience. The remaining 42 participants were final (24) and third-year (16) undergraduate (novices) computer science students of authors' institutions. Of the 42 participants, 12 have industry experience with about 0–4 years of working experience.

The study sought to gauge the experience of participants in terms of the following: software practice (Fig. 4), general experience with software requirements documents (Fig. 5), experience with object-oriented software requirements documents (Fig. 6), familiarity with reuse design guidelines (Fig. 7), and experience performing impact analysis (Fig. 8). With the aid of charts, we present the outcome of these findings. Undergraduate participants were not targeted for these findings. We analysed and grouped the findings according to the software requirements document assigned to each participant during the experiment.

We deduced from the outcomes that professional participants have adequate professional and substantial experience in software practice and software requirements documents. They are equally very familiar with software reuse guidelines and, as expected, far more in number than undergraduate students.

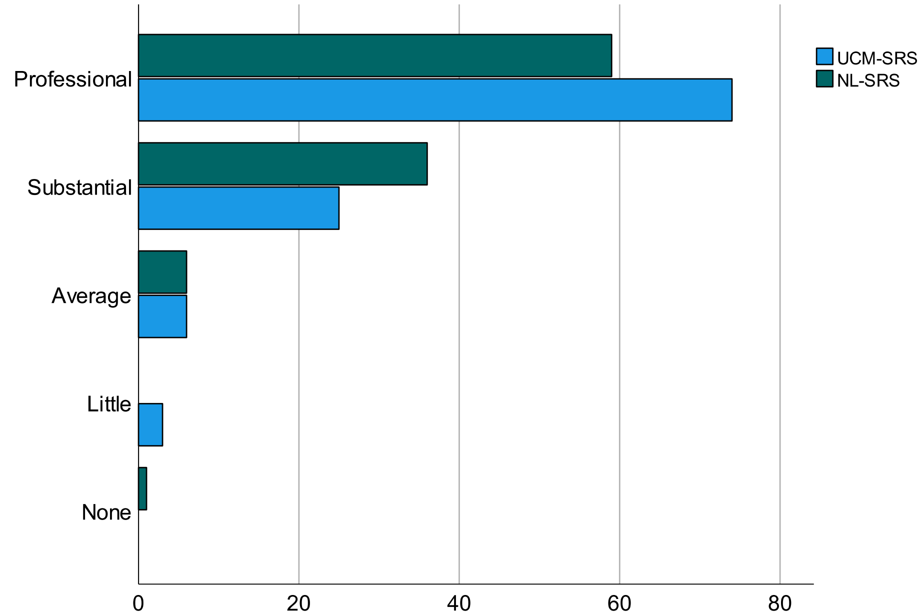



**Fig. 4** Number of experimental participants and their software requirements experience

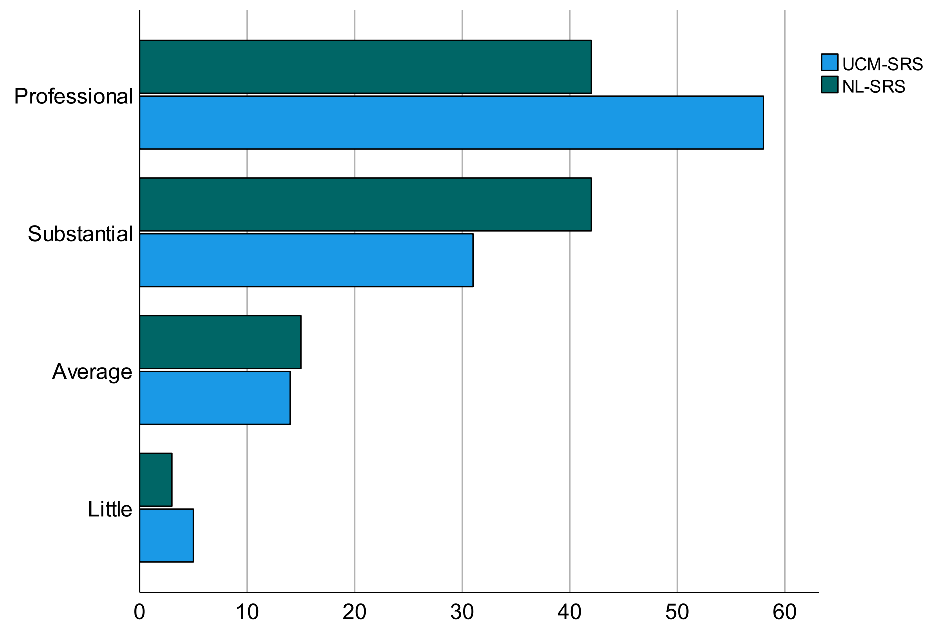



**Fig. 5** Number of experimental participants and their object-oriented software requirements experience

## Anomalies in the Data

In SE experiments, anomalies in data are bound to be observed owing to varying degrees in participants' ability [29]. To help explain data anomalies, especially when small sample sizes are involved, subject debriefing questionnaires are considered useful [32]. Our experiment worked with relatively high sample sizes of 129 and 123 for each treatment. Therefore, we did not envisage many anomalies in the data and results.

Expected anomalies in a result dataset could also be attributed to outliers in the data. We assessed the 252 data points for multivariate outliers using a Mahalanobis Distance Test [44]. We identified nine (9) outliers (2 for the "more transparent" SRS and 5 for the "less transparent" SRS (see Fig. 9)). Concerning the distribution of the outliers, the "more transparent" SRS consists of a novice and an expert. On the other hand, the "less transparent" SRS consists of two experts and three novices, respectively. To obtain a balanced dataset between both treatments, 9 data points (with low $p$-values) from the "more transparent" SRS were excluded. This resulted in 118 data points for each treatment.

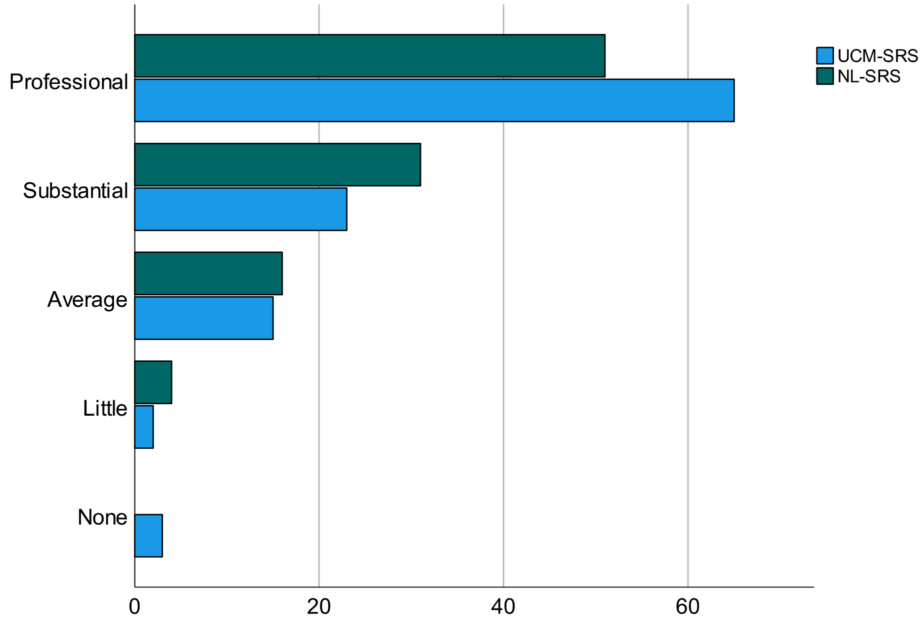



**Fig. 6** Number of professional participants and their impact analysis experience

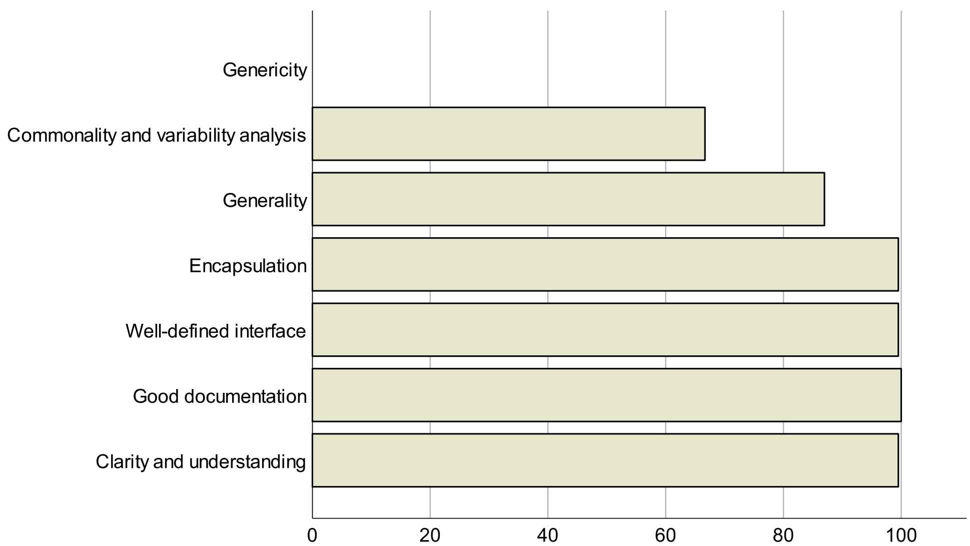



**Fig. 7** Number of professional participants and their familiarity with software reuse guidelines

## Maintainability of Software Requirements Specifications–Theme A (RO1)

This section explores the perceived benefits of transparency in enabling the maintainability of software requirements specifications as previously provided by goal *G1*. An independent t-test was used for this analysis. We consider the transparency quality constructs of modifiability and understandability together with their corresponding metrics.

### $H_{a1}$–Ease of Modification

Table 2 presents a summary of the statistical means ($\bar{x}$), number of participants ($N$), the median ($\tilde{m}$), minimum ($Min$), maximum ($Max$), and standard deviation ($s$) of all constructs related to maintainability including the overall computed modifiability (Mod) and understandability (Und) constructs. Based on Table 2, we can determine how constructs performed, especially regarding their means. Also, Table 3 presents a summary of the statistical tests for five dependent variables related to $H_{a0}$. The table indicates no statistically significant difference in the amount of time (MT) participants spent performing the impact analysis tasks though a negligible effect size was reported for the variable. There was a statistically significant difference between the variables MEffec, MEffic, MComp, and SoI, which measured the effectiveness, efficiency, and completeness of the modification tasks and information size. The effect size (see Table 3 column 2) detected for MT was lower than the minimum threshold for Cohen's proposed indices. However, the

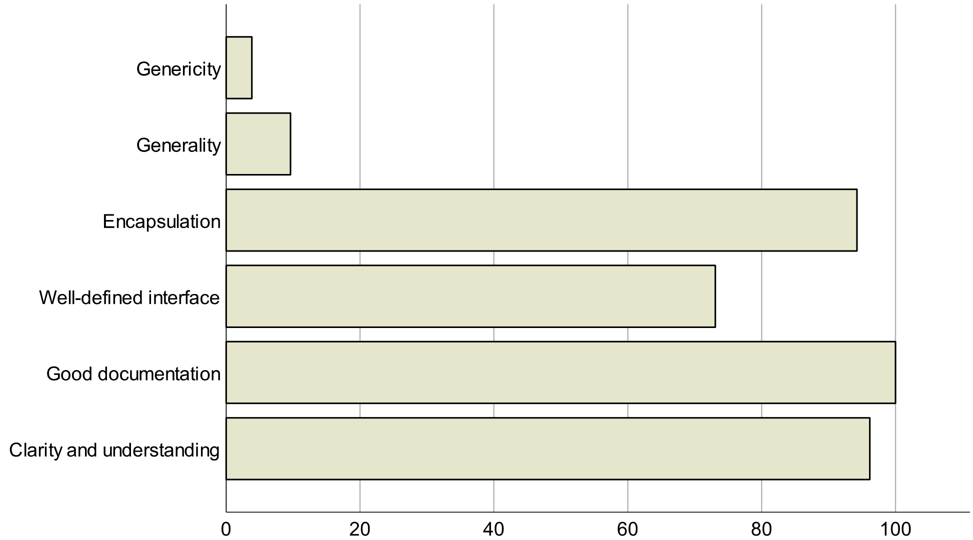



**Fig. 8** Number of student participants and their familiarity with reuse guideline

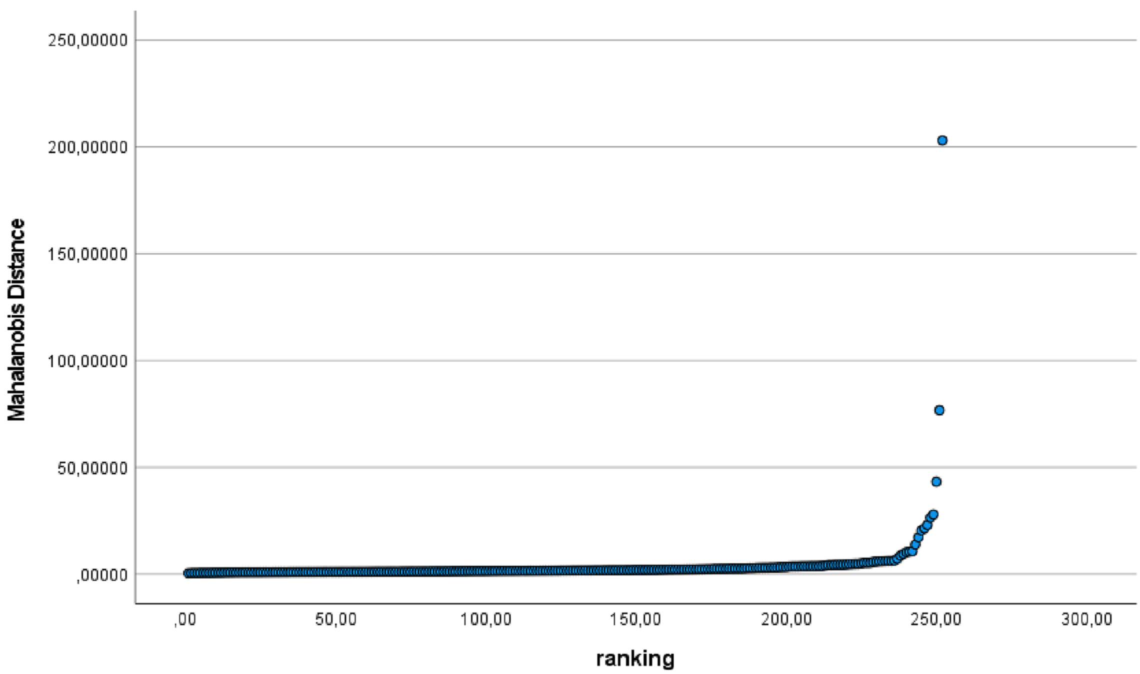



**Fig. 9** The Mahalanobis jackknife distance for outliers

other variables produced medium to large effect sizes in the threshold of our acceptable value for practical significance. Overall, Mod was also found to be statistically significant. The results favour alternative hypothesis $H_{a1}$ and the null hypothesis is, therefore, rejected. Several factors might explain the non-statistically significant difference recorded for MT. We will next consider why it is reasonable to reconsider this opposite outcome, hence the need to include a debriefing questionnaire.

