Test cases as a measurement instrument in experimentation

O. DIESTE, Universidad Politécnica de Madrid, Spain
F. UYAGUARI, ETAPA Telecommunications Company, Ecuador
S. VEGAS, Universidad Politécnica de Madrid, Spain
N. JURISTO, Universidad Politécnica de Madrid, Spain

Background: Test suites are frequently used to quantify relevant software attributes, such as quality or productivity. **Problem:** We have detected that the same response variable, measured using different test suites, yields different experiment results. **Aims:** Assess to which extent differences in test case construction influence measurement accuracy and experimental outcomes. **Method:** Two industry experiments have been measured using two different test suites, one generated using an *ad-hoc* method and another using *equivalence partitioning*. The accuracy of the measures has been studied using standard procedures, such as ISO 5725, Bland-Altman and Interclass Correlation Coefficients. **Results:** There are differences in the values of the response variables up to ±60%, depending on the test suite (*ad-hoc* vs. *equivalence partitioning*) used. **Conclusions:** The disclosure of datasets and analysis code is insufficient to ensure the reproducibility of SE experiments. Experimenters should disclose all experimental materials needed to perform independent measurement and re-analysis.

CCS Concepts: • **General and reference** → **Measurement**; **Experimentation**.

Additional Key Words and Phrases: Test suite, measuring instrument, accuracy, agreement

ACM Reference Format:
O. Dieste, F. Uyaguari, S. Vegas, and N. Juristo. 2021. Test cases as a measurement instrument in experimentation. *ACM Trans. Softw. Eng. Methodol.* 99, 9, Article 999 (September 2021), **??** pages. https://doi.org/10.1145/9999999.9999999

1 INTRODUCTION

Test-driven development (TDD) research frequently uses the external quality (QLTY) and productivity (PROD) response variables. QLTY is typically measured as the ”amount” of correct functionality delivered by the developers’ code. PROD has a similar definition but is related to a time frame (e.g., the duration of an experimental session). "Functionality" is an abstract concept, not directly observable. In TDD research, test cases are often used as surrogates of functionality.

We have conducted a family of experiments on TDD, as part of the Empirical Software Engineering Industry Lab (ESEIL) project. We used different test suites, as recommended by Shadish et al. [**?** , 81-82], to measure QLTY and PROD values, thus preventing the mono-method threat to validity. We anticipated some variability among measures, but differences were much larger than we expected.

Authors’ addresses: O. Dieste, odieste@fi.upm.es, Universidad Politécnica de Madrid, Boadilla del Monte, 28660, Spain; F. Uyaguari, fuyaguar@etapa.net.ec, ETAPA Telecommunications Company, Cuenca, 10204, Ecuador; S. Vegas, svegas@fi.upm.es, Universidad Politécnica de Madrid, Boadilla del Monte, 28660, Spain; N. Juristo, natalia@fi.upm.es, Universidad Politécnica de Madrid, Boadilla del Monte, 28660, Spain.

1049-331X/2021/9-ART999 $15.00
https://doi.org/10.1145/9999999.9999999

The experimental analyses yield different results, sometimes reversing the effect of the independent variables, depending on the test suite used [**?**].

This paper aims at evaluating to what extent test cases influence the measurement of response variables in TDD experiments. Although the discussion is specifically framed in TDD research, measurement using test cases is frequent in software engineering (SE) research, e.g., [**? ? ?**]; other SE areas can thus benefit from our findings.

The contributions of this paper are:

- We show that the results of TDD experiments vary depending on the test suites used as measuring instruments. We have assessed this fact in our experiments, but we are certain that the same harmful effect happens in other TDD experiments.
- We introduce specialized terminology and methods, borrowed from Metrology, the Natural and the Social sciences, to study the accuracy of the test suites when used as measuring instruments.
- We assess that the measures made using different test suites yield very different results. The same piece of code may exhibit ±60% score differences depending on the test suite used.
- The publication of datasets and analysis code, as currently required by some publishers, may be sufficient for ensuring reproducibility [**? ?**], but insufficient to evaluate the influence of the measuring instruments. We propose some recommendations to improve the situation: (1) experiments should disclose all experimental materials needed to perform independent measurements, and (2) the practice of re-analysis [**? ?**] should be adopted in SE to improve experimental research.

This paper has been written using reproducible research principles. The manuscript LaTeX code is available at https://github.com/GRISE-UPM/TestSuitesMeasurement (including data files, Java and R code). Analyses have been carried out using R [**?**] version 4.0.2 (2020-06-22), and the packages *lme4* [**?**], *xtable* [**?**], *texreg* [**?**], *broom* [**?**], *MethComp* [**?**], *Hmisc* [**?**], *emmeans* [**?**], and *xlsx* [**?**].

The paper is structured as follows: Section **??** describe the research problem. Section **??** sets out the research goals. In Section **??** we introduce the terminology and methods used in Metrology and other sciences for the comparison of measuring instruments. The actual comparison is performed in Section **??**. We discuss the implication of our findings in Section **??** and, finally, provide some recommendations in Section **??**.

2 PROBLEM DESCRIPTION

In this paper, we are using two replications conducted in the industry as running examples. These replications will be referred to as **PT** and **EC** to maintain companies' anonymity. PT and EC replications have been described in detail in [**?**] and [**?**], respectively.

2.1 Experimental replications

PT and EC replications explore two programming strategies: TDD and incremental test-last development (ITLD). TDD requires writing tests before production code, whereas ITLD proceeds inversely. The experimental design is described in Table **??**, where the programming strategy is a within-subjects factor.

The programming strategies have been applied on two greenfield experimental tasks, namely Mars Rover API (MR[1]) [**?**] and Bowling Score Keeper (BSK[2]) [**?**]. MR and BSK are crossed across programming strategies to avoid confounding. This type of design is frequent in SE when participants need to receive specific training, and a few experimental subjects are available.

The assignment of subjects to groups was performed randomly. 17 and 20 experimental subjects participated in PT and EC, respectively. They were programmers with different degrees of experience, employed in the corresponding companies. PT programmers used Java and jUnit, whereas EC ones used C++ and Boost Test.

Table 1. Experimental design

		Treatment	
		(Session 1)	(Session 2)
		ITLD	TDD
Group	Group $MR \rightarrow BSK$	MR	BSK
	Group $BSK \rightarrow MR$	BSK	MR

2.2 Response variables and measurement procedure

We studied external quality (QLTY) and productivity (PROD) response variables. QLTY represents the software quality measured in terms of compliance with the software requirements. Similarly, PROD represents the amount of functionality delivered by programmers. These response variables have been frequently explored in TDD research, e.g., [**? ? ? ?**], as well as in our previous research, e.g., [**? ? ?**].

