Metrics for Data Uniformity of User Scenarios through User Interaction Diagrams

Douglas Hiura Longo and Patrícia Vilain
Informatics and Statistics Department,
Federal University of Santa Catarina,
Florianopolis, Brazil
douglashiura@inf.ufsc.br, patricia.vilain@ufsc.br

***Abstract*— In the software development process, the acceptance testing may be used by non-technician users to define software requirements. In this article, we use the US-UIDs (User Scenarios through User Interaction Diagrams) as automated acceptance tests in order to provide communications and collaboration between programmers and users. We propose three metrics for measuring the data uniformity in the US-UIDs. These metrics are investigated in four projects. The resulting measures from the investigations of the four projects are used to build a scale with classes to classify the uniformity of the US-UIDs. The classes (duplication, uniformity, irregularity) were created from empiric evaluation and compared to the measures from the offered metrics. The classification purpose is to identify the US-UIDs as uniform or irregular.**

Keywords: US-UID; User Scenarios Through User Interactions Diagrams; ATDD; Uniformity; Acceptance Test; Automated Test; Executable Test; Executable Requirement; Quality Factor; Requirements; Requirements Specification; Metrics.

I. INTRODUCTION

Analogous to the Test-Driven Development (TDD) [1], the Acceptance Test-Driven Development (ATDD) includes team members with different perspectives (client, developer, tester) collaborating to write acceptance testing before deploying the functionality [2]. Teams that try ATDD generally find that only by defining acceptance tests when discussing requirements outcomes there will be a better understanding. However, the acceptance tests force us to reach a solid agreement about the exact behavior that the software should expose [3].

The User Scenarios through User Interaction Diagrams (US-UIDs) are suggested to allow that non-technician users define software functional requirements in the ATDD approach [4, 5]. The US-UIDs are used to specify primarily the values of the information exchanged between the user and the computer in tasks that represent functional requirements, mainly in information systems and can be used as automated acceptance tests [6].

To use a US-UID as an automated test, the following steps are performed: specification of the US-UID, nomination of the fixtures that represents the US-UID elements, and creation of the glue code that will link these fixtures to the SUT (System Under Testing) code [6]. Although data uniformity problems are generated in the specification step, the identification of these problems occurs usually in all three steps of test automation.

Figure 1 shows a pair of US-UIDs to exemplify the uniformity problem in the data. The example considers only a part of the original US-UIDs to show data with uniformity problem. Both US-UIDs show the same functionality, the specification of the authentication system, but show different values to the information. The example highlights two uniformity problems, where the first is related to the values of the user inputs “Mary” and “John”. Both values are different from one another but have the same sense. The sense is clearly a user name. This sense can be extracted through the experience with people names and with data from close elements in the US-UID. In this authentication system, it is hard to deconstruct the sense of user names for these values, however, for specific systems with little known specialties between the stakeholders, the non-uniform data causes loss of sense. The second uniformity problem is related to the system outputs “Enter” (Figure 1, US-UID A) and “Log In” (Figure 1, US-UID B). Both systems outputs represent in SUT implementation the button text for the action of entering in the system. For communication purposes, the stakeholder can understand the text of these two system outputs that, although different, have the same sense. However, as an automated test, there will be a problem with the implementation. As the system outputs of a US-UID are also assertions and the assertions rely and capture the SUT values, it is inviable that an implementation, in an automated manner, could answer to two distinct values (“Enter” and “Log in”) in the same way, unless the same action is duplicated in the SUT.

The irregular data generation occurs mainly in the shared specification, i.e., when more than one stakeholder specifies US-UIDs to a same system. Thus, when there is more than one person specifying different US-UIDs, usually there is no care to keep the data uniform, as each person uses the data domain that he/she knows for the test. Other way to generate non-uniform data is by the partial repetition of a US-UID path. The partial repetition is necessary because there cannot be deviations and branches in the US-

DOI reference number: 10.18293/SEKE2018-075.

UIDs [4, 5]. Therefore, when you repeat manually parts of the US-UIDs, it is usual to change small details and not apply the changes in all the US-UIDs, that way, there will be a uniformity problem that will be spread in all test automation steps. Furthermore, as the US-UIDs of a project increases, more difficult becomes the uniformity problems identification.

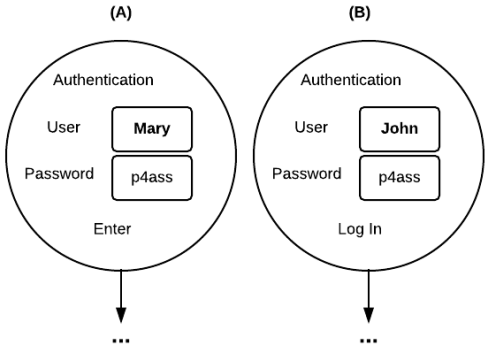

Figure 1. Two fragments of US-UIDs with dat uniformity problems

The general purpose of this work is to suggest metrics for measuring data uniformity from US-UIDs. The specific purpose is the empiric evaluation of the metrics to create a classification that allows the identification of US-UIDs with irregular data.

This paper is organized as follows. The second section shows the theory basics about the US-UIDs. The proposal is detailed in the third section. The fourth section shows the projects of the evaluation. The fifth section shows the results of the evaluation. Finally, the sixth section shows the conclusions.

II. BACKGROUND

The US-UIDs are used for specifying software requirements. The US-UIDs have been suggested as a specialization of the UID technique [7], where the abstract information is replaced with concrete values from the user scenarios. The applicability of the US-UIDs is usually made by non-technician users to create acceptance testing before the development. In agile development teams, the US-UID can be used for communication and collaboration between the stakeholders in software development.