An examination of the means for MT in Table 2 shows that UCM-SRS participants spent less time (4.69) than the mean time of 4.63 by NL-SRS participants. Though this outcome does not support the alternative hypothesis, it is not yet statistically significant, as the effect size for MT further indicates. A further examination of the mean of MEffec showed that UCM-SRS recorded 0.40, whereas NL-SR was 0.34. This result is also supported by the fact that UCM-SRS participants reported more correct answers to the impact analysis than NL-SRS. They also recorded higher efficiency (MEffic) in performing the modification tasks and completed more questions (MComp). UCM-SRS reported a more correct number of places requiring

**Table 2** Descriptive statistics for dependent variables for maintainability

| | More transparent UCM-SRS | | | | | | Less transparent NL-SRS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Variable | *N* | $\bar{x}$ | $\tilde{m}$ | *Min* | *Max* | *s* | *N* | $\bar{x}$ | $\tilde{m}$ | *Min* | *Max* | *s* |
| UnT | 118 | 5.31 | 3.00 | 1.00 | 30.00 | 4.60 | 118 | 5.70 | 4.00 | 1.00 | 21.00 | 3.73 |
| UnEffec | 118 | 3.75 | 4.00 | 0.0 | 4.0 | 0.71 | 118 | 3.26 | 4.00 | 0.0 | 4.00 | 1.15 |
| UnEffic | 118 | 0.55 | 0.70 | 0.0 | 2.00 | 0.30 | 118 | 0.40 | 0.40 | 0.00 | 1.00 | 0.22 |
| UnNotion | 118 | 4.58 | 6.00 | 1.0 | 6.00 | 1.91 | 118 | 3.86 | 4.0 | 1.00 | 6.00 | 1.29 |
| RNotion | 118 | 4.58 | 6.00 | 1.0 | 6.00 | 1.90 | 118 | 3.80 | 4.00 | 1.00 | 6.00 | 1.25 |
| ConI | 118 | 0.50 | 0.50 | 0.50 | 0.5 | 0.00 | 118 | 0.80 | 0.80 | 0.50 | 0.80 | 0.03 |
| RoI | 118 | 0.40 | 0.40 | 0.40 | 0.4 | 0.00 | 118 | 0.40 | 0.40 | 0.40 | 0.40 | 0.00 |
| Und | 118 | 14.34 | 17.40 | 6.0 | 18.90 | 4.21 | 118 | 12.50 | 12.8 | 3.20 | 18.20 | 3.18 |
| MT | 118 | 4.69 | 3.0 | 0.0 | 40.0 | 4.67 | 118 | 4.63 | 3.0 | 0.00 | 30.0 | 5.37 |
| MEffec | 118 | 0.40 | 0.40 | 0.00 | 1.50 | 0.21 | 118 | 0.34 | 0.50 | 0.0 | 0.80 | 0.22 |
| MEffic | 118 | 0.45 | 0.50 | 0.00 | 1.00 | 0.28 | 118 | 0.32 | 0.30 | 0.0 | 1.00 | 0.25 |
| MComp | 118 | 0.40 | 0.50 | 0.00 | 0.50 | 0.17 | 118 | 0.25 | 0.30 | 0.0 | 0.50 | 0.16 |
| SoI | 118 | 17.0 | 17.0 | 17.0 | 39.0 | 5.57 | 118 | 38.81 | 39.0 | 17.0 | 39.0 | 2.03 |
| Mod | 118 | 18.23 | 18.40 | 17.0 | 19.10 | 0.56 | 118 | 0.911 | 1.10 | 0.00 | 2.00 | 0.59 |

**Table 3** Independent samples t-test for dependent variables for modification

| Variable | $\gamma$ | t-value | df | *p*-value |
|---|---|---|---|---|
| MT | 0.02 | 0.09 | 229.70 | |
| MEffec | 0.25 | 1.88 | 231.08 | |
| MEffic | 0.50 | 3.77 | 229.45 | 0.00 |
| MComp | 0.85 | 6.55 | 233.65 | 0.00 |
| SoI | 3.41 | 117.00 | 117.00 | 0.00 |
| Mod | 13.5 | 103.35 | 132.03 | 0.00 |

**Table 4** Independent samples t-test for dependent variables for understanding

| Variable | $\gamma$ | t-value | df | *p*-value |
|---|---|---|---|---|
| UnT | 0.10 | 0.73 | 224.64 | |
| UnEffec | 0.51 | −3.89 | 194.23 | 0.00 |
| UnEffic | 0.61 | −4.70 | 214.06 | 0.00 |
| UnNotion | 0.44 | −3.40 | 205.31 | 0.00 |
| RNotion | 0.46 | −3.73 | 202.90 | 0.00 |
| ConI | 0.45 | 117.00 | 117.00 | 0.00 |
| RoI | 0.42 | 3.66 | 192.90 | 0.00 |
| Und | 0.50 | −3.80 | 217.62 | 0.00 |

modification in relation to the total number of locations they found. It is important to note that the correctness of answers formed part of the components for computing MEffec. Again, this outcome supports our hypothesis that UCM-SRS is more effective than NL-SRS. Considering the overall modifiability SRS at the construct level, the UCM-SRS reported higher modifiability with a mean of 18.23 compared to 0.91 reported by NL-SRS.

### $H_{b1}$–Ease of Understanding

Table 4 presents the result summary of the statistical tests for the five dependent variables concerned with $H_{b0}$ and the computed overall understandability (Und). The dependent variable is represented in columns one and two, the effect size detected, and column three, the t-value of independent samples t-test. Column four degrees of freedom, and column five reports the *p*-value if it is below 0.05.

Table 4 indicates no statistically significant difference in the time (UnT) spent performing the understandability task. However, the table shows a statistically significant difference for UnEffec, UnEffic, UnNotion, RNotion, ConI, and RoI variables. These variables measured the effectiveness of the task, efficiency, understanding, and readability perceptions of the participants, conciseness, and readability indices of the SRS. We also found a statistically significant difference in overall understandability computed from all contributing variables. The effect sizes recorded for all variables were well above our preset value of 0.2 for practical significance. This statistical outcome supported the alternative hypothesis $H_{b1}$.

To account for SRS, which gained from the significant difference, we used Table 2, which provides SRS's means and standard deviations. According to Table 2, participants who worked with UCM-SRS spent less time completing the understanding task. Their mean-time was 5.31 lower than NL-SRS's 5.70, though this difference was not statistically significant. Regarding UnEffec, UCM-SRS participants answered more correct questions and were more effective than NL-SRS participants. They polled a mean of 3.75 compared to NL-SRS's mean of 3.26.

The UCM-SRS participants performed the understanding task more efficiently (UnEffic) than NL-SRS

**Table 5** Descriptive statistics for dependent variables for effective communication

| | More transparent UCM-SRS | | | | | | | Less transparent NL-SRS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Variable* | *N* | $\bar{x}$ | $\tilde{m}$ | *Min* | *Max* | *s* | *N* | $\bar{x}$ | $\tilde{m}$ | *Min* | *Max* | *s* |
| Acc | PoI | 118 | 4.48 | 6.0 | 1.0 | 6.0 | 1.95 | 118 | 3.91 | 4.5 | 1.0 | 6.0 | 1.59 |
| | ComI | 118 | 4.55 | 6.0 | 1.0 | 6.0 | 1.96 | 118 | 3.97 | 4.0 | 1.0 | 6.0 | 1.12 |
| | EFI | 118 | 5.43 | 6.0 | 3.0 | 6.0 | 0.75 | 118 | 4.57 | 5.0 | 1.0 | 6.0 | 0.70 |
| | Acc | 118 | 14.46 | 17.0 | 6.0 | 18.0 | 4.06 | 118 | 12.44 | 14.0 | 3.0 | 18.0 | 2.81 |
| Usa | UsT | 118 | 4.89 | 3.0 | 1.0 | 20.0 | 4.09 | 118 | 6.42 | 4.0 | 2.0 | 41.0 | 6.41 |
| | UsEffec | 118 | 3.81 | 4.0 | 0.0 | 4.0 | 0.54 | 118 | 3.07 | 3.0 | 0.0 | 4.0 | 0.94 |
| | UsEffic | 118 | 0.61 | 0.70 | 0.0 | 2.0 | 0.32 | 118 | 0.34 | 0.35 | 0.0 | 0.8 | 0.10 |
| | EoU | 118 | 4.36 | 5.0 | 1.0 | 6.0 | 1.94 | 118 | 4.03 | 4.0 | 1.0 | 6.0 | 1.62 |
| | PoU | 118 | 4.40 | 5.0 | 1.0 | 6.0 | 1.83 | 118 | 3.54 | 3.5 | 1.0 | 6.0 | 1.21 |
| | PU | 118 | 4.47 | 6.0 | 1.0 | 6.0 | 1.96 | 118 | 3.90 | 5.0 | 1.0 | 6.0 | 1.77 |
| | DIC | 118 | 4.53 | 6.0 | 1.0 | 6.0 | 1.92 | 118 | 3.86 | 4.0 | 1.0 | 6.0 | 1.46 |
| | SU | 118 | 4.52 | 5.0 | 1.0 | 6.0 | 1.80 | 118 | 3.50 | 4.0 | 1.0 | 6.0 | 1.31 |
| | Usa | 118 | 26.7 | 31.7 | 8.0 | 35.0 | 9.43 | 118 | 22.23 | 24.05 | 8.4 | 33.7 | 6.66 |
| Und | UnT | 118 | 5.31 | 3.00 | 1.00 | 30.0 | 4.60 | 118 | 5.70 | 4.00 | 1.00 | 21.0 | 3.73 |
| | UnEffec | 118 | 3.75 | 4.0 | 0.0 | 4.0 | 0.71 | 118 | 3.26 | 4.0 | 0.0 | 4.0 | 1.15 |
| | UnEffic | 118 | 0.55 | 0.7 | 0.0 | 2.0 | 0.30 | 118 | 0.40 | 0.40 | 0.00 | 1.0 | 0.22 |
| | UnNotion | 118 | 4.58 | 6.0 | 1.0 | 6.0 | 1.91 | 118 | 3.86 | 4.0 | 1.00 | 6.0 | 1.30 |
| | RNotion | 118 | 4.58 | 6.0 | 1.0 | 6.0 | 1.9 | 118 | 3.80 | 4.0 | 1.0 | 6.0 | 1.25 |
| | ConI | 118 | 0.50 | 0.50 | 0.5 | 0.50 | 0.00 | 118 | 0.80 | 0.80 | 0.5 | 0.8 | 0.28 |
| | RoI | 118 | 0.4 | 0.4 | 0.4 | 0.4 | 0.00 | 118 | 0.40 | 0.4 | 0.4 | 0.4 | 0.0 |
| | Und | 118 | 14.34 | 17.40 | 6.0 | 18.9 | 4.21 | 118 | 12.50 | 12.8 | 3.20 | 18.2 | 3.18 |
| Mod | MT | 118 | 4.70 | 3.0 | 0.0 | 40.0 | 4.68 | 118 | 4.63 | 3.0 | 0.0 | 30.0 | 5.38 |
| | MEffec | 118 | 0.4 | 0.4 | 0.0 | 1.5 | 0.21 | 118 | 0.34 | 0.5 | 0.0 | 0.8 | 0.23 |
| | MEffic | 118 | 0.45 | 0.50 | 0.0 | 1.0 | 0.28 | 118 | 0.32 | 0.3 | 0.0 | 1.0 | 0.25 |
| | MComp | 118 | 0.40 | 0.5 | 0.0 | 0.5 | 0.17 | 118 | 0.25 | 0.3 | 0.0 | 0.5 | 0.16 |
| | SoI | 118 | 17.0 | 17.0 | 17.0 | 17.0 | 0.00 | 118 | 38.81 | 39.0 | 17.0 | 39.0 | 2.02 |
| | Mod | 118 | 18.23 | 18.40 | 17.0 | 19.1 | 0.56 | 118 | 0.911 | 1.10 | 0.00 | 2.00 | 0.59 |
| Reu | EoU | 118 | 4.53 | 5.0 | 1.0 | 6.0 | 1.80 | 118 | 3.95 | 4.0 | 1.0 | 6.0 | 1.45 |
| | CoI | 118 | 4.58 | 5.0 | 1.0 | 6.0 | 1.73 | 118 | 3.87 | 4.0 | 1.0 | 6.0 | 1.47 |
| | GD | 118 | 4.48 | 5.0 | 1.0 | 6.0 | 1.62 | 118 | 3.56 | 4.0 | 1.0 | 6.0 | 1.18 |
| | EoR | 118 | 3.97 | 4.0 | 1.0 | 6.0 | 1.38 | 118 | 3.41 | 4.0 | 1.0 | 6.0 | 1.11 |
| | ARP | 118 | 4.08 | 4.0 | 1.0 | 6.0 | 1.38 | 118 | 3.51 | 4.0 | 1.0 | 6.0 | 1.18 |
| | Reu | 118 | 21.64 | 25.0 | 7.0 | 30.0 | 7.36 | 118 | 18.29 | 20.5 | 5.0 | 28.0 | 5.71 |

participants. They recorded a mean of 0.55, higher than the NL-SRS mean of 0.40. UCM-SRS participants reported that the structure and presentation of the assigned SRS added to their understanding (UnNotion) and readability (RNotion). Both measures reported an approximate mean of 4.58 compared to the NL-SRS mean of 3.86 and 3.80 for both measures, respectively. The objective metrics (ConI and RoI) reported means of 0.50 and 0.40 for UCM-SRS and 0.80 and 0.40 for NL-SRS. At the construct level, the UCM-SRS reported a high understandability with a mean of 14.34 compared to NL-SRS's 12.8. Based on the statistical outcomes for modifiability and understandability, it can be concluded that UCM-SRS is more maintainable than NL-SRS.