QLTY and PROD were measured using specifically designed *unit tests suites*[3]. One test suite was reused from previous experiments [**? ?**]. It was generated using an *ad-hoc* (AH) strategy and coded in jUnit. Ad-hoc means that a formal procedure to create the test cases has not been used; the authors of the test suites (MR and BSK) applied their best judgment to derive a set of test cases from the functional requirements.

To avoid the mono-method threat to validity [**?** , 81-82], we designed new test suites (for MR and BSK) using the *equivalence partitioning* (EP) technique, and coded them in jUnit. We applied equivalence partitioning according to [**?** , Chapter 4]; details can be found in [**?**]. Later, both test suites were ported to Boost Test.

The test suites give a percentage (0%-100%) as a result. For instance, in the case of QLTY, this percentage represents the degree to which the code complies with the software requirements: A 0% value means that the code does not satisfy any requirement; a 100% value means that the code satisfies all requirements.

2.3 Characteristics of the MR and BSK test suites

The AH and EP test suites are composed of a varying number of test classes/methods/assertions, as indicated in Table **??**. BSK's requirements are well defined, hence the AH and EP test suites exhibit strong similarities: They have the same number of test classes (which are roughly equivalent to functional requirements), and a comparable number of assertions. The EP technique provides a perfect correspondence between test methods and assertions.

[1]MR and BSK task specifications are included in https://github.com/GRISE-UPM/TestSuitesMeasurement/tree/master/experimental_tasks.

[2]See footnote **??**.

[3]The test suites are provided as a single Eclipse workspace containing four projects. They are available as one Eclipse workspace at https://github.com/GRISE-UPM/TestSuitesMeasurement/tree/master/test_suites.

Table 2. Characteristics of the AH and EP test suites

			Test suite	
			AH	EP
Task	MR	Test classes	11	9
		Test methods	52	32
		Assertions	89	32
	BSK	Test classes	13	13
		Test methods	48	72
		Assertions	55	72

Table 3. Coverage of the AH and EP test suites

			Test suite	
			AH	EP
Task	MR	Statement coverage	100%	100%
		Branch coverage	88.1%	94.4%
	BSK	Statement coverage	100%	96.5%
		Branch coverage	100%	94.4%

MR is defined at a high level and misses a stable specification. Consequently, the AH and EP test suites diverge considerably. However, *divergence* does not imply *measurement differences*. The same code can be measured using different test suites and obtain the same measurement results. For instance, the function `int sum(int a, int b){ return a + b; }` gets a 100% QLTY with both test suites below:

Listing 1. Equivalent tests suites, from the measurement viewpoint

```
public class Suite1{
   @Test
   public void testOnePlusOneGivesTwo() {
        assertEquals(2, sum(1, 1)); }
}

public class Suite2{
   @Test
   public void testThreePlusTwoGivesFive() {
        assertEquals(500, sum(300, 200)); }

   @Test
   public void testThreePlusMinusTwoGivesOne() {
        assertEquals(300, sum(400, -100)); }
}
```

From a testing perspective, the AH and EP test suites are largely equivalent. When we exercise the test suites on correct implementations of MR and BSK, the coverage is almost identical, as shown in Table **??**. Statement coverage is virtually 100% in all cases. Branch coverage is somewhat smaller but exceeds 90% (except the AH test suite when applied to the MR task, which has an 88% branch coverage).

Given the coverage values, it is reasonable to assume that both test suites give the same or strongly correlated results. Simple correlation analysis can be used to assess convergent validity

[? , p.67]. Table ?? shows the results. All correlations are large ($r > 0.5$, according to Cohen [?]), with the only exception of QLTY at PT ($r = 0.41$, quite close to 0.5), and statistically significant (which is remarkable given the limited sample sizes). At the outset, AH and EP test suites seem to provide similar measures; in the case of EC, to a large extent.

Table 4. Correlations between *ad-hoc* and *equivalence partitioning* measures for PT and EC

Experiment	Variable	r	$p-value$
PT	QLTY	0.41	0.02
	PROD	0.67	<0.001
EC	QLTY	0.72	<0.001
	PROD	0.82	<0.001

2.4 Problem detection

PT and EC experiments were analyzed as recommended by Vegas et al. [?], i.e., using a mixed model where *Treatment*, *Task* and *Group* are fixed factors, and *Subject* is a random factor embedded within each *Group*. The analysis model using the *lme4* package [?] is:

$$Y \sim Treatment + Task + Group + (1|Subject) \quad (1)$$

where Y can be QLTY or PROD. We will restrict the discussion to the QLTY response variable, but the comments below match PROD's behavior as well. Tables ?? and ?? show the analysis results for QLTY at PT and EC, respectively. The numbers between parentheses represent the *standard error* of the *fixed effect* located to its left. The degree of statistical significance is reported using asterisks. The differences between the AH and EP test suites are substantial:

- The *Task* effect **reverses depending on the test case definition strategy**. For AH, the effect is negative whereas, for EP, the effect is positive. The changes are dramatic in PT's QLTY (from -24.47 to 20.99 percentage points). Differences are statistically significant both for PT and EC experiments.
- The *Group* effect is **positive for AH, and void for EP**. The analysis does not give statistically significant results in this case.
- Fixed effects are **larger for AH than EP** regardless of the variable (the *Task* at EC is the exception). Standard deviations are also **larger for AH than EP** in all cases.

There is just one coincidence between the AH and EP measurements:

- The *Treatment* (ITLD vs. TDD) **is largely unaffected**. The AH and EP test suites give different values, but the sign and the statistical significance is preserved.

We expected some disagreement between AH and EP, but not such large discrepancies.

In practice, it implies that the experiment outcomes change depending on the test suites used as a measuring instrument, to the point of obtaining contradictory results.