Figure 2 shows an example of US-UID with the interactions of the sum operation using of a calculator. According to Longo and Vilain [4], this example was adopted to explain to non-technician users how to specify the US-UIDs. With the knowledge acquired from the example, non-technician users have participated in experiments to evaluate the correctness and the completeness of the US-UIDs [4, 5].

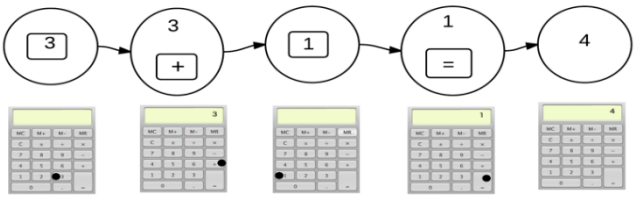

Figure 2. Example of a US-UID of a calculator sum operation, as suggested by Longo and Vilain [4, 5]

In this example, the user enters the values of the sum operation (3 + 1 =) and the system shows the result (4). In the example, five states of interaction are shown, meaning each state of interaction (ellipse) contains the user input and the system output values. The state of interaction flow is represented by the arrow direction through the states of interaction. The initial state is the first state of interaction that follows the arrows and the end state is the last state in the flow. Table I shows the language symbols from the US-UIDs.

TABLE I. SYMBOLS FOR THE LANGUAGE OF US-UIDS [1, 2]

Symbol	Use
	**Ellipse** – represents a state of interaction.
	Arrowed line – represents the direction flow, i.e., the transition between interactions states.
	**Rectangle** – represents the user input, its value is represented by a set of characters placed within the rectangle.
Characters sequence	**Value** – represents the system output, where a set of characters is placed within the ellipse.

A. Mathematical Model of the US-UIDs

The US-UIDs can be represented in a mathematical model. Thus, as suggested by Longo et al. [6], the structure of a US-UID is formed by a set of states of interaction and each state of interaction is, in turn, formed by a set of user inputs and a set of system outputs. A state of interaction is represented by:

$$\delta_i = \{\varepsilon_{i1}, \varepsilon_{i2}, \varepsilon_{ij}, \dots, \varepsilon_{in}, o_{i1}, o_{i2}, o_{il}, \dots, o_{im}\}, \quad (\forall i\ (i = 1;\ k)), (\forall j\ (j = 1;\ n)), (\forall l\ (l = 1; m)) \tag{1}$$

Given a state of interaction, n is the amount of system outputs and m is the amount of user inputs, o_{il} is the l-th system output from i-th state of interaction, ε_{ij} is the j-th user input of the i-th state of interaction. A US-UID is represented as follows:

$$\tau_t = \{\delta_{t1}, \delta_{t2}, \dots, \delta_{ti}, \dots, \delta_{tk}\}, \quad (\forall i\ (i = 1; k)), (\forall t\ (t = 1; d)) \tag{2}$$

The k is the amount of states of interaction of the US-UID. The δ_{ti} is the i-th state of interaction of the t-th US-

UID. As a restriction, the set must have at least a state of interaction.

III. PROPOSAL

The lack of uniformity in data may cause problems in the communication and collaboration between the stakeholders, when defining the fixture names and the glue code. Therefore, it is important to evaluate the data uniformity in the US-UID specification step. For evaluation, easy-to-apply uniformity metrics are useful, especially, computational metrics, with measures of easy availability for the stakeholders.

This paper proposes three metrics to measure the uniformity of US-UIDs data. The proposed metrics are of absolute uniformity, absolute irregularity and relative uniformity. The metrics for absolute uniformity and absolute irregularity are created by comparing pairs of US-UIDs. A set of pairs of US-UIDs is generated from a set of US-UIDs. The set of pairs of US-UIDs is defined by:

$$\psi = \{(\tau_1, \tau_2), (\tau_1, \tau_3), \dots, (\tau_t, \tau_q), \dots, (\tau_{(d-1)}, \tau_d), \dots, (\tau_d, \tau_{(d-1)})\}, (\forall t\ (t = 1; d)),\ (\forall q\ (q = 1; d)), t \neq q, d > 1 \quad (3)$$

where, d is the amount of US-UIDs from generated set. The set generated with the pairs of US-UIDs should have at least two US-UIDs. Both τ_t and τ_q are two any US-UIDs from a US-UIDs set. (τ_t, τ_q) is a pair formed by distinct US-UIDs. For the pairs formation, the restriction is that the formed pairs cannot exist with the same US-UIDs, such as $t \neq q$.

A. Metrics for the absolute uniformity

The absolute uniformity is calculated for each pair (τ_t, τ_q) where each pair of US-UID is split by user inputs and system outputs. The user inputs and system outputs are also compared in pairs.

A pair of uniform system outputs is formed by a system output of τ_t and by another system output of τ_q. The criterion to form the pairs of system outputs is that the data must be identical. The measure of the absolute uniformity of system outputs from a pair (τ_t, τ_q) is calculated by the following formula:

$$UniformOutput_{(\tau_t,\tau_q)} = \sum_{i=1}^{k}\sum_{l=1}^{m} \begin{matrix} 1\ if\ o_{il} \in \tau_q \\ 0\ otherwise \end{matrix}, \quad (\forall \delta_{ti} | \delta_{ti} \in \tau_t), (\forall o_{il} | o_{il} \in \delta_{ti}) \quad (4)$$

The expression $o_{il} \in \tau_q$ means that the system output o_{il} belongs to one of the system outputs among all the states of interaction τ_q. The absolute uniformity of the system outputs is equal to the count of all system outputs from all states of interaction of a US-UID that have a pair compared to another US-UID. The absolute uniformity of the user inputs ($\text{UniformInput}_{(\tau_t,\tau_q)}$) is built in a similar way to the metrics for system outputs