## Effectiveness of Communication–Theme B (RO2)

This section presents the experiment's outcome concerning achieving effective communication in SEP *(G2, Q2)* using transparency. An independent t-test was also utilised for this statistical analysis. We compared the "more transparent" SRS with the "less transparent" SRS based on the quality constructs of transparency and their corresponding metrics. We compared the "more transparent" SRS with the "less transparent SRS based on transparency constructs and their corresponding metrics. Understandability and modifiability constructs have been previously considered concerning maintainability. They equally contribute to effective communication; therefore, they are not considered in this section.

### $H_{c0}$–Usability

Table 5 summarises all constructs' statistical means and standard deviations and their various contributing variables. Table 6 summarises the statistical test results for the eight dependent variables concerning $H_{c0}$ and the overall computed usability construct (Usa). The dependent variable is represented in columns one and two, the size of the effect detected, and in column three, the t-value of the independent samples t-test. Column four, degrees of freedom, and column five report the *p*-value if it is below 0.05. The table indicates a statistically significant difference for the variables UsT, UsEffec, UsEffic, PU, PoU, DIC, and SU. These variables assessed the time participants spent completing the usability task, the effectiveness of the task, and the efficiency in completing the task. The other variables measured participants' perceptions of preference, ease of use, degree of information connectedness, and satisfaction derived from usage. Overall usability Usa was also statistically significant. All the variables reported medium to large effect sizes within our preset threshold for practical significance.

An examination of Table 5 regarding the means for the usability variables revealed that UCM-SRS participants spent less time performing the usability task than NL-SRS participants. UCM-SRS participants performed the usability task equally and more effectively than NL-SRS participants. The mean for effectiveness was 3.81 compared to NL-SRS mean of 3.07. NL-SRS recorded a mean of 0.34 for efficiency compared to UCM-SRS's mean of 0.61.

For the subjective variables of PoU, DoC, and SU, UCM-SRS participants reported they found the UCM-SRS they prefer its use. They found the information SRS provided more connected to the usability task they performed and felt satisfied using UCM-SRS. The means for these variables, as shown in Table 9, were higher than the means of NL-SRS participants. This is, however, not the case with EoU, which favoured NL-SRS. At construct levels, UCM-SRS performed better, with a Usa mean 26.7 compared to NL-SRS's Usa of 22.3. The results support the alternative hypothesis $H_{c1}$.

**Table 6** Independent samples t-test for dependent variables for ease of use

| Variable | $\gamma$ | t-value | df | *p*-value |
|---|---|---|---|---|
| UsT | 0.28 | 2.18 | 198.60 | 0.00 |
| UsEffec | 0.97 | −7.48 | 186.24 | 0.00 |
| UsEffic | 1.03 | −7.88 | 196.64 | 0.00 |
| PU | 0.31 | −2.37 | 231.52 | 0.00 |
| EoU | 0.18 | −1.42 | 226.88 | |
| PoU | 0.55 | −2.24 | 203.11 | 0.00 |
| DIC | 0.40 | −3.01 | 218.48 | 0.00 |
| SU | 0.65 | −4.97 | 214.50 | 0.00 |
| Usa | 0.36 | −2.74 | 233.48 | 0.00 |

### $H_{d1}$–Reusability

Table 7 summarises the result of the statistical tests for the four dependent variables concerning $H_{d0}$, including the overall computed reusability construct (Reu). The table shows a statistically significant difference for the variables EoU, CoI, GD, EoR, and ARP that measured the ease of understanding, clarity of information, ease of reuse, and adherence to reusability principles, respectively. According to Cohen's proposed indices, the variables reported medium to large effect sizes. The reported effect sizes were above our minimum acceptable value for practical significance.

Considering their means as provided in Table 5, UCM-SRS reported higher means for all variables measuring reuse compared to the lower means produced by NL-SRS participants. Based on these results, UCM-SRS is perceived to be more reusable than the NL-SRS as it is easy to understand. It also provided more clarity and better documentation. UCM-SRS participants also reported that UCM-SRS adhered more to reusability design principles and could be easily reused. Considering a comparison at the construct level, the UCM-SR reported more reusability (Reu mean=21.64) than NL-SRS, which has a mean of 18.29. The results favour the retention of the alternative hypothesis $H_{d1}$.

**Table 7** Independent samples t-test for dependent variables for ease of reuse

| Variable | $\gamma$ | t-value | df | p-value |
|---|---|---|---|---|
| EoU | 0.35 | −2.67 | 230.18 | 0.00 |
| CoI | 0.44 | −3.41 | 228.24 | 0.00 |
| GD | 0.65 | 11.81 | 204.34 | 0.00 |
| EoR | 0.45 | −5.00 | 213.86 | 0.00 |
| ARP | 0.44 | −3.37 | 226.10 | 0.00 |
| Reu | 0.51 | −3.91 | 220.44 | 0.00 |

**Table 8** Independent samples t-test for dependent variables for ease of access

| Variable | $\gamma$ | t-value | df | p-value |
|---|---|---|---|---|
| PIC | 0.33 | −2.50 | 224.76 | 0.00 |
| EFI | 1.20 | −9.20 | 232.10 | 0.00 |
| ComI | 0.37 | −2.81 | 186.50 | 0.00 |
| Acc | 0.58 | −43.46 | 208.16 | 0.00 |

#### $H_{e1}$–Accessibility

Table 8 summarises the outcomes of the result for three out of the initial four dependent variables concerning $H_{e0}$ and the computed overall accessibility construct (Acc).

The table shows a significant difference for PIC, EFI, and ComI variables. These variables assessed the portability of SRS, the ease of finding information, the completeness of the information, and the overall accessibility of the SRS. Their effect sizes were between Cohen's recommended indices' medium and larger thresholds. As mentioned earlier, the critical availability variable was not included in the test because SRS was available to all participants. In an ideal situation, availability, as a measure of transparency, will vary across stakeholders and software systems.

Considering SRS that accounted for the significant difference, Table 5 shows the means for PIC, ComI, EFI, and Acc were higher for UCM-SRS than for NL-SRS. The results further show that UCM-SRS participants found the information provided more portable, complete, and easy to locate. Again, at the construct level, NL-SRS reported less accessibility with an Acc mean of 12.44 compared to UCM-SRS's mean of 14.46. Based on these outcomes and the overall accessibility construct, UCM-SRS is more accessible by participants than NL-SRS, thus providing support for $H_{e1}$.

### Stakeholders' Productivity–Theme C (RO3)

The third goal *G3* of our experiment is to investigate the productivity of stakeholders when using different types of software requirements documents based on transparency. This section presents the outcome of the experiment in this context. For this analysis, we used the independent t-test. Table 9 presents a summary of the statistical results. Column one of Table 9 presents the dependent variables of usability time (UsT), usability effectiveness (UsEffec), and usability efficiency (UsEffic) for the usability task. These are followed by understandability time (UnT), understandability effectiveness (UnEffec), and understandability efficiency (UnEffic) for the understandability task.

**Table 9** Independent samples t-test for dependent variables for stakeholders' productivity

| Variable | $\gamma$ | t-value | df | p-value |
|---|---|---|---|---|
| UsT | 0.28 | 2.18 | 198.60 | 0.00 |
| UsEffec | 0.97 | −7.48 | 186.24 | 0.00 |
| UsEffic | 1.03 | −7.88 | 196.64 | 0.00 |
| UnT | 0.10 | 0.73 | 224.64 | |
| UnEffec | 0.51 | −3.89 | 194.23 | 0.00 |
| UnEffic | 0.61 | −4.70 | 214.06 | 0.00 |
| MT | 0.02 | 0.09 | 229.70 | |
| MEffec | 0.25 | 1.88 | 231.08 | 0.00 |
| MEffic | 0.50 | 3.77 | 229.45 | 0.00 |

#### $H_{a0}$–Task Completion Time

Task completion time is assessed by the variables UsT, UnT, and MT, respectively. Table 9 shows no statistically significant difference in participants' time completing the modifiability and understandability tasks. However, there was a significant difference for the UsT variable that measured the usability task's completion time. Table 9 also shows that the effect sizes for MT and UnT were less than the minimum value recommended by Cohen. On the other hand, the UsT variable produced a medium effect size. To account for the difference in usability completion time between the two groups of participants, we examined the means in Table 5. For UCM-SRS participants, the reported mean for UsT was 4.89, whereas NL-SRS participants recorded a mean of 6.42. Also, UCM-SRS participants recorded a mean of 5.31 against NL-SRS participants' mean of 5.70 for UnT

Concerning MT, UCM-SRS and NL-SRS participants recorded means of 4.70 and 4.63, respectively. Based on these results and for the usability and understandability tasks, UCM-SRS participants spent less time than NL-SRS participants. This outcome is not the case with MT, where NL-SRS participants spent less time than UCM-SRS participants. The outcome of these results tends to favour the alternative hypothesis $H_{a1}$ for the variables of UsT and UnT. However, the overall computed productivity based on time did not indicate any statistically significant difference between the two requirements documents. We note that spending less time performing a task does not necessarily mean a task has been correctly performed. Therefore, we must consider other variables, such as effectiveness and efficiency. Based on these time-based variables, we cannot conclude that UCM-SRS participants were more productive than NL-SRS participants.

#### $H_{b0}$–Task Efficiency

Concerning task efficiency, Table 9 indicates a statistically significant difference in task efficiency for the dependent variables of UsEffic, UnEffic, and MEffic. The variables produced acceptable Cohen's effect between medium to large sizes. The overall efficiency computed using all efficiency variables reported a statistically significant difference with a large Cohen's d effect size of 1. These outcomes support the alternative hypothesis $H_{b1}$.

In order to account for the difference in task efficiency between UCM-SRS participants and NL-SRS participants, we again examined the means provided in Table 5. For UsEffic, UCM-SRS participants recorded a mean of 0.61, higher than that of 0.34 recorded by NL-SRS participants.

Also, UCM-SRS participants reported a mean of 0.55 higher than NL-SRS participants, who registered a mean of 0.40 for UnEffic. For MEffic, UCM-SRS participants produced a mean of 0.45 compared to 0.32 produced by NL-SRS participants. These results show that the UCM-SRS participants reported a higher efficiency than NL-SRS participants. The NL-SRS and UCM-SRS participants reported approximately equal mean time for MT. As previously stated, the MEffic result indicates that UCM-SRS participants gained efficiency. This outcome also hints that spending less or equal time accomplishing a task does not necessarily imply the task was completed efficiently.

#### $H_{c0}$–Task Effectiveness

The variables of UEffec, UnEffec, and MEffec measure task effectiveness. Table 9 shows a significant difference in task effectiveness for all the variables under consideration. Acceptable and high Cohen's effect sizes of 1 and 0.5 were reported by UsEffec and UnEffec, respectively, whereas MEffec indicated a minimum effect size. The overall effectiveness computed using all effectiveness variables was statistically significant, with a large effect size greater than 1. These results show that UCM-SRS participants were more effective than their NL-SRS counterparts, thus supporting the alternative hypothesis $H_{c1}$. An examination of the means for the variables in Table 5 shows UCM-SRS participants recorded 3.75 for UsEffec, higher than the 3.26 by NL-SRS participants. For UnEffec, UCM-SRS participants reported a mean of 3.78, higher than the 3.28 reported by NL-SRS participants. For MEffec, UCM-SRS participants also recorded a mean of 0.4, whereas NL-SRS participants recorded a mean of 0.34.

The overall result of productivity measured in terms of time, efficiency, and effectiveness suggests that UCM-SRS participants were more productive than NL-SRS participants.

### Correlations Between Quality Factors of Transparency –Theme D (RO4)

One of our experimental goals *(G4)* is to analyse the interrelationships between overall transparency and transparency-describing constructs and between the proposed constructs and their corresponding metrics. The "less transparent" NL-SRS and the "more transparent" UCM-SRS were each used to analyse correlations and regressions, respectively.

To analyse the interrelationships between the proposed quality constructs that help achieve transparency, we first computed the sum of each standardised reflector variable (i.e., their respective metrics) for all data points.

• *Consistency* (Reliability). Table 10 summarises the results of the statistical reliability tests for the constructs of accessibility, usability, understandability, modifiability and usability. The results of the SRSs that participants used during the experiment are presented for both. The idea is to see how our proposed measures perform under both treatments. Cronbach alpha measures the internal consistency reliability of the items purported to measure a construct. There is currently no consensus on the acceptable Cronbach alpha value for reliability. Some sections in the literature favour an alpha value from 0.6 and 0.7 [45]. However, an alpha value of 0.5 is acceptable in some conditions, especially when the scale contains few items [46]. All constructs for both SRS reported acceptable alpha values in the generally recommended threshold of 0.6 and above. The variables UsT, UnT, MT, and SoI were also unloaded to achieve high reliability and proceeded with further statistical tests.