3 RESEARCH QUESTIONS AND METHODOLOGY

3.1 Research questions

Test suites are being used routinely as measuring instruments in TDD experiments, e.g., [? ? ? ?]. TDD experiments are being combined through meta-analysis [?]. We have shown that

Table 5. Analysis of the QLTY response variable for PT

	AH	EP
(Intercept)	84.76 (10.02)***	28.76 (6.94)***
TreatmentTDD	−15.93 (9.44)	−6.30 (5.05)
TaskMR	−24.47 (9.44)**	20.99 (5.05)***
Group*MR* → *BSK*	16.78 (11.14)	.00 (8.77)
AIC	310.84	284.87
Num. obs.	34	34
Var: Subject (Intercept)	148.66	217.63
Var: Residual	754.33	215.80

***$p < 0.001$; **$p < 0.01$; *$p < 0.05$

Table 6. Analysis of the QLTY response variable for EC

	AH	EP
(Intercept)	55.56 (12.24)***	22.91 (7.36)**
TreatmentTDD	14.06 (12.22)	6.91 (6.87)
TaskMR	−.10 (12.22)	18.90 (6.87)**
Group*MR* → *BSK*	7.41 (12.27)	−.20 (7.82)
AIC	388.01	351.10
Num. obs.	40	40
Var: Subject (Intercept)	6.29	69.72
Var: Residual	1492.43	472.41

***$p < 0.001$; **$p < 0.01$; *$p < 0.05$

the experimental results are conditional on the test suites. The same applies, indirectly, to the meta-analyses based on those TDD studies.

We are concerned about the use of test suites as measuring instruments. We aim to evaluate to which degree similar test suites, e.g., with comparable branch coverage, give different measures. **A better understanding of the role of test suites for measurement will provide decision criteria for the selection or construction, utilization, and sharing of test suites in SE experiments**.

To the best of our knowledge, **this problem has not been addressed in the SE literature**. Given its relevance for the TDD community (and from a general perspective to the entire empirical SE), this paper sets out the following research questions:

RQ1: *How* can we assess the accuracy of the measures obtained using test suites?

Measurement is a complex process. Scientists and engineers have developed specific procedures to assess the accuracy of measures, and compare measurement instruments. These procedures can be applied to test suites.

RQ2: *How much* do the AH and EP datasets differ from each other?

The statistical analyses in Section **??** yield clearly different results. However, such results do not provide an indication of the extent to which the AH and PE datasets differ from each other. Common sense suggests that the differences are large, but we miss a concrete description of *how large* they are.

3.2 Research method

The research questions posed above imply the comparison of two sets of measurements (AH and EP) generated using different test suites (*ad-hoc* and *equivalence partitioning*).

The comparison of measurements is not new in SE. Quite a few papers address the comparison of metrics, e.g., [**? ? ? ? ?**]. However, these works do not put the metrics themselves into question, but they typically examine their predictive ability to choose the ”best” metric for a purpose. Other works, e.g., [**?**] provide metric validation criteria, but these criteria do not include procedures and methods to compare metrics and decide which ones are more accurate. To conclude, **we miss theoretical foundations to analyze and compare measurements in SE**.

In turn, different scientific disciplines (e.g., Medicine, Psychology, and Metrology particularly) have dealt with the problem of comparing measurements, giving rise to different comparison approaches. To the best of our knowledge, none of them has been used in SE so far.

To answer RQ1, we provide in Section **??** an abridged description of the different comparison approaches that apply to our research problem.

To answer RQ2, we apply in Section **??** all suitable comparison procedures to the AH and EP datasets, with a threefold purpose: (1) quantify how large the differences between measurements are, (2) illustrate how the different comparison approaches can be used in practice, and (3) choose the simplest procedure for routinely use in SE.

4 COMPARING MEASUREMENTS

In this section, we will answer **RQ1: *How* can we assess the accuracy of the measures obtained using test suites?**

Depending on the scientific area, different strategies to compare measurement methods are used. Engineers and Natural Science practitioners are probably acquainted with the approaches advocated by JCGM (Joint Committee for Guides in Metrology) and ISO (International Standards Organization). In the Health Sciences, the Bland-Altman plot and the Intraclass Correlation Coefficient (ICC) are often used. The ICC has also been used in Psychology and the Social Sciences.

4.1 Fundamental concepts

Measure theory is the branch of mathematics dealing with the definition and properties of measures [**?**]. In SE, ”measures” are often referred to as ”metrics”. Albeit interchangeable in practice, we will use the term ”measure” due to its specificity. Fenton and Bieman clarify the differences between both concepts [**?** , pp.120-121]).

Metrology is the scientific counterpart, specifically interested in the practical implementation of measurements [**?**] and, more importantly for our purposes, the **comparison of measurements**.

Measure theory has been the target of a substantial amount of research in SE. In turn, metrology has been overlooked (with few exceptions such as [**? ?**]), giving rise to the absence of a methodological background to perform the comparisons of measurements. To fill this gap, we introduce specialized terminology in the following Sections and, later, the comparison methods themselves.

4.1.1 Basic definitions. Metrology uses a standard vocabulary, collected in the VIM[4] (Vocabulaire International de Métrologie) [**?**].

According to the VIM, a *measurement* (2.1)[5] is the "process of experimentally obtaining one or more quantity values" from a *measurand* (2.3).

[4]ISO 3534-1 [**?**] is an ISO standard that provides definitions similar in most respects to VIM. Other bodies, e.g., national standardization agencies may have defined their vocabularies, typically closely related to VIM.

[5]We include the VIM definition number between parentheses, so the reader can trace back to the standard easily.

In our case, the measurand is C++ or Java code satisfying some requirements specification (MR, BSK), and the measurement is the code's QLTY.

Measurement is conducted according to some *measurement method* (2.5), which describes the "logical organization of operations used in a measurement". Among other components, a measurement method includes a *measuring instrument* and a *measurement procedure*. A measuring instrument (3.1) is a "device used for making measurements, alone or in conjunction with one or more supplementary devices".

The *ad-hoc* and *equivalence partitioning* test suites are **measuring instruments** aimed at obtaining QLTY measurements. These instruments were used in conjunction with Eclipse and jUnit/Boost Test frameworks.

A *measurement procedure* (2.6) is a "detailed description of a measurement", typically intended for a human operator. The measurement method corresponds with the utilization of measurement instruments in practice.

Measurement was performed by one researcher (F. Uyaguari) and involves several steps: (1) connecting the subjects' code with the test suites, resolving syntactic and semantic disagreements, (2) running the test suites, and (3) collecting the pass/failure information for the test cases.

4.1.2 Accuracy. For this research, the concept of *accuracy* is particularly relevant. Accuracy (2.13) is the "closeness of agreement between a measured quantity value and a true quantity value of a measurand". Accuracy has two components: *trueness* and *precision*, , as shown in Fig. **??**.

- *Trueness* (2.14) is the "closeness of agreement between the average of an infinite number of replicate measured quantity values and a reference quantity value". Trueness has its origin in the presence of *systematic errors* (2.17), also known as *bias* (2.18).
- *Precision* (2.15) is the "closeness of agreement between [...] measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions". Precision emerges due to the existence of *random errors* (2.19).

4.2 Determination of accuracy in Engineering and the Natural Sciences

The determination of accuracy can be performed in different circumstances or *measurement conditions*: (1) Repeatability; (2) Intermediate precision, and (3) Reproducibility.

4.2.1 Repeatability condition. Repeatability (2.20) is the *uncertainty* (2.26) of a set of measurements conducted using "the same measurement procedure, same operators, **same measuring [instrument]**, same operating conditions and same location [...] on the same or similar objects over a **short period of time**". Repeatability is the metrology concept behind the saying "Measure thrice, cut once" [**?** , p. 3], i.e., the concept of repeatability captures the random variability in the measurement process.

The authoritative source for repeatability analysis is the *Guide to the Expression of Uncertainty in Measurement* (GUM) [**?**]. GUM defines the repeatability uncertainty (denoted s_r) as the standard

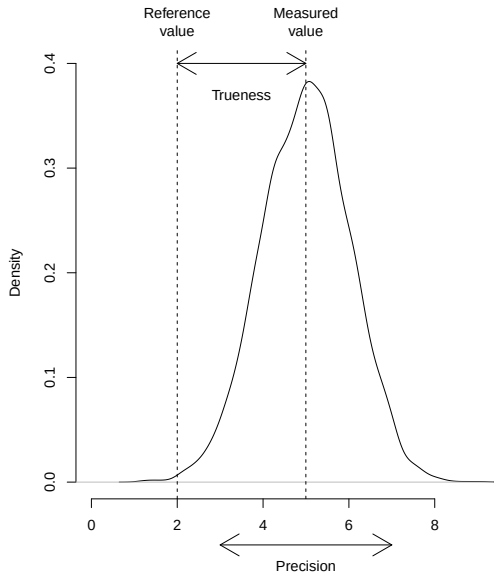

Fig. 1. Components of measurement *accuracy* according to VIM. This figure assumes that the distribution of the random error is *gaussian*, which is usual but not necessarily true in all circumstances

deviation[6] of a set of measures $Y = \{y_1, y_2, ..., y_n\}$:

$$s_r(Y) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2}$$

In this research, Y represents the response variables QLTY and PROD. $\bar{y}$ is the average of all measures, that is, the *measured value* represented in Fig. **??**. s_r represents the *precision* component of the accuracy. The *trueness* component cannot be captured by repeated measurement because the measurement method can be biased. Bias can be removed by calibration, that is, comparing the measurand with a *reference value* (5.18). For instance, lengths can be traced back to the International Bureau of Weights and Measures' meter bar. Reference values are unusual but not impossible in SE. For instance, when we measure the correctness of a program against a specification (our research problem) we could code reference programs of known quality, e.g, programs satisfying none or all requirements. Any of these programs could be used as the reference value (at least in principle; see section **??** for a brief discussion). Trueness would thus be calculated as:

$$\text{trueness} = \bar{y} - \text{reference value} \tag{3}$$

When a reference value is not available, trueness cannot be calculated. However, the repeatability uncertainty is not affected, because the standard deviation is insensitive to location change, i.e.,

$$s_r(Y) = s_r(Y - \text{trueness}) \tag{4}$$

In practice, repeatability analysis assumes that the instruments are not biased.

[6]In some cases, measurements are not obtained directly. For instance, determining the distance to a distant object using parallax is an indirect measurement, based on two direct measurements: length and angle. The contributions of the length and angle measurements to the uncertainty should be independently determined and later combined using a formula related to Eq. **??**; see [**?** , Section 5] for details.

4.2.2 Intermediate precision analysis. The *intermediate precision* (2.22) is a "condition of measurement [...] that includes the same measurement procedure, same location, and replicate measurements on the same or similar objects over an **extended period of time**, but may include other conditions involving changes". The changes "can include [...] operators, and **measuring [instruments]**".

The intermediate precision[7] deals with two different measurement situations:

- The variability in the measures that take place over time naturally, e.g., due to varying temperatures throughout the year.
- The changes in the measurement environment. A typical scenario is the usage of different instruments, e.g., thermometers, to perform the measurements.

> Our research addresses the second situation. The measuring instruments (the *ad-hoc* and *equivalence partitioning* test suites) have different precisions that influence the overall precision of the measurements.

The intermediate precision, denoted s_{R_w}[**?** , p. 1], is calculated as[8]:

$$s_{R_w} = \sqrt{s_M^2 + s_r^2} \tag{5}$$

where s_r^2 is the reproducibility uncertainty described in the previous Section and s_M^2 is the uncertainty due to the measuring instrument. There are several recommendations to calculate s_M^2: the GUM [**?**], NordTest TR 537 [**?**], and ISO 5725-3 [**?**]. The later is particularly useful for its simplicity. The intermediate precision is calculated using the nested model[9]:

$$Y = Program/Instrument + \epsilon \tag{6}$$

The *Instrument* is the random factor that represents the measuring instruments implemented with the **AH and EP test suites**. s_M is given by the associated component of variance, that we will describe in Section **??**. ϵ represents the lack of precision that *cannot* be assigned to the *Instrument*, i.e., $\epsilon = s_r^2$ (the repeatability uncertainty).

4.2.3 Reproducibility uncertainty. Reproducibility (2.24) is the uncertainty of a set of measurements performed on "**different condition[s] of measurement**, out of a set of conditions that includes different locations, operators, measuring [instruments], and replicate measurements on the same or similar objects".

The calculation of the reproducibility uncertainty (denoted s_R) is described in ISO 5725-2 [**?**]. In this part of the standard, reproducibility uncertainty is defined as the uncertainty due to the lab where measurements are conducted. This assumption does not match our research problem, so we will not elaborate it further. However, it could be relevant in other measurement comparison scenarios in SE, e.g., when different research groups participate in the measurement of a multi-site experiment.

4.2.4 Standard vs. expanded uncertainties. s_r, s_{R_w} and s_R are standard uncertainties, that is, the σ parameter of the normal distribution displayed in Fig. **??**. However, in a normal distribution, a large percentage of values (around 32%) are more than 1 standard deviation apart the average (the measured value in Fig. **??**). The *expanded uncertainty* (2.35) provides more significant information

[7]The intermediate precision is also termed "within-lab reproducibility" [**?** , p. 1] because it addresses the variability that happens inside a measuring facility. This stands in contrast to the "between-lab reproducibility", described in Section **??**.
[8]The square root is one of the realizations of the *propagation law* defined in the GUM [**?**]. It applies when several (≥ 2) independent uncertainties are added.
[9]Notice that the same programs are measured using AH and EP

about the degree to which measures may differ. Expanded uncertainties are the limits of the interval which includes $(1-\alpha)\times 100\%$ of the differences among measures.