**Table 10** Test of reliability of measurement items

| | UCM-SRS | | | NL-SRS | |
|---|---|---|---|---|---|
| | Variable | $\alpha$ | Unloaded item | $\alpha$ | Unloaded item |
| Reliability | Accessibility | 0.76 | – | 0.69 | – |
| | Usability | 0.98 | UsT | 0.85 | UsT |
| | Understand-ability | 0.67 | UnT | 0.65 | UnT |
| | Modifiability | 0.67 | MT | 0.90 | MT, SoI |
| | Reusability | 0.96 | – | 0.92 | – |

As we have previously advanced, transparency is a quality requirement that is orthogonal to the functionality of software or process. We, therefore, are not required to comply with stringent statistical thresholds. Furthermore, empirical research of this nature, which requires participants from the industry, often falls short in sample size. The consequence is that more advanced statistical analysis is not always possible. This rationale is even more so as transparency is a multifaceted and complex concept that embodies different types of measurement, as our metrics have shown. Our study, therefore, could have tolerated alpha values of up to 0.5. It is reasonable to make this decision because most constructs have fewer scale items except for the usability construct.

Furthermore, the column for the unloaded items shows items that were removed due to their low-reliability contribution. Their inclusion resulted in low internal consistencies between items measuring the same construct. Despite not being loaded on the scale, these unloaded items are crucial in measuring transparency. They, therefore, remain valid transparency measures in our study. It is worth noting that in our study, the alpha value is not final but used together with other evaluation indices. The evidence from subjective tests such as face and content validity, inter-item correlations, and other tests can be relied upon to determine the validity of measures. To further buttress this point, the t-test result supports that the measures may be used to analyse the transparency of different software artefacts and systems.

• *Relevance* (Validity): Table 11 provides a summary of the correlations between the transparency constructs under investigation. Both SRSs reported strong positive correlations between all constructs. All constructs were also found to have statistically significant correlations with overall perceived transparency. We can, therefore, deduce that for both SRSs, interrelationships exist between all proposed constructs. These interrelationships are also substantial, and the constructs are relevant contributors to transparency, as regression analysis will later show in the current paper.

Therefore, for both SRS, alternative hypotheses $H_{a1}$ to $H_{j1}$ are supported and we reject the null hypotheses.

• *Model fitness* (Regression analysis): We performed a multiple regression analysis to investigate how the independent transparency constructs contribute to each other and transparency. Using the UCM-SRS, we conducted a multiple linear regression to determine the overall transparency of UCM-SRS based on accessibility, usability, understandability, reusability, and modifiability. A significant regression equation was found ($F(5,112) = 22.35$, $p < 0.000$), with an adjusted $R^2$ of 0.47. All independent constructs reported a statistically significant contribution to the overall transparency of UCM-SRS except usability, understandability, and modifiability constructs. Similarly, a significant equation was found ($F(5,112) = 9.37$, $p < 0.000$), with an adjusted $R^2$ of 0.26 for NL-SRS. All independent constructs except reusability were found not to be statistically significant contributors to overall transparency. The adjusted $R^2$ implies that 47% and 26% of the difference in the overall transparency of both requirements documents are explained by transparency quality constructs. It can be deduced from the $R^2$ produced in the case of UCM-SRS that the model performed better than NL-SRS. In both cases, all proposed quality constructs contribute to transparency, even though some contributions were not statistically significant.

Based on UCM-SRS, we first conducted a multiple linear regression to determine how accessibility, usability, understandability and modifiability influence the factor of reusability. A significant regression equation was found ($F(4,113) = 200.54$, $p < 0.000$), with an adjusted $R^2$ of 0.87. All constructs except modifiability were statistically significant contributors to reusability. On the other hand, NR-SRS reported a significant regression equation ($F(4,113) = 78.02.233$, $p < 0.000$), with an adjusted $R^2$ of 0.73.

For the factor modifiability based on UCM-SRS, a statistically significant regression equation ($F(4,113) = 4.68$, $p0.000$), with an $R^2$ of 0.11 for the constructs of accessibility, usability, reusability and understandability. Only usability and understandability were found to be statistically significant contributors to modifiability. NL-SRS also reported a statistically significant regression equation ($F(4,112) = 3.58$, $p < 0.000$), with an $R^2$ of 0.08 for the constructs of accessibility, usability, reusability and understandability. In this case, only understandability was found to be a statistically significant contributor to modifiability.

For UCM-SRS understandability, a significant regression equation was reported ($F(4, 113) = 21.53$, $p < 0.000$), with an $R^2$ of 0.41. Of all the constructs entered, only usability and reusability indicated a statistically significant contribution to understandability. The NR-SRS also reported a significant equation ($F(4, 113) = 31.81$, $p < 0.000$), with an $R^2$ of 0.51. Only usability and modifiability were statistically significant contributors to understandability.

Concerning usability, and UCM-SRS, a significant equation ($F(4, 113) = 140.21$, $p < 0.000$), with an $R^2$ of 0.83 was equally reported. All constructs entered were statistically significant contributors to usability. Similarly, NL-SRS also produced a statistically significant regression equation ($F(4, 112) = 45.49$, $p < 0.000$), with an $R^2$ of 0.60. Reusability and understandability indicated statistically significant contributions to usability.

• *Independence of constructs* (Discriminant validity): It is reasonable to ascertain that the constructs of transparency independently contribute to transparency. A multicollinearity analysis was conducted as part of the regression analysis to determine the constructs' independence. The result indicated tolerance values averaging 0.4 for both UCM-SRS and UCM-SRS. This outcome is well above the required minimum (0.1) tolerance level to ascertain independence. This result indicates that all constructs considered are independent determinants of the overall transparency of both requirements documents.

**Table 11** Correlations between transparency quality constructs

| | More transparent | | | | | Less transparent | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Usa | Reu | Und | Mod | Acc | Usa | Reu | Und | Mod | Acc |
| Usa | 1 | | | | | 1 | | | | |
| Reu | 0.960** | 1 | | | | 0.913** | 1 | | | |
| Und | 0.898** | 0.860** | 1 | | | 0.786** | 0.766** | 1 | | |
| Mod | 0.494** | 0.463** | 0.598** | 1 | | 0.328** | 0.308** | 0.459** | 1 | |
| Acc | 0.927** | 0.914** | 0.906** | 0.548** | 1 | 0.874** | 0.848** | 0.824** | 0.381** | 1 |

** Correlation is significant at the 0.01 level (2-tailed)

## Differences in Participants' Perceptions Towards the Use of Object-Oriented Requirements and their Debriefing –Theme E (RO5)

In this section, the study explores the outcome of several participants' perceptions and self-assessments concerning how they performed during the experiment *(G5)*. Again, a debriefing questionnaire can explain the raw and result data anomalies, especially after completing a controlled experiment. We look at these perceptions and the hypotheses we stated under this theme. Therefore, it is important to note that the variables investigated in this case are not part of our proposed transparency metrics that we aim to empirically validate.

Table 12 summarises the outcomes of the statistical tests for seven dependent variables concerning Theme E. The dependent variable is represented in columns one and two, the size of the effect detected, and column three, the t-value of independent samples t-test. Column four indicates degrees of freedom, and column five reports the *p*-value if it is below 0.05.

The table shows a significant difference for the variables Mot ($H_{a1}$) and UnTask ($H_{c1}$), assessing the motivation and general understanding of the tasks performed. Under this theme, we also considered the variables CorrUsTask, CorrUnTask, CorrIA, and AccIA that measure the correctness of the usability and understanding of tasks they completed and the correctness and accuracy of the impact analysis they performed. These additional variables equally reported a statistically significant difference. There was no statistically significant difference in confidence between UCM-SRS and NL-SRS participants.

The variables CorrUnTask, CorrIA, AccIA, and CorrUsTask reported acceptable small to medium effect sizes (see Table 13) based on Cohen's proposed indices. This outcome shows that the effect detected is, at least, not trivial, nor by chance. The variables Mot and recorded a medium effect size. other variables, Conf, CorrUsTask, and UnTask, were found not statistically significant and indicated small effect sizes in some cases. However, from a scientific standpoint, variables with small effect sizes cannot be ignored, given their positive statistical significance and small effect size of 0.2 (non-negligible effect).

Table 13 summarises the means and standard deviations of the seven dependent variables, assessing various perceptions for both software requirements specifications. The table is complemented with bar charts for each respective perception variable. We next consider how both requirements specifications fared in each case.

### $H_{a0}$–Motivation

The result concerning motivation shows that participants who experimented with UCM-SRS were more motivated than participants who used the NL-SRS. This outcome is further represented by Fig. 10. The UCM-SRS participants were highly and well-motivated, respectively. The mean for UCM-SRS was higher than the mean of NL-SRS, as Table 7 indicates. This result may explain why UCM-SRS participants performed better in the experimental tasks. The outcome of this result supports $H_{a1}$.

### $H_{b1}$–Confidence

For confidence, the outcome indicates that UCM-SRS participants were generally more confident than NL-SRS participants. However, UCM-SRS participants reported a higher level of confidence (80.8%) than NL-SRS participants who recorded 69.6%. This result is also represented by Fig. 11.

**Table 13** Independent samples t-test for dependent variables for participants perceptions

| Variable | $\gamma$ | *t*-value | *df* | *p*-value |
|---|---|---|---|---|
| Mot | 0.48 | −3.67 | 232.25 | 0.00 |
| Conf | 0.14 | −1.06 | 205.07 | |
| UnTask | 0.17 | −1.31 | 232.42 | |
| CorrUnTask | 0.34 | −2.61 | 205.07 | 0.00 |
| CorrUsTask | 0.21 | −1.57 | 202.58 | |
| CorrIA | 0.44 | −3.35 | 223.18 | 0.00 |
| AccIA | 0.48 | −3.38 | 230.00 | 0.00 |

**Table 12** Descriptive statistics for dependent variables for participant perceptions

| | More transparent UCM-SRS | | | | | | Less transparent NL-SRS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Variable | *N* | $\bar{x}$ | $\tilde{m}$ | *Min* | *Max* | *s* | *N* | $\bar{x}$ | $\tilde{m}$ | *Min* | *Max* | *s* |
| Motivation | 118 | 3.83 | 4.0 | 1.0 | 5.0 | 0.85 | 118 | 3.344 | 3.0 | 2.0 | 5.0 | 0.78 |
| Conf | 118 | 3.72 | 4.0 | 0.0 | 4.0 | 0.65 | 118 | 3.63 | 4.0 | 1.0 | 4.0 | 0.70 |
| UnTask | 130 | 4.56 | 5.0 | 3.0 | 5.0 | 0.62 | 118 | 4.45 | 5.0 | 2.0 | 5.0 | 0.67 |
| CorrUnTask | 118 | 3.80 | 4.0 | 1.0 | 4.0 | 0.53 | 118 | 3.57 | 4.0 | 0.0 | 4.0 | 0.79 |
| CorrUsTask | 118 | 3.80 | 4.0 | 1.0 | 4.0 | 0.54 | 118 | 3.66 | 4.0 | 0.0 | 4.0 | 0.83 |
| CorrIA | 118 | 3.61 | 4.0 | 0.0 | 4.0 | 0.70 | 118 | 3.26 | 3.0 | 0.0 | 4.0 | 0.88 |
| AccIA | 118 | 3.54 | 4.0 | 0.0 | 4.0 | 0.74 | 118 | 3.19 | 3.0 | 0.0 | 4.0 | 0.88 |

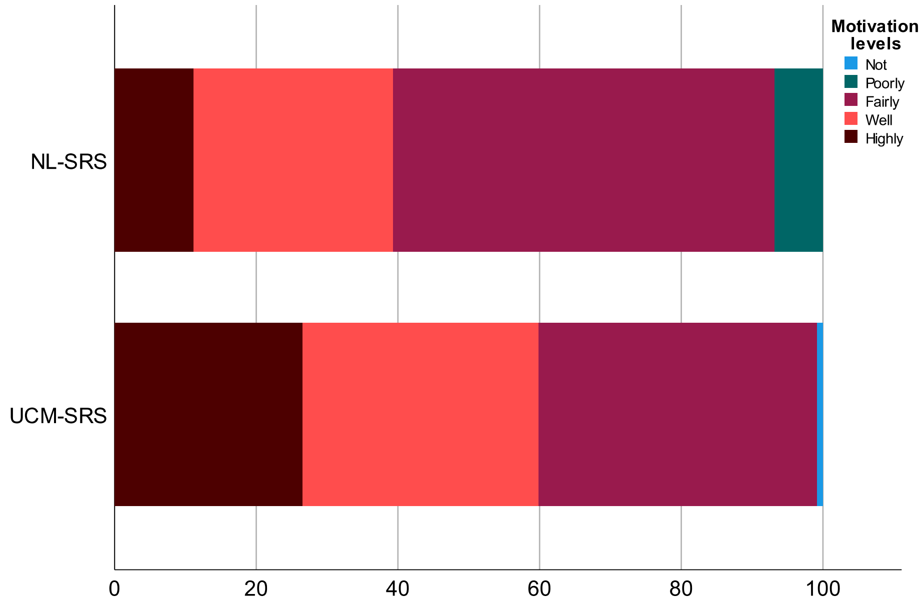



**Fig. 10** Number of experimental participants and their level of motivation using different SRSs

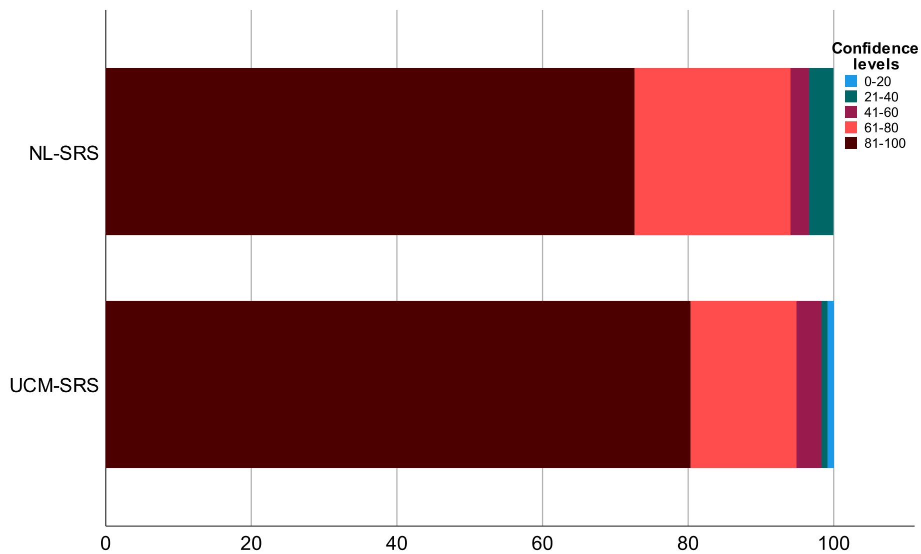



**Fig. 11** Number of experimental participants and their level of confidence in using different SRSs

NL-SRS participants reported a lower mean of 3.63 compared to UCM-SRS mean of 3.72, as indicated in Table 7. This result was found not statistically significant and, therefore, in support of $H_{b0}$.

#### $H_{c0}$–Understanding

Concerning participants' general understanding of tasks, both reported a good understanding of the tasks they were asked to perform (see Fig. 12. UCM-SRS participants again recorded a higher (61.5%) understanding of the tasks than NL-SRS participants' 54.7%. However, there was no statistically significant difference in understanding levels between both participants even as their respective negligible difference in mean (see Table 12) further indicated. This outcome supports retaining the null hypothesis $H_{c0}$.

#### $H_{d0}$–Correctness of Understandability Task

For the correctness of the understanding task performance, participants were asked to perform, participants generally believed that they had performed the task correctly. A high number (84.6%) of participants who used UCM-SRS believed they correctly performed the task. Figure 13 further indicates this. Table 12 shows that UCM-SRS participants recorded a higher mean of 3.80 compared to NL-SRS' mean of 3.57. The statistical significance of the result supports the alternative hypothesis $H_{d1}$.

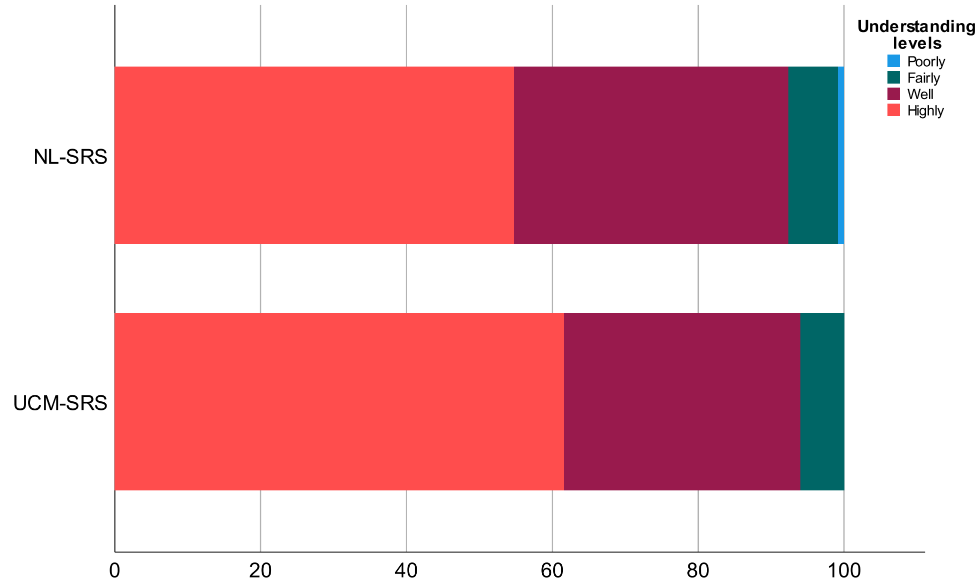



**Fig. 12** Number of participants and their level of understanding various tasks with different SRSs

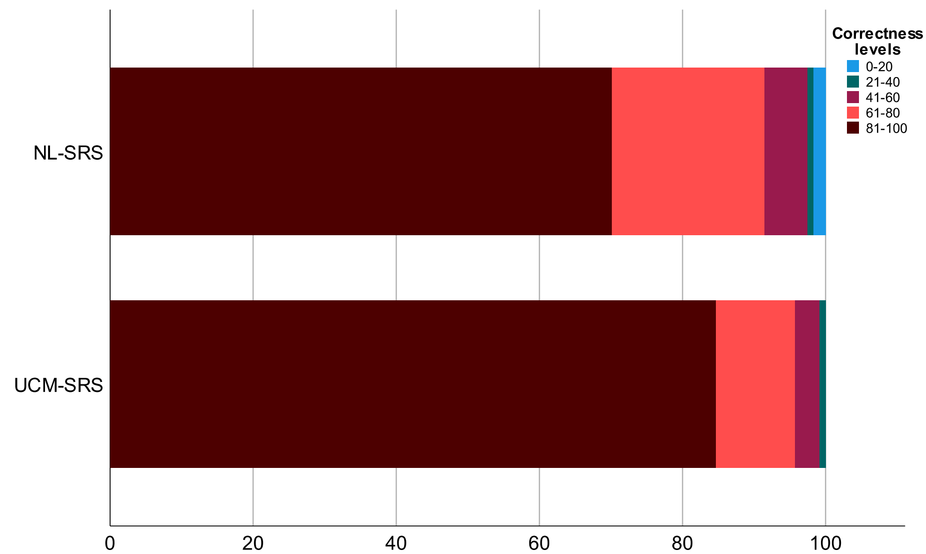



**Fig. 13** Number of experimental participants and their correctness of the understanding task performed with different SRSs

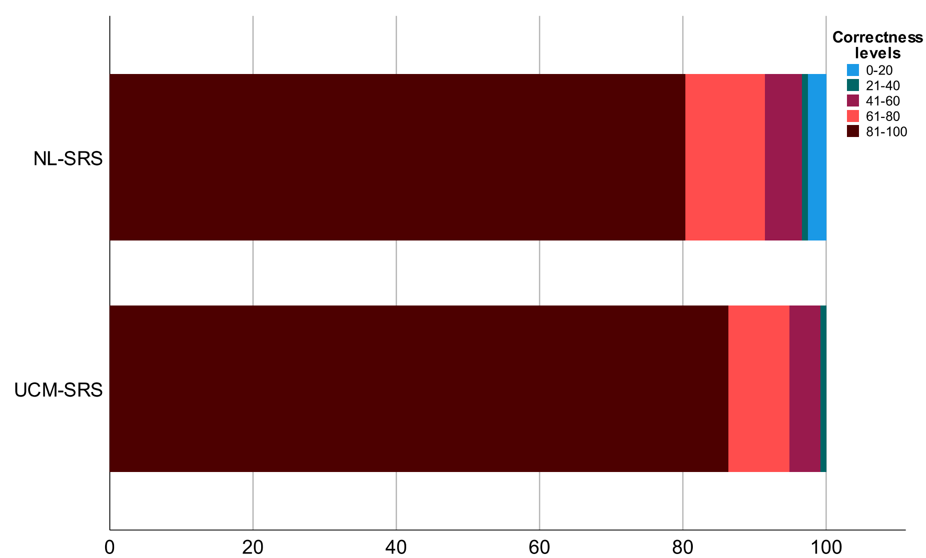



**Fig. 14** Number of participants and correctness of usability task performed with different SRSs

### $H_{e0}$–Correctness of Usability Task

Regarding the usability task's perceived correctness, UCM-SRS participants (86.2%) believed they performed it correctly compared to NL-SRS 73.0%, as Fig. 14 indicates. Table 12 shows that NL-SRS reported a mean of 3.66 lower than UCM-SRS mean of 3.80. The overall result was statistically significant, supporting $H_{e1}$.

### $H_{f0}$–Correctness of Impact Analysis

Considering the correctness of the modification task, 70.1% of UCM-SRS participants recorded higher perceived correctness than 46.2% of NL-SRS participants (see Fig. 15). Table 12 also indicates that UCM-SRS reported a higher mean (3.61) than NL-SRS mean of 3.26. A statistically significant result was obtained thus, supporting $H_{f1}$

### $H_{g0}$–Accuracy of Impact Analysis

For the accuracy of the modification, 64.1% of UCM-SRS participants believed they performed the task accurately in the upper threshold of 80–100. This outcome was more promising than NL-SRS participants' 39.3% on the same threshold (see Fig. 16). The UCM-SRS polled a mean of 3.54, whereas NL-SRS polled a mean of 3.19, respectively. A statistical significance of the result support hypotheses $H_{g1}$

### Perceived Difficulty

Concerning the difficulty faced during the experiment (see Fig. 17), UCM-SRS participants (80.2%) reported no problem with the structure or presentation of the assigned SRS. This outcome is far higher than the 19.8% recorded by NL-SRS participants for the same measure. However,

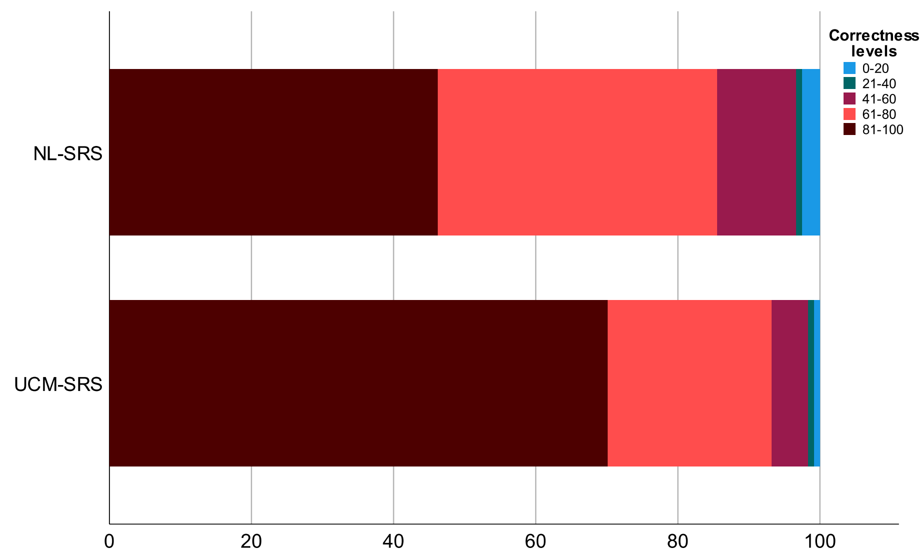



**Fig. 15** Number of participants and their correctness of impact analysis task performed using different SRSs

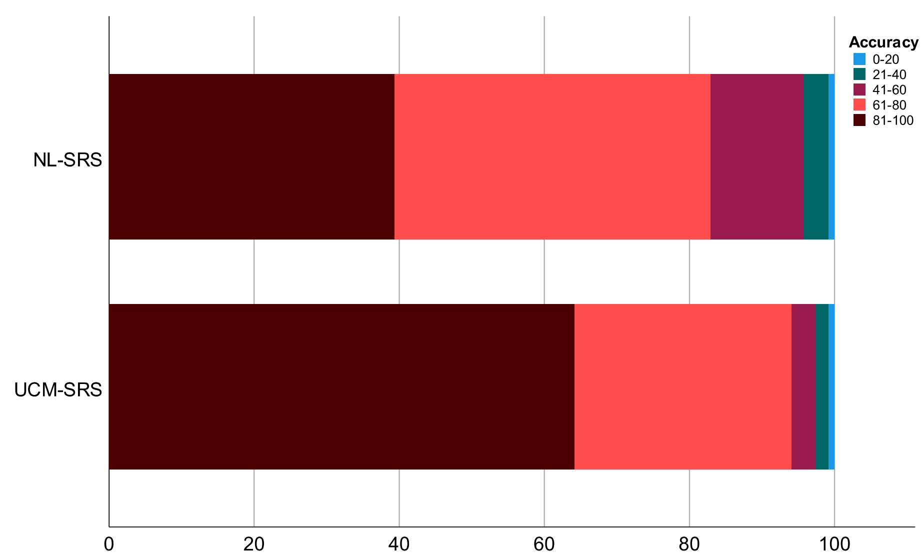



**Fig. 16** Number of participants and their accuracy of impact analysis performed using different SRSs

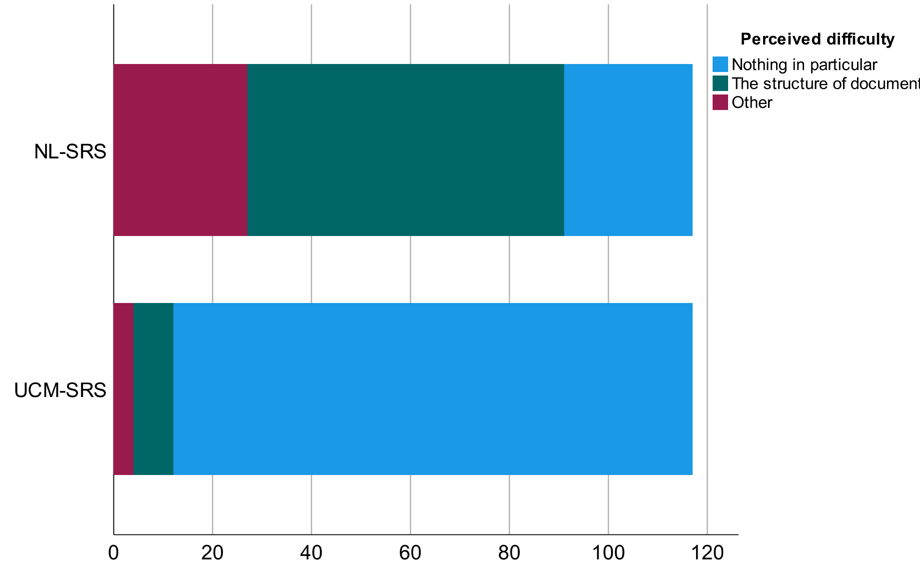



**Fig. 17** Number of participants and their perceived difficulty in performing various tasks with different SRSs

a significant number (90.00%) of NL-SRS participants reported difficulty with the requirements documents they used during the experiment. On the other hand, 11.1% UCM participants reported they had difficulty with the requirements document assigned.