In the case of this research (2 measuring instruments, n pieces of code, and one measure per instrument) the expanded uncertainty[10] would be:

$$t_{(1-\alpha/2),(n-2)} \times s_{R_w} \tag{7}$$

$k = t_{(1-\alpha/2),(n-2)}$ is known as *coverage factor* (2.38). When $n \geq 30$, the normal approximation can be used. Typically, $\alpha = 0.05$, and $k = Z_{(1-\alpha/2)} = 1.96$. In practice, k is rounded up to 2.0 [**?** , p. 24]. The expanded uncertainty is thus defined as:

$$2 \times s_{R_w} \tag{8}$$

and represents the fact that any measure can **differ up to** $\pm 2 \times s_{R_w}$ **units** from the average *measured value* (see Fig. **??**).

4.3 Determination of accuracy in the Health and Social Sciences

The correlation coefficient has been the usual procedure to compare measurements in the Health and Social Sciences. This procedure is rather well-known and we have used it already in Section **??**.

However, the correlation coefficient exhibits several problems as a comparison procedure. For instance, data with poor agreement[11] can produce high correlations [**?**]; this is exactly what we have observed in Section **??**. These problems recommended the design of alternative procedures, such as the Bland-Altman method and the usage of the ICC. We will describe both of them in the following sections.

4.3.1 Bland-Altman method. The Bland-Altman method [**?**] is the *de facto* standard for the comparison of measuring instruments in medicine. Contrary to the GUM and ISO, several objects (not the *same* or *similar* object) are involved in the measurement. Each object is measured twice using a different instrument. In our research problem, it implies that we would need two sets of measures: $Y_{AH} = \{y_{AH_1}, y_{AH_2}, \ldots, y_{AH_n}\}$ and $Y_{EP} = \{y_{EP_1}, y_{EP_2}, \ldots, y_{EP_n}\}$, being $1 \leq i \leq n$ different pieces of code.

The Bland-Altman method starts with the calculation of the difference between[12] measurements:

$$d_i = (y_{AH_i} - y_{EP_i}) \tag{9}$$

Next, the average of the differences d_i is calculated as:

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n}(y_{AH_i} - y_{EP_i}) \tag{10}$$

and the standard deviation as:

$$s_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2} \tag{11}$$

The Bland-Altman method does not assume that the instruments are unbiased; for this reason, $\bar{d}$ represents the mean difference between the measurements obtained with the AH and EP test

[10] The same formula applies to s_r and s_R

[11] The term "accuracy" is not often used in the Health and Social Sciences. When variables have interval/ratio types, the term *reliability* is frequently used; for nominal/ordinal variables, the most common term is *agreement* [**?**]. The terms *consistency* and *conformity* can also be found as synonyms of precision and trueness [**?**]. Nevertheless, the terms vary depending on the source. For instance, Bland and Altman [**?**] use the term "Agreement" instead of "Reliability" with ratio scales. We will use the terms defined in the VIM [**?**] and reported in Section **??**.

[12] Procedures for more than two measuring instruments have been proposed in the literature, e.g., [**?**].

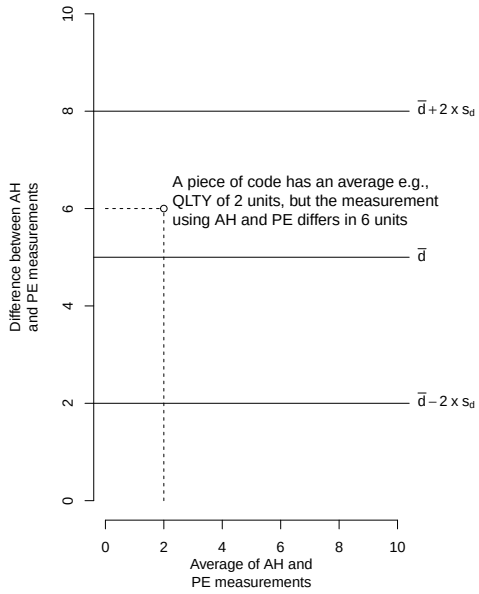

Fig. 2. Interpretation of the Bland-Altman plot

suites. If the instruments were unbiased, then $\bar{d} = 0$. s_d is the standard deviation of the difference between measures.

The Bland-Altman method has an associated graphical representation (the Bland-Altman plot), shown in Fig. **??**. This graph plots the mean values obtained by both measuring instruments:

$$\frac{y_{AH_i} + y_{EP_i}}{2} \tag{12}$$

that is, the best estimation of the true measurement, against their difference:

$$y_{AH_i} - y_{EP_i} \tag{13}$$

A horizontal line is drawn at the mean difference $\bar{d}$. Additionally, the graph also displays two additional horizontal lines located at[13]:

$$\bar{d} \pm 2 \times s_d \tag{14}$$

Assuming a normal distribution for the differences, these limits enclose 95% of the differences d_i. **These limits represent how much the measure on the same object varies when one instrument (AH) or another (EP) is used**.

4.3.2 Intraclass correlation coefficient. In the Health and Social Sciences, measurements are often scores assigned by human judges. In this case, "accuracy" is termed (inter-rater) *agreement*, and represents the degree to which judges coincide with each other when rating the same item. The similarity between "judges" and "measuring instruments" is apparent. The connection was made for the first time by Lee et al., [**?**]; they proposed using the ICC as a measure of accuracy.

[13]According to Bland and Altman [**?**], either the level $k = 2$ or $k = 1.96$ can be used. We use $k = 2$ to highlight the similarities among comparison methods.

There are diverse strategies to assess inter-rater agreement[14]. For our research problem, the right one is the *Inter-class Correlation Coefficient* (ICC) Model 3 (mixed factorial design), also known as ICC(3, 1) in some statistical packages such as SPSS® [**?** **?**].

The underlying idea is as follows: ICC measures the strength of the relationship among items belonging to some class and compares it to the total variability. In this case, the class is each piece of code. Each class contains two values: the measures taken on that piece of code using both the AH and EP test suites. Mathematically speaking, ICC(3,1) is defined as:

$$\rho = \frac{s_M^2}{s_M^2 + s_e^2} \tag{15}$$

where s_M^2 is the between-method variance[15], and s_e^2 the error variance. s_M^2 and s_e^2 can be obtained from the linear model:

$$y = Instrument + Program + \epsilon \tag{16}$$

whose interpretation is similar to Eq. **??**.