Furthermore, participants (87.1% for NL-SRS and 13.0% UCM-SRS) who used either SRS reported other difficulties during the experiment. The NL-SRS appeared to record more problems than UCM-SRS participants. A common problem NL-SRS participants reported was the length of the SRS since it required them to flip through many pages during the exercise. One of the experienced participants wrote, "The document is too long in the number of pages. These days UML use case combines better with just texts so it reduces the number of pages one has to read." Another participant wrote: "I am in favour of UML-based requirements documents they are clearer, shorter and more reusable." Many other participants reechoed the same sentiments regarding the length of SRS and their preference for UML-based requirements documents. In their opinion, UML has the potential to reduce long SRS if it is adopted. Curiously, none of NL-SRS complained of time.

UCM-SRS participants did not report too many other problems with SRS. Two of the participants did report problems with the modifiability task. Their statements did not indicate their problems with the impact analysis, so we could judge that they did not understand the task. The other participants reported problems not directly related to SRS but their network connection or other issues. These results indicate that UCM-SRS participants fared better when working with UCM-SRS than NL-SRS participants.

### Approach to Performing Tasks

Concerning the approach adopted in accomplishing the tasks, both participants (NL-SRS 42.3%, UCM-SRS 57.7%) favoured jumping straight into performing the tasks without first reading SRS. Other participants (NL-SRS 54.0%, UCM-SRS 46.1%) favoured reading the documents before performing the tasks. Some participants (NL-SRS 55.6%, UCM-SRS 44.4%) favoured the reading approach, whereas thinking and performing the tasks.

## Evaluation of Metrics Based on Pre-established Requirements

In our previous work [30], we established requirements for our metrics to enhance their application and improve transparency. This section considers to what extent the metrics fulfilled each requirement.

• *Clear definition of reflective indicators*. We have clearly defined several variables for each transparency construct that assess the construct as a unit. Additionally, we conducted a reliability test that yielded an acceptable result.

• *Output logical outcome*. The metrics' application in assessing the transparency of different SRS produced results that achieved practical and statistical significance at the recommended threshold we had preset. The results obtained were logical and useful.

• *Independence*. The metrics can measure and analyse the transparency of different software artefacts irrespective of the development approach or language used in creating them.

• *Easy application*. The proposed metrics were unambiguously defined. Data collection for each metric can easily be accomplished within an organisation that intends to implement transparency as part of SEP. As the experiment further showed, data for each metric may be collected from each stakeholder of the software system under consideration. As support for transparency diagnostics, we do not envisage it to be a burden on SEP.

## Threats to Validity

In this section, we discuss various threats to the study's validity. According to [29], threats to the validity of research include construct, internal and external validity.

### Construct Validity

Construct validity is the extent to which dependent and independent variables accurately measure the concepts or phenomena they are intended to measure [29]. We have identified the following possible threats:

(a) Transparency is a complex and multifaceted concept that embodies several measures that could either be subjective or objective. Transparency is a soft concept that, if adopted, may positively impact the maintainability of software artefacts in the early stage of SDLC. It may also help achieve effective communication among stakeholders and software artefacts. In this study, we have proposed the concepts of accessibility, usability, understandability, modifiability, and reusability as transparency measures. These concepts are difficult to measure in practice. The dependent variables we have proposed are intuitive and reasonable measures. There are other ways that some of the concepts could have been measured. For instance, the measurement of reusability in terms of "potential for reuse" could have been measured in real terms of "actual use", where participants reuse the artefact. However, measuring in terms of actual use does not benefit the early stages of SEP. Similarly, an alternative measurement dimension for modifiability could have been to have participants implement the changes they found during the experiment. Realising all possible measurement dimensions for the proposed variables is often not feasible in one experiment. We have, therefore, focused on realistically possible transparency measures.

(b) Given its multifaceted and complex nature, transparency may not be measured with precision or accuracy in the traditional sense of measurement. However, there is a consensus that a software artefact or process is either "more" or "less" transparent. Furthermore, what constitutes more or less transparency needs to be agreed upon before assessing transparency. We refer to this as stakeholders' transparency requirements. In our study, the measures we have proposed target the assessment and fulfilment of transparency requirements we have proposed in another study. The targeted requirements may not be representative of stakeholders' transparency in SEP.

(c) The SRSs we used, the accompanying questionnaire, and the wording of the questions are also possible sources of threats to validity. The first objective experiment on transparency already validated the requirements documents we used for the experiment. An expert panel validated the questionnaire we used before conducting a pre-test. The pre-test and panel validation results were used to improve and create a final questionnaire. The validation and pre-test feedback helped to mitigate any further threats to validity. We also minimised the number of questions and participants' time on each task. For the online version of the experiment, controlling the time the participant spent is impossible.

(d) Another possible source of threat concerns how we assessed the correctness and completeness of the tasks performed by participants and the avoidance of bias. For instance, in the correctness and completeness of impact analysis, UCM-SRS participants had more places to identify than NL-SRS participants. The UCM-SRS presented each functional requirement more elaborately than the NL-SRS, which summarised and merged some of the functional requirements. To create a seeming balance and avoid bias, we decided not to count all correct places found by UCM-SRS participants even when they were right. This decision resulted in both participants having equal places to report. The same scenario concerned the number of correct change places found relative to the overall places each participant identified. This similar scenario was managed in the same way. To ensure participants provided answers from the expected locations, they were asked to write down the sections, page numbers, and paragraphs. In reality, it is challenging to have experimenters obey instructions in all cases.

### Internal Validity

Internal validity concerns the degree to which study conclusions can be reached regarding the cause-effect of the independent variable on the dependent variable [29]. The possible threats we identified are as follows:

(a) *Selection effect.* A selection effect usually emanates from differences in ability that may exist between experimental groups. Our study adopted a convenient sampling approach, and the objects of the study were randomly assigned to participants. Therefore, the selection should not usually be a source of threat to validity. Also, given the relatively high sample size we recorded, random assignment was seen to be more effective than if we worked with a small sample size.

(b) *Instrumentation effect.* This threat could arise owing to differences in the materials used for the experiment. A threat to validity, in this case, concerns possible differences between the two requirements specifications and their potential to cause performance differences among participants. As mentioned earlier, the more transparent SRS was created from the less transparent SRS. The authors ensured that differences were only limited to the form of presentation based on the different approaches employed in developing SRS. Both SRSs, therefore, contained the same information that described the same software system. Furthermore, the questions were carefully selected and worded to reflect the content of both documents, thereby reducing bias.

(c) *Missing data points.* To mitigate the threat of incomplete responses and random responses that are incoherent due to frustration on the participant's part, the length of the questionnaire was kept minimal. Tasks and questions were carefully selected, worded, and arranged so participants did not feel frustrated. In the online version of the experiment, all questions were made compulsory, so participants did not have to submit incomplete questionnaires. A possible threat to this is the likelihood of participants entering any response to proceed to completion. There was no way of controlling this scenario. Consequently, such responses were treated as invalid and deleted from the dataset. The authors worked vigorously to attain a considerable population size such that incomplete and invalid responses do not affect the resulting sample size.

(d) *Scale conversion.* Some of the measures we have proposed are subjective and require using the Likert (ordinal) scale for their representation and capturing. The conversion of the Likert scale into numeric values and their use in performing a parametric test such as the t-test could cause a possible threat to validity. There is no consensus on using parametric tests on Likert scale data and violating statistical rules by Likert data type. Norman [39] opine that parametric tests could be used to analyse ordinal data without arriving at a wrong conclusion. We converted the Likert data to a numeric scale and computed their sums for regression analysis; parametric tests are considered appropriate.

(e) *Random heterogeneity.* Variability of experimental subjects' abilities may threaten the validity of the experiment outcome and cause anomalies in the result [29]. Our experiment recorded many professionals (210) and fewer undergraduates (24). However, of the 24, 12 had working experience. Because of this distribution, it is reasonable to claim that the professionals and the undergraduate student professionals primarily account for the results we obtained. Expecting professionals and postgraduate students to perform better than undergraduate students is equally reasonable. In mitigating this threat, we introduced a debriefing questionnaire, which may also help explain the effect of this threat on the result and possible anomalies.

## External Validity

External validity refers to the degree to which results can be generalised to the population under consideration and other study environments [29]. The principal threats we have identified are as follows:

1. *Subject representativeness.* The sample population we worked with was dominated by industry professionals and postgraduate students, most of whom had working experience and university academics. The number (46) of undergraduates who participated in the experiment was far less than the other experienced group. From our assessment based on many professional participants, we can reasonably infer that our outcomes attained an appreciable level of external validity in this context. Stakeholders such as clients and end-users did not participate in the experiment, so our outcome may not be generalised to their population. However, a future study would investigate the benefit of transparency in the context of end-users and clients. We note that the study results were obtained based on considering SRS, a product of the RE phase of SEP. We have mainly focused on how use case modelling captures the functional requirements of the software to facilitate the achievement of transparency. We consider this achievement implicit as it only occurs due to the adoption of use case modelling. These results, therefore, are limited and may not be generalised to other artefacts produced by other SEP phases. The outcome of our study may serve as a foundation for further investigation of transparency in the other phases.
2. *Experimental material used.* The SRS employed in this study and the nature of tasks participants completed may not be representative of their complexity and size. This study is exploratory, and we are initiating another research programme to build upon it. However, we do not consider this threat a limiting factor to our results, as the results may further serve as a basis for future studies. It is worth noting that the study results align in many aspects with the results of the first empirical study on transparency in SE reported in [5]. This is even more so as both studies focused on the same setting.

## Proposed Transparency Improvement Model (RO6)

Assessing transparency should not be considered an end by itself but a means to an end to improve the degree of transparency. This motivated the inclusion of the improvement scheme as a component in the proposed transparency evaluation framework. This section proposes a transparency improvement scheme which defines a model for improving the transparency of software artefacts. It extends and builds on the other components of the framework. The improvement model covers software artefacts produced by the various SDLC phases. It integrates the proposed transparency quality factors and metrics. This section fulfils the improvement component of the framework in a way that yields a theoretical significance. The practical significance of this section is that it provides a sequence of steps and a checklist that may be used by stakeholders, especially developers generally, to improve the transparency of software artefacts, thereby tackling problems of communication and maintainability and enhancing developer productivity.

In proposing an improvement model, we adapted GQM paradigm by including four elements, namely, "Transparency Requirements," "Transparency Quality Factors," "Analysis of Result," "Improvement Feedback," and "Improvement Process." Fig. 18 summarises the overall improvement model. Next, we explain each of the steps of the improvement process.

–*Step one: Establish transparency requirements.* In this step, the organisation agrees on a set of transparency requirements that can be measured. In our previous work [7], we provided an initial set of transparency requirements in the context of SEP.

–*Step Two: Establish transparency goal.* In this step, the organisation sets a top-level transparency goal. In the context of our study, this goal will be to evaluate and improve the transparency of a software artefact such as a software requirements document or design document during the software development process. The application of transparency, in this case, will be in terms of achieving both software maintainability and effective communication as well as stakeholders' productivity. This goal should align with established transparency requirements the organisation is interested in, as provided in Step One.

–*Step Three: Establish questions.* In this phase of the evaluation process, the organisation creates a set of questions that targets the established goal in Step Two. These questions can be derived from transparency quality factors. Previously, we have proposed a set of five factors, including accessibility, understandability, usability, modifiability, and reusability. Each quality factor assesses one or more

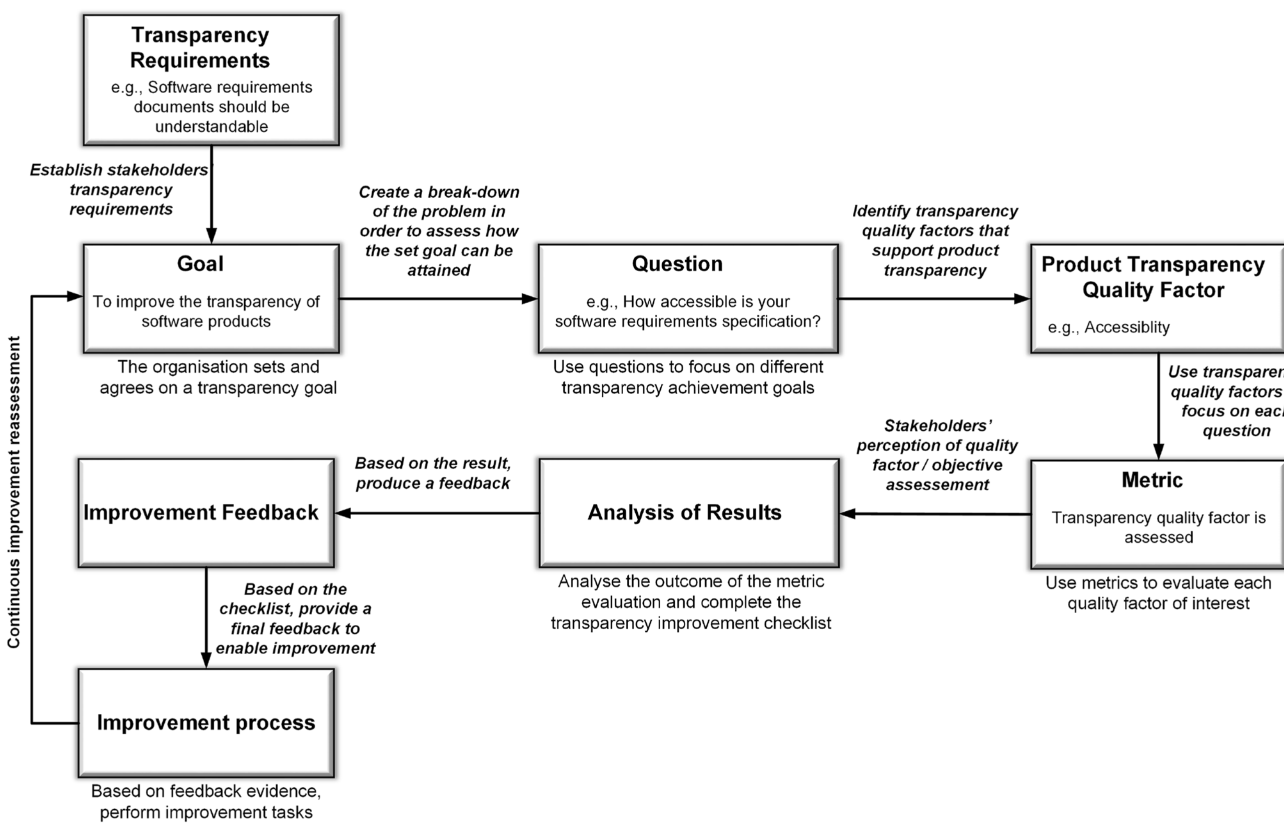



**Fig. 18** A model for improving transparency

transparency requirements provided in Step One. For instance, a question concerning understandability could be asked to determine how much stakeholders, such as requirements engineers, understand SRSs.

–*Step Four: Identify transparency quality factors.* This step complements Step Three. In this step, the organisation identifies and prioritises the transparency quality factors it may use in the evaluation and improvement process. However, the choice of quality factors should be guided by the transparency requirements the organisation aims to achieve, as provided in Step One.

–*Step Five: Identify transparency quality metrics.* Similar to Step Four, the organisation identifies and prioritises the quality metrics to apply in the evaluation process. Each quality factor has one or more corresponding quality metrics that assess a transparency requirement. Our previous work has also provided an initial set of transparency quality metrics that can be used in this regard.

–*Step Six: Create transparency evaluation tasks.* In this step, a set of transparency evaluation tasks is created for targeted stakeholders, especially requirements engineers, designers and developers. The performance of the tasks will enable the collection of some of the metrics. In this paper, we have shown how tasks could be defined and assessed using the metrics. Some metrics (e.g., readability index, conciseness) that are obtained based on the internal attributes of the software artefact are computed by the team member leading the evaluation and improvement process. Tasks that can be created include usability, understandability and modifiability tasks. The tasks should be defined as part of a questionnaire, which also provides questions that enable the collection of other metrics based on their proposed definition and measuring instrument.

–*Step Seven: Analysis of result.* In this step, the results obtained from the execution of Step Six are analysed. Based on the results, a checklist is completed. In Table 14, we have proposed a checklist that can be used in this regard.

–*Step Eight: Generate improvement feedback.* Based on the results in Step Seven, improvement feedback is generated. Improvement feedback should clearly outline the metrics that performed poorly and where more effort should be expended to yield more positive results.

–*Step Nine: Improvement process.* In this step, the feedback is used to initiate an improvement process that addresses specific problems discovered based on the analysis of the result in Step Seven. For instance, a problem an improvement process may address is making a software requirements specification more accessible by providing more information that requirements engineers consider not available or incomplete.

–*Step Ten: Perform reassessment.* This step involves conducting a reassessment of the goal following the completion of the improvement process. Embarking on reassessing the transparency goal forms a loop, as indicated in Fig. 18. The completion of re-evaluation produces a final transparency evaluation checklist.

Besides evaluating and improving transparency according to the proposed model in Fig. 18, the transparency of software artefacts can be partially and implicitly fulfilled by adopting software development approaches and techniques that encourage transparency. In this paper, we empirically showed that using object-oriented use-case modelling in developing SRSs results in "more" transparency. This outcome corroborates an earlier result provided in [5] and further lends credence to the call by Leite [47]. These outcomes further support the notion that the object-oriented paradigm remains a strong candidate in implicit transparency provisioning advocated by OMG.

## Discussion

The overall goal of this paper is to explore the usefulness of the proposed transparency measures and metrics. *RO1* aimed to apply the transparency measures and metrics to analyse transparency's impact on software maintainability, effective communication, and stakeholder productivity. Generally, experiment results showed that adopting use case modelling in developing SRS has led to participants finding it easy to work with OO-SRS, hence its ease of maintenance. This outcome considered participants' understanding and performance of impact analysis.

Concerning *RO2*, the result also showed that adopting use case modelling in developing SRS has resulted in more effective and efficient communication and high productivity *RO3*. The qualitative evidence from the analysis of the debriefing questionnaire completed by experimental participants supported the outcome in both cases. The debriefing questionnaire also assessed participants' perceptions *(RO5)* towards each SRS, and analysed results showed that OO-SRS attracted more positive perceptions than NOO-SRS. These combined results support our view that implicit transparency can be achieved by adopting object-oriented use case modelling in preparing SRS. In contrast, the explicit achievement of transparency could rely on metrics to diagnose transparency-related problems in software and software artefacts.

The research objective (*RO4*) was to analyse the interrelationships between the proposed constructs of transparency and their corresponding metrics. The proposed metrics are a combination of both subjective and objective metrics. The multifaceted and complex nature of transparency necessitates this combination. This implies transparency may not be measured with precision in the traditional sense of measurement. This outcome results in using qualifiers such as "more" or "less" transparent.

**Table 14** Transparency evaluation checklist

| Quality factor/metric | Question | Check(✓) if attained |
|---|---|---|
| *Accessibility* | | |
| Availability of information channel and information (AICI) | Is the SRS available to stakeholders? | Yes❐ No ❐ |
| Portability of the information channel (PIC) | Is the SRS portable? | Yes❐ No ❐ |
| Ease of finding information (EFI) | Is it easy for stakeholders to find information? | Yes❐ No ❐ |
| Completeness of information (CoI) | Does the SRS provide complete information? | Yes❐ No ❐ |
| *Usability* | | |
| Usability time (UsT) | Is the SRS usable based on time? | Yes❐ No ❐ |
| Usability effectiveness (UsEffec) | Is the SRS usable based on effectiveness? | Yes❐ No ❐ |
| Usability efficiency (UsEffic) | Is the SRS usable based on efficiency? | Yes❐ No ❐ |
| Perceived ease of use (PEoU) | Do stakeholders perceive the SRS to be easy to use? | Yes❐ No ❐ |
| Satisfactory usage (SU) | Are stakeholders satisfied with their use of the SRS? | Yes❐ No ❐ |
| Preference of usage (PoU) | Do stakeholders prefer to use the SRS? | Yes❐ No ❐ |
| Usefulness of information (UoI) | Do stakeholders find the information in the SRS useful? | Yes❐ No ❐ |
| Degree of information connectedness (DoIC) | Do stakeholders find the information in the SRS connected to their tasks? | Yes❐ No ❐ |
| *Understandability* | | |
| Understandability time (UnT) | Is the requirements document understandable based on time? | Yes❐ No ❐ |
| Understandability effectiveness (UnEffec) | Is the SRS understandable based on effectiveness? | Yes❐ No ❐ |
| Understandability efficiency (UnEffic) | Is the SRS understandable based on efficiency? | Yes❐ No ❐ |
| Readabiity of information (RoI) | Is the readability index of the SRS acceptable? | Yes❐ No ❐ |
| Conciseness of information (ConI) | Is the conciseness of the SRS acceptable? | Yes❐ No ❐ |
| Readability notion (ReadNotion) | Is the readability of the SRS by stakeholders acceptable? | Yes❐ No ❐ |
| Understandability notion (UnNotion) | Is the understandability of the SRS by stakeholders acceptable? | Yes❐ No ❐ |
| *Modifiability* | | |
| Modifiability time (MT) | Is the SRS modifiable based on reported time? | Yes❐ No ❐ |
| Modifiability effectiveness (MEffec) | Is the SRS modifiable based on effectiveness? | Yes❐ No ❐ |
| Modifiability efficiency (MEffic) | Is the SRS modifiable based on efficiency? | Yes❐ No ❐ |
| Modifiability completeness (MComp) | Is SRS modifiable based on number of tasks completed? | Yes❐ No ❐ |
| Size of information (SoI) | Is the SRS modifiable based on the size of information? | Yes❐ No ❐ |
| *Reusability* | | |
| Ease of reuse (EoR) | Is the SRS easy to reuse? | Yes❐ No ❐ |
| Clarity of information (CloI) | Does the SRS provide clear information? | Yes❐ No ❐ |
| Good documentation (GD) | Does the SRS provide good documentation? | Yes❐ No ❐ |
| Adherence to reusability principles (ARP) | Does the SRS adhere to good reusability principles? | Yes❐ No ❐ |
| Ease of understanding (EoU) | Is the SRS easy to understand? | Yes❐ No ❐ |

(*RO5*) sought to gauge the perceptions of experimental participants towards different SRSs based on a debriefing questionnaire that gathered data about their motivations, confidence, and understanding, among other indicators. It was found that participants were generally more positive towards OO-SRS compared to NO SRS. We note that this outcome is neither a confirmation nor a generalisation of stakeholders' perceptions. It is important to note that the perceptions investigated in this case are not part of the proposed transparency metrics. As we stated previously, these debriefing data on perceptions only enabled us to find explanations for any anomalies in data and experimental results.

Based on our research outcomes, we proposed a transparency evaluation and improvement model (*RO6*) that utilises the transparency measures and metrics we provided in this paper. The model provided a comprehensive sequence of steps that software teams can practically follow to evaluate and improve transparency during SEP. This process can be mainstreamed in a real industry software development setting.

The results of the experiment using different SRSs showed that the metrics assessed each of the constructs of transparency except in the case of time. As with all measures, not all variables usually assess an external factor, especially in a reliability test using Cronbach alpha. As we discussed earlier, the reliability test is not the only test that provides evidence for the reliability of measures. Inter-item correlations were found acceptable for all measures. Face, discriminant, and content validity are acceptable reliability measures. Overall, we found statistically significant interrelationships between the constructs of transparency. The outcome of model fitness resulting from a multiple regression analysis showed that the constructs positively and significantly contribute to other constructs as hypothesised. All constructs correlate strongly with transparency and are significant contributors to overall transparency. The experiment's outcome generally showed that the constructs and the entered metrics measured what they purported to measure. They are valid for a transparency improvement programme to consider.

The findings from our study and the experiment face several threats to validity and limitations, as we previously discussed under various themes. Our study focused on RE phase of SEP and specifically considered SRSs. Our results are, therefore, limited to this phase and may not be generalised to other phases. The experimental materials and the tasks we developed for experimental participants may not be representative in size and complexity. These may require improvement in future replication. It is worth noting that the requirements documents we adopted from a previous related experiment were not modified for use in our study. In addition to complying with use case modelling standards, future document improvement could consider their compliance with other SRS qualities, such as consistency and traceability. Participants would have been more guided in impact analysis if traceability had been incorporated into the documentation. Our study was also limited to a high proportion of ICT professionals and excluded end-users and clients. The results may not be generalised to this population. The current study is exploratory, and we hope to build on it in our future study

*Implications and Challenges of Transparency*

A significant implication of the achievement of transparency in SE and IS is fostering stakeholders' confidence and trust in the development and deployment of software. Our approach to engineering or achieving transparency during SEP is predicated on a requirement engineering methodology. In [7, 10, 25], we demonstrated that stakeholders' transparency requirements must first be established to achieve transparency. When these requirements are evaluated and improved through transparency measures and metrics, transparency can be said to have been achieved. Since transparency requirements are elicited from stakeholders, their fulfilment potentially enhances trust, confidence, and the eventual adoption of the deployed software.

However, transparency implementation poses several challenges, which we have provided in our previous work [10]. The multifaceted and complex nature of transparency presents an inherent challenge in terms of the varying ambiguities of transparency. This complexity underscores the need to implement transparency without antagonising privacy, security, and confidentiality. We consider privacy, security, and confidentiality as antagonistic properties of transparency. We maintain that transparency requirements that seek to preserve antagonistic transparency properties must be elicited and implemented for a comprehensive achievement of transparency. The management of "negative" transparency properties has been reported in [48, 49].

Cappelli et al. [48] opine that dealing with conflicting transparency requirements is challenging, even as their approach further suggests. However, they proposed a method to address antagonistic requirements based on indirect knowledge elicitation by Gonzalez-Baixauli et al. [50, 51], who used the personal construct theory (PCT) [52] to identify interactions among transparency requirements. We briefly highlight and illustrate how their proposal fits into a comprehensive transparency evaluation and improvement scheme. Claudia et al. indirect elicitation strategy systematically determines relationships among a given set of transparency requirements (i.e., transparency quality issues), including privacy and security. The strategy outlines three essential steps: (1) Identifying the constructs of transparency (e.g., accessibility, privacy), (2) analysing relationships among constructs based on PCT, (3) defining the contribution or influence (positive or negative) of constructs to each other based on a defined set of rules. These rules should dictate how the transparency quality factor relates to another. Completing these steps results in new interactions between all constructs and their contribution to each other. The full implementation of the strategy is provided in [48].

Another challenge with transparency implementation is the cost, which can potentially increase the total cost of software development. Because transparency is considered orthogonal to software, its incorporation may not be attractive to some project managers despite its importance. Other challenges include the non-formalisation of the concept in a manner our study is motivated to provide to enhance understanding and measurement of transparency. In future, our results could enable the development of tools that can diagnose, track, and visualise transparency problems during SEP and post-deployment of the software.

*Limitations of study.* Software projects usually involve several stakeholders. The evaluation of transparency factors and metrics was performed by ICT professionals knowledgeable in SRSs. Stakeholders such as non-ICT professionals did not participate in the evaluation. Therefore, study results are limited to ICT professionals and may not be generalised to other stakeholders. We have embarked on a research programme to evaluate the transparency of SRSs using non-ICT professionals. As the authors envisaged, some evaluation factors and metrics may not apply to non-ICT professionals since they need prior knowledge. Furthermore, study results may not be generalised across all SDLC phases and their corresponding artefacts because we considered only SRSs produced during the RE phase. Other limitations in the form of threats to the validity of results have been previously addressed.