5 COMPARISON OF THE AH AND EP MEASURING INSTRUMENTS

In this section, we will answer **RQ2: *How much* do the AH and EP datasets differ from each other?**

In terms of measurement theory, analyzing the differences between the AH and EP datasets is equivalent to assessing the accuracy (trueness and precision) of the AH and EP measuring instruments. We will use the four comparison methods (repeatability, intermediate precision, Bland-Altman plot and the ICC) described in Section **??**.

5.1 Repeatability analysis

Test suites are popular measuring instruments because the measurement is automatic and **repeatable**. Running the same test suite on the same code yields always the same results[16].

The only source of uncertainty when using test suites for measurement is the human operator. To perform the measurement, the operator at least should: (1) download the subject's code, (2) add the AH end EP test suites, (3) make the necessary adjustments to the code (e.g., resolve compilation problems), (4) run the test suites, and (5) write down the results. Measurement problems take place in the steps 3-4. Steps 1-2 influence sample preparation (not measurement), whereas in step 5 only transcription problems may take place.

Step 3 can be performed in different ways. One option is not to make any change to the subjects' code. In this case, due to the repeatable character of the measurement with test suites, the obtained measures have always the same value. This implies that $s_r = 0$.

> In the PT and EC experiments, the measurer made small changes (e.g., method names, the order of parameters, etc.) to avoid zero QLTY scores due to clerical errors. Using this strategy, when measurements are repeated **in a short time**, the results do not vary, because the changes are predictable. Thus, $s_r = 0$ again.

[14]Refer to [**?**] for an in-depth description.

[15]The values of s_M^2 given by Eqs. **??** and **??** are close but not alike. An example appears in section **??**.

[16]Varying results are possible when the code depends on some random input. For proper testing in those cases, the random portion should be isolated, e.g., using mocking, to achieve deterministic results [**?**].

More complex strategies for connecting the subject's code and the test cases (e.g., fixing loop bounds, order or invocation, etc.) may be more demanding in memory's terms so that the measures do change, giving $s_r > 0$. It does not happen in this research.

5.2 Intermediate precision

The intermediate precision s_{R_w} represents the uncertainty produced by the measuring instruments. The intermediate precision assumes that the replicate measurements are made "on the same or similar object" (see Section **??**). This implies a problem in SE experiments. We cannot assume that the pieces of code collected in an experiment are similar; in fact, they exhibit a great degree of variation.

Providentially, the nested model proposed in ISO 5725-3 [**?**] can be expanded with additional factors, as long as the nesting structure is specified in the model. We have performed the measurement on 74 programs from both the PT and EC experiments[17]. The Measurement Instrument is nested within the new factor Program. The corresponding model is a simple extension [18] of Eq. **??**:

$$QLTY = Program/Instrument + \epsilon \tag{17}$$

The analysis was conducted with the following R command:

```
lm <- aov(QLTY ~ Program/Instrument,
          data = expdata)
```

Table 7. Estimation of s_M using Eq. **??**

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Program	73.00	104051.38	1425.36		
Program:Instrument	74.00	71577.93	967.27		
Residuals	0.00	0.00			

Table **??** shows the analysis results. The residual is zero, because as we explained above $s_r = 0$. s_M^2 is calculated as [**?** , pp. 348-350]:

$$s_M^2 = MS(Program : Instrument) = 967.27 \tag{18}$$

The intermediate precision of the measuring instruments is $s_M = \sqrt{s_M^2 + s_r^2} = \sqrt{967.27 + 0} = 31.1$. Using $k = 2$, the expanded uncertainty (see Eq. **??**) is $2 \times 31.1 =$.

> When the Ah and EP test suites are used as measuring instruments (e.g., in two different experiments, later combined using meta-analysis), measures that theoretically speaking should be similar (e.g., because the measured programs exhibit the same QLTY) can differ up to ±62.2%.

5.3 Bland-Altman method

The Bland-Altman method uses the differences between measures (Eq. **??**) to calculate the accuracy of the measuring instruments. The mean difference $\bar{d} = -33.21$ means that the AH and EP measuring

[17] The origin of the code (PT or EC sites) is irrelevant in the accuracy of the AH and EP test suites

[18] This model can be seen as a restricted version of a more general procedure for the comparison of variances. See [? , Chapter 9] for details.

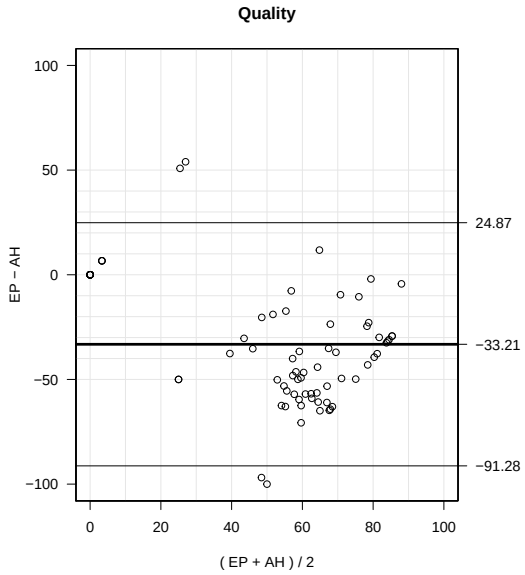

Fig. 3. Bland-Altman plot

instruments differ 33.21% units *in average* (notice that QLTY is measured as a percentage). The standard deviation of the differences is $s_d = 29.04$.

Fig. **??** shows the same information visually. The central line represents the mean difference $\bar{d} = -33.21$, whereas the top and bottom lines delimit the range of variation of the differences between measurements (the points displayed in the plot). Those limits are calculated as $\bar{d} \pm 2 \times s_d = -33.21 \pm 58.08$.

> According to the Bland-Altman method, the measures made on the same code by the AH and EP test suites may vary up to 58.08% in either direction. Actually, we can see in Fig. **??** some measures that even exceed such limits. The AH test suite tends to give higher values (33.21% in average) than the EP test suite.

5.4 ICC

The ICC is obtained using Eq. **??** which, in turn, requires the calculation of the lineal model depicted in Eq. **??**. That model is calculated using the R command:

```
lm <- aov(QLTY ~ 1 +  Instrument + Program,
          data = expdata)
```

The results of the analysis is shown in Table **??**. There are strong similarities with Table **??** because the models are rather similar. s_M can be calculated from Table **??** as [**?** , Table II]:

$$s_M^2 = \frac{MS(Instrument) - MS(Residual)}{p} \tag{19}$$

The notation has the same meaning than in Section **??**. s_M^2 is thus:

Table 8. Estimation of s_M and ϵ using Eq. **??**

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Instrument	1.00	40803.13	40803.13	96.79	0.00
Program	73.00	104051.38	1425.36	3.38	0.00
Residuals	73.00	30774.79	421.57		

$$s_M^2 = \frac{4.080313 \times 10^4 - 421.57}{74} = 545.7$$

The value of s_M^2 is slightly different than the one obtained in Section **??** because we are using different linear models in the calculations. The ICC is (see Eq. **??**):

$$\rho = \frac{s_M^2}{s_M^2 + s_\epsilon^2} = \frac{545.7}{545.7 + 421.57} = 0.56$$

The interpretation of ρ is not evident. As a general rule, the lower the value, the less related (/similar) the measures taken by the AH and EP test suites for the same program. To interpret the ICC more easily, reference values are typically given in the literature.

> Two measurement instruments are considered to have a good agreement when $\rho \geq 0.75$ [**?**]. In our case, the AH and EP test suites do not achieve that level.

6 DISCUSSION

6.1 Accuracy of the AH and EP test suites

The interpretation of the ICC is contrived. This fact, in addition to some weaknesses [**? ?**] of the ICC makes this procedure not that useful as a comparison method. In turn, the expanded uncertainties provided by the ISO and Bland-Altman are self-explanatory.

Regardless of the comparison method used (ISO5725-3, Bland-Altman, or ICC), the accuracy of the AH and EP test suites is rather weak. The Bland-Altman method is particularly illustrative in this regard. The difference between two measures taken on the same program can differ up to 58.08% in either direction.

The Bland-Altman plot, shown in Fig. **??**, is even more illustrative. Only three points, i.e., three programs, are close to the value "0" of the vertical axis, which denotes the coincidence between the measurements obtained with the AH and EP test suites. All other points are 10%, 20%, or further apart.

> The obvious conclusion is that the AH and EP test suites are incompatible measuring instruments. They cannot be used together, because the difference between measurements obtained with each of them is too large. Such differences are likely the origin of the different experimental analysis results that we described in Section **??**.

6.2 Reason of measurement differences

Measure differences may have multiple origins, some of them minute details. For instance, the test cases in Listing **??** pass for the Java code `int sum(int a, int b){ return a + b; }`, but the test case `testThreePlusMinusTwoGivesOne()` fails for the C code `unsigned char sum(unsigned char a, unsigned char b){ return a + b; }` (in fact, the code probably would not even compile).

The AH and EP test suites are not affected by data types issues, like in the previous example. When the experiments PT and EC were conducted, experimental subjects received code stubs

including class and method definitions. The reason for the inconsistent measures lies in the type, and number, of test cases defined in each test suite.

Figure **??** shows a scatter plot. On the x-axis, we represent the *true value* of a measurement. This true value was obtained using reference code, i.e., code that satisfies all requirements[19]. The task that appears in Figure **??** is BSK, and the metric displayed is PROD (the plot is easier to understand using PROD instead of QLTY). The y-axis value is the measured value using the AH test suite. If AH provided accurate measures, all points would lie in the diagonal line. Departures from the diagonal line represent measurement errors, the larger the farther apart from the diagonal line the points are. Figure **??** displays the same information for the EP test suite.

The points in Figure **??** are scattered around the diagonal line. It implies that AH captures the meaning of the PROD metric. However, individual measures may have large errors. These errors have their primary origin in redundant test cases, i.e., test cases that check the same testing condition. It is fairly easy that *ad-hoc* test case designers insist on multiple testing the same requirement, especially when such requirement is perceived as important. Such test cases pass or fail together, causing large up and down variations in the measured values.

The points in Figure **??** exhibit a different shape. Most points are located behind the diagonal line, meaning that measured values are systematically lower than true values. Measured PROD values never exceed 40%. The heuristics of equivalence partitioning testing explain this behavior. Equivalence partitioning puts special emphasis in *invalid classes* which programmers (and *ad-hoc* test case designers) tend to ignore. Actually, the reference code used to create Figures **??** and **??** was obtained from high performing experimental subjects. Overlooking invalid classes lead to systematic low PROD values.

Notice that we are not expressing an opinion about test case design methods. We simply trace (in a simplified manner, as issues may be multiple) the measurement errors to the test suite construction strategies. However, the strategy is not the critical point. The key is that measurement instruments could have been piloted to verify that they produce the right measures, i.e., the ones in the diagonal line, before actual use.

6.3 Impact in TDD research

Several synthesis works on TDD have been published recently, e.g., [**? ? ? ?**]. These works identify 90 empirical publications, including surveys, experience reports, case studies, quasi-experiments, and controlled experiments. At least, another 24 studies have been published in the past years, e.g., [**?**], thus not being included in the synthesis works[20]. Out of those 114 publications, 15 experiments measure external quality, i.e., the response variable QLTY used in the PT and EC experiments. 13 out of these 15 experiments use test suites for measurement. The listing is available in Table **??**. Some patterns are easily noticeable:

- TDD experiments require subjects to code some experimental task. However, task specifications **are not usually disclosed**. In some cases, not even the name of the task is reported.
- Measurement is always performed using test suites. However, with the sole exception of George & Williams [**?**], the tests suites **are not publicly available**.
- Finally, roughly 50% of the test suites have been created by the researchers themselves; **the origin of the remaining 50% is unknown**. The strategy for test suite creation **is never reported**.

[19]The reference code and dataset generation procedures for this section are available as one Eclipse workspace at https://github.com/GRISE-UPM/TestSuitesMeasurement/tree/master/calculation_of_deviations.

[20]We exclude our own publications, e.g., [**?**] from these figures.

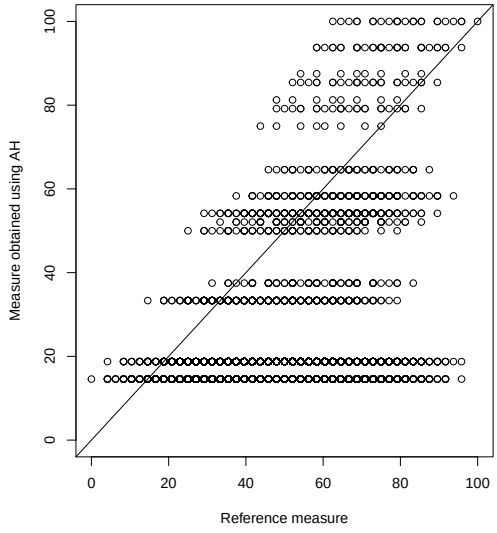

Fig. 4. Deviations from reference value (BSK using the AH test suite)

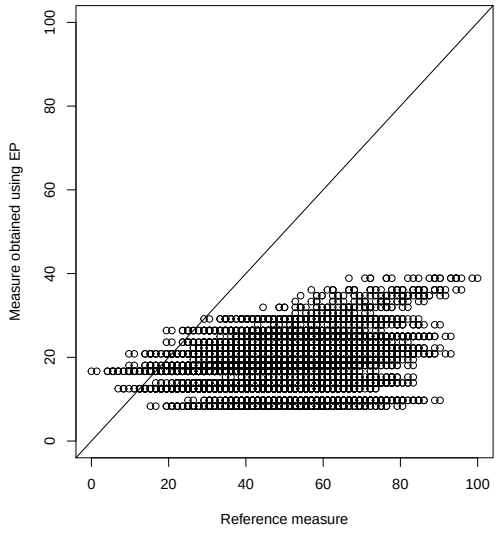

Fig. 5. Deviations from reference value (BSK using the EP test suite)

Given the different measures that each test suite yields, TDD experiments may come to different conclusions due to measurement only. The fact that almost 50% of experiments in Table **??** use Robert Martin's Bowling Score Keeper (BSK) strengthens our beliefs because our experiments use

a modified version of the same task[21] and we have already confirmed the impact of the test suite construction strategy on BSK.

6.4 Impact in SE research

Test suites are not used in TDD only. Many experiments, e.g., [**? ? ?**], conducted in other areas of SE, utilize test cases for measurement purposes. It is highly likely that the same dependence between test suites and experimental results take place there too.

7 CONCLUSIONS

The software metrics area is quite mature in SE. Their formal properties [**?**] and some common pitfalls, e.g., [**?**], are well understood. However, when dealing with experimental data, it seems that we have overlooked, at least partially, the complexities of measurement. There are several reasons for that: (1) in many cases, the metrics and measuring instruments should be specifically designed for an experiment, (2) the entities of interest in SE are often theoretical constructs, so that objective measurement instruments cannot, by definition, be ever available, (3) we probably trust in excess in the power of statistics, etc.

We have confirmed in this research that different test suites (often used as measuring instruments) give different measures on the same program. Such differences are so radical that they reverse the effects of the factors in the statistical analyses. We have restricted our inquiry to the response variable *external quality*, frequently used in TDD experiments. However, we believe that our findings can be extrapolated to other response variables, research areas and even other research methods, e.g., case studies.

Experimentation is not mature enough to introduce standard measures and measurement instruments. Of course, benchmarks can be adopted. However, the mere adoption of a benchmark does not solve the problems described in this paper because nothing guarantees that such a benchmark provides the *right* measures. Even so, some action should be taken to avoid the harmful effects of metrics and measuring instruments in SE experimental research. In our opinion, three measures can be beneficial for the experimental community:

- Researchers should disclose not only the measurement results, i.e., the refined data which proceeds to analysis but also the **raw data** (e.g., subjects' code) and the **measurement procedure and instruments**. This enables later critical examination and the conduction of a wide range of replications, in particular, re-analysis [**? ?**], where the raw data is independently re-processed before analysis.
- The properties of the measures and measuring instruments should be considered before actual measurement takes place. In many cases, measurement standards, e.g., programs satisfying subsets of requirements, can be easily created well before the experiment is conducted, so that formal analysis and empirical studies are possible.
- When standards are not available, experimenters could use different measures and instruments to avoid threats to construct validity. Coherent results obtained with different instruments would increase the confidence in the experiment results. Just to cite an example, in the PT and EC experiments, the *Treatment* (ITLD vs. TDD) **was largely unaffected** by the AH and EP test suites. This fact provides us some relief. If the treatments were unaffected, the meta-analyses and other secondary studies based on these data would produce correct results.

Finally, in this research, we have only addressed the influence of the measuring instruments in the measurement results. However, the measurement process, as indicated in Section **??**, has several components. They all (in particular, the measurer and the manipulations before applying

[21] See footnote **??**.

Table 9. Tasks and test suites used by TDD experiments

Study	Site	Experimental task	Spec avail.?	Test suite avail.?	Provenance	Construction strategy
Čaušević et al. [?]	Academy	Robert Martin's Bowling Score Keeper (BSK)	No	No	Not specified	Not specified
Desai et al. [?]	Academy	Not specified	No	No	Created by the researcher(s)	Not specified
Erdogmus et al. [?]	Academy	Robert Martin's Bowling Score Keeper (BSK)	No	No	Created by the researcher(s)	Not specified
Fucci and Turhan [?]	Academy	Robert Martin's Bowling Score Keeper (BSK)	No	No	Created by the researcher(s)	Not specified
Fucci et al. [?]	Academy	Robert Martin's Bowling Score Keeper (BSK)	No	No	Not specified	Not specified
George and Williams [?]	Industry	Robert Martin's Bowling Score Keeper (BSK)	In B. George's Ph.D. Thesis [?]	In B. George's thesis [?]	Created by the researcher(s)	Not specified
Geras et al. [?]	Industry	Program A (registering a new project) + Program B (recording time against a project)	Yes	No	Created by the researcher(s)	Not specified
Gupta and Jalote [?]	Academy	Student registration system / Simple ATM system	Yes	No	Created by the researcher(s)	Not specified
Muller and Hagner [?]	Academy	GraphBase (related to graphics)	No	No	Not specified	Not specified
Munir and Moayyed [?]	Industry	Robert Martin's Bowling Score Keeper (BSK)	In H. Munir and M. Moayyed's M.Sc. Thesis [?]	No	Not specified	Not specified
Pančur and al. [?]	Academy	Not specified	No	No	Not specified	Not specified
Pančur and Ciglarič [?]	Academy	Distributed database server with built-in data replication mechanism + Chat server	No	No	Created by the researcher(s)	Not specified
Vu et al. [?]	Academy	Not specified	No	No	Not specified	Not specified

the measuring instrument) can influence the measurement result too. The assessment of the impact of those elements will be future research.

8 ACKNOWLEDGMENTS

This work was partially supported by the Spanish Ministry of Economy and Competitiveness research grant TIN2014-60490-P, and the Finish TEKES research grant ESEIL (FiDiPro scholarship).