*Summary of Contributions*

The current paper builds on the outcomes of our previous research effort since 2022. In [10], we comprehensively examined transparency's existing state of affairs as an NFR quality, conceptualisations, and operationalisations in SE and IS. We showed that studies in both disciplines have not provided a straightforward, theory-driven approach to evaluating and improving transparency, especially in SE. Additionally, the transparency measures and metrics reported are poorly defined, while most do not apply to SEP. Studies in SE are few, as our previous work further showed and did not provide the necessary formalisms to support transparency. This finding motivated us to propose in [25] a comprehensive conceptual framework for evaluating and improving transparency in SE. The conceptual framework kick-started our journey to exploring and providing the necessary formalisms that may help transparency understanding and its measurement. Given our RE approach to realising transparency, in [7], we identified software stakeholders. We also elicited an initial set of stakeholders' transparency requirements that can be evaluated and improved during SEP. This outcome fulfilled the stakeholders and transparency requirements component of our framework.

Our previous work [10] also showed that existing studies have not empirically shown the correlations between transparency constructs and metrics. Constructs and metrics are required to assess the fulfilment of transparency requirements. It is important to note that SE studies have not provided suitable metrics. Inspired by existing SE studies and in this paper, we proposed several transparency factors and metrics [30] to evaluate the transparency requirements we elicited. Most importantly, this paper improves upon existing SE and IS transparency studies by providing empirical evidence of the correlations between transparency constructs and metrics. Additionally, measures and metrics are only helpful if they measure what they purported to measure via empirical studies. This is an essential step towards theory-building, which previous studies on transparency have not considered, as our previous work showed. We further demonstrated the usefulness of our metrics in contexts not previously considered, including software requirements, maintainability, and productivity. This paper addresses the overall conceptual framework's metrics and quality factor components and provides the improvement scheme component of the framework. The improvement scheme harnesses the factors and metrics to develop a practical transparency evaluation and improvement programme for SEP.

## Conclusion

Our study considered transparency as the degree of openness of information about software products and processes. Because transparency may not be measured directly, we used an integrated approach to develop and validate constructs and metrics for measuring transparency. To measure transparency, we identified vital transparency constructs of accessibility, usability, understandability, modifiability, and reusability upon which the transparency metrics were derived. Thus, we provided a set of metrics for each construct that assesses the targeted construct. An experiment was necessary to empirically validate and demonstrate the usefulness of transparency constructs and metrics. It also enabled us to evaluate the transparency of different kinds of SRS regarding their maintainability, communication effectiveness, and stakeholders' productivity. Positive correlations were found among constructs and metrics, and the metrics measured what they purported to measure.

The metrics measurement outcomes showed that the development technique influenced SRS' maintainability and communication effectiveness. The natural language text-based technique was less effective, and its SRS was less maintainable than use case modelling. The experiment confirmed and quantified the positive effects of using the use-case modelling approach in developing SRS. This outcome aligns with the call for transparency through the object-oriented paradigm, supporting the benefits of object orientation promoted by OMG. The time, effectiveness, and efficiency-based metrics generally showed that use case modelling-based SRS offer more productivity than natural language-based SRS.

The study opens several opportunities for further research, including an external experiment replication. There is a need to investigate how other phases of SEP may benefit from transparency and the generalisation of results in those contexts. The outcome of this study is not an end but a means to an end, which is to create a transparency improvement programme. Work is ongoing to further develop the transparency improvement scheme and apply it in a real-world scenario. An improvement scheme fulfils part of our transparency-improving framework. An improvement scheme would guide stakeholders in SEP, especially requirements engineers and business analysts, in diagnosing transparency problems. It would also generally improve the transparency of the software artefacts they create.

## Appendix A: Online Version of SRSs and Questionnaire

The objects or materials used for the experiment are provided as follows. *UCM-SRS* https://drive.google.com/drive/folders/1jCR7OEIvZBb_CAoBDWbR0g8LYQ31l8rl?usp=drive_link

*NL-SRS* https://drive.google.com/drive/folders/1RdL9iORHIi45OGCkmSiqgvqcDerDsMcE?usp=drive_link

*Questionnaire*https://forms.gle/ry8jbGTfzupEC8b48

## Appendix B: Survey Responses

The raw questionnaire responses are available at https://doi.org/10.21227/8gj2-b420.

**Acknowledgements** This research is part of the first author's PhD study, which is supported by the Data Science and Computing research unit of the Department of Computer Science, North-West University, and the author's primary institution, the Federal University of Lafia, Nigeria, with support from the Tertiary Education Trust Fund.

**Funding** Open access funding provided by North-West University.

**Data Availability** The authors declare that the data supporting this study's findings are available within the paper, and the raw questionnaire responses are provided under Appendix B.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethics Approval and Consent to Participate** This study was performed in line with the principles of the Declaration of Helsinki. Approval was granted by the Faculty of Natural and Agricultural Sciences Ethics Committee (FNAS-REC), (Date: 29/09/2022/No. NWU-01396-22A).



## References

1. Leite JCSdP, Cappelli C. Software Transparency. Bus. Inf. Syst. Eng. 2010;https://doi.org/10.1007/s12599-010-0102-z.
2. Hosseini M, Shahri A, Phalp K, Ali R. Four reference models for transparency requirements in information systems. Requirements Engineering. 20https://doi.org/10.1007/s00766-017-0265-y.
3. Spagnuelo D, Lenzini G. Transparent Medical Data Systems. J Med Syst. 2017;41(1):1–12. https://doi.org/10.1007/s10916-016-0653-8.
4. Tu YC, Thomborson C, Tempero E. Illusions and perceptions of transparency in software engineering. In: 18th Asia-Pacific Software Engineering Conference. Ho Chi Minh city, Vietnam: IEEE; 2011. p. 365–372.
5. Tu YC, Tempero E, Thomborson C. An experiment on the impact of transparency on the effectiveness of requirements documents. Empirical Software Engineering. 2016;21:1035–1066https://doi.org/10.1007/s10664-015-9374-8.
6. Meunier P. Software transparency and purity. Commun ACM. 2008;51(2):104–104.
7. Ofem P, Isong B, Lugayizi F. Stakeholders' Transparency Requirements in the software engineering process. In: IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society. IEEE; 2022. p. 1–6.
8. Blackwelder B, Coleman K, Colunga-Santoyo S, Harrison JS, Wozniak D. The volkswagen scandal [Case study]. University of Richmond Scholarship Repository. 2016;.
9. Johnston P, Harris R. The Boeing 737 MAX saga: lessons for software organizations. Software Quality Professional. 2019;21(3):4–12.
10. Ofem P, Isong B, Lugayizi F. On the Concept of Transparency: A Systematic Literature Review. IEEE Access. 2022;10:89887–914.
11. Tu YC. Transparency in Software Engineering [Thesis]. New Zealand: University of Auckland; 2014.
12. Fenton N, Bieman JM. Software Metrics: A Rigorous and Practical Approach. 3rd ed. New York, USA: CRC Press; 2014.
13. Gilb T, Finzi S. Principles of software engineering management, vol. 11. MA: Addison-wesley Reading; 1988.
14. Van Solingen R, Basili V, Caldiera G, Rombach HD. Goal question metric (gqm) approach. Encyclopedia of software engineering. 2002;.
15. Michener G, Bersch K. Identifying transparency. Information Polity. 20https://doi.org/10.3233/IP-130299.
16. Cappeli C, Leite JCSdP, Oliveira AdPA. Exploring Business Process Transparency Concepts. In: 15th IEEE International Requirements Engineering Conference (RE 2007); 2007. p. 389–390.
17. Leite JCSdP, Cappeli C. Exploring i* Characteristics that Support Software Transparency. In: CEUR Workshop Proceedings. Recife, Brazil; 2008. p. 51–54.
18. Cappeli C. An approach for business processes transparency using aspects [Thesis]. school=Dissertation, Departamento de Informática, PUC-Rio, Ago (in Portuguese); 2009.
19. Hosseini M, Shahri A, Phalp K, Ali R, Towards engineering transparency as a requirement in socio-technical systems. In,. IEEE 23rd International Requirements Engineering Conference (RE). Ottawa, Canada. 2015;2015:268–73.
20. Hosseini M, Shahri A, Phalp K, Ali R. Foundations for transparency requirements engineering. In: Proceedings of the 22nd International Working Conference on Requirements Engineering: Foundation for Software Quality. vol. 9619. Gothenburg, Sweden: Springer-Verlag; 2016. p. 225–231.
21. Hosseini M, Shahri A, Phalp K, Ali R. A modelling language for transparency requirements in business information systems. In: International Conference on Advanced Information Systems Engineering. Ljubljana, Slovenia: Springer; 2016. p. 239–254.
22. Spagnuelo D, Bartolini C, Lenzini G. Metrics for Transparency. In: Data Privacy Management and Security Assurance. Heraklion, Crete, Greece: Springer International Publishing; 2016. p. 3–18.
23. Spagnuelo D, Bartolini C, Lenzini G. Modelling Metrics for Transparency in Medical Systems. In: Trust, Privacy and Security

in Digital Business. Lyon, France: Springer International Publishing; 2017. p. 81–95.
24. Tu YC, Tempero E, Thomborson C. In: Evaluating presentation of requirements documents: Results of an experiment. Springer; 2014. p. 120–134.
25. Isong B, Ofem P, Lugayizi F. Towards a Framework for Improving Transparency in the Software Engineering Process. In: 2022 12th International Conference on Software Technology and Engineering (ICSTE); 2022. p. 19–28.
26. Gregor S. The nature of theory in information systems. MIS quarterly. 2006;p. 611–642.
27. Sjøberg DIK, Dybå T, Anda BCD, Hannay JE. In: Building Theories in Software Engineering. London: Springer London; 2008. p. 312–336. Available from: https://doi.org/10.1007/978-1-84800-044-5_12.
28. Creswell JW, Creswell JD. Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications; 2017.
29. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A. Experimentation in software engineering. Springer, Heidelberg New York Dordrecht London: Springer Science & Business Media; 2012.
30. Ofem P, Isong B, Lugayizi F. Defining Metrics for Evaluating Transparency. In: 2024 14th International Conference on Software Technology and Engineering (ICSTE); 2024. Accepted.
31. ISO/IEC. Systems and Software-Systems and Software Quality Requirements and Evaluation(Square)-System and Software Quality Models (25010. ISO/IEC; 2011.
32. Briand LC, Bunse C, Daly JW. A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. IEEE Trans Software Eng. 2001;27(6):513–30. https://doi.org/10.1109/32.926174.
33. Genero M, Poels G, Piattini M. Defining and validating metrics for assessing the understandability of entity-relationship diagrams. Data & Knowledge Engineering. 2008;64(3):534–57.
34. Dwivedi YK, Wastell D, Laumer S, Henriksen HZ, Myers MD, Bunker D, et al. Research on information systems failures and successes: Status update and future directions. Inf Syst Front. 2015;17:143–57.
35. Roger SP, Bruce RM. Software engineering: a practitioner's approach. McGraw-Hill Education; 2015.
36. Coughlan J, Macredie RD. Effective Communication in Requirements Elicitation:A Comparison of Methodologies [Journal Article]. Requirements Eng. 2002;7:47–60.
37. Falessi D, Juristo N, Wohlin C, Turhan B, Münch J, Jedlitschka A, et al. Empirical software engineering experts on the use of students and professionals in experiments. Empir Softw Eng. 2018;23:452–89.
38. Kitchenham BA, Pfleeger SL. In: Shull F, Singer J, Sjøberg DIK, editors. Personal Opinion Surveys. London: Springer, London; 2008. p. 63–92.
39. Norman G. Likert scales, levels of measurement and the "laws'' of statistics. Adv in Health Sci Educ. 2010;15(5):625.
40. Cohen J. Statistical power analysis for the behavioral sciences. Academic press; 2013.
41. Cumming G. Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis. Routledge; 2013.
42. Fritz CO, Morris PE, Richler JJ. Effect size estimates: current use, calculations, and interpretation. J Exp Psychol Gen. 2012;141(1):2.
43. Ofem P, Isong B, Lugayizi F.: Software Requirements Documentation Experimentation. IEEE Dataport. Available from: https://dx.doi.org/10.21227/8gj2-b420.
44. Tabachnick BG, Fidell LS, Ullman JB. Using multivariate statistics. vol. 6. pearson Boston, MA; 2013.
45. Nunnally JC. Psychometric theory 3E. Tata McGraw-Hill Education; 1994.
46. Taber KS. The use of Cronbach's alpha when developing and reporting research instruments in science education. Res Sci Educ. 2018;48(6):1273–96.
47. Leite JCSdP. The Prevalence of code Over Models: Turning it Around with Transparency [Journal Article]. 2018 IEEE 8th International Model-Driven Requirements Engineering Workshop (MoDRE). 2018 August;p. 56–57https://doi.org/10.1109/MoDRE.2018.00013.
48. Cappelli C, Cunha H, Gonzalez-Baixauli B, do Prado Leite JCS. Transparency versus security: early analysis of antagonistic requirements. In: Proceedings of the 2010 ACM symposium on applied computing. Sierre, Switzerland: ACM; 2010. p. 298–305.
49. Serrano M, Leite JCSP, Capturing transparency-related requirements patterns through argumentation. In,. First International Workshop On Requirements Patterns. Trento, Italy. 2011;2011:32–41.
50. Gonzalez-Baixauli B, Laguna MA, Leite JCSdP. Aplicación de la Teoría de Constructos Personales a la Elicitación de Requisitos. IEEE LATIN AMERICA TRANSACTIONS. 2005;3(1).
51. Gonzalez-Baixauli B, do Prado Leite JCS, Laguna MA. Eliciting non-functional requirements interactions using the personal construct theory. In: 14th IEEE International Requirements Engineering Conference (RE'06). IEEE; 2006. p. 347–348.
52. Butt T. George Kelly: The psychology of personal constructs. Bloomsbury Publishing; 2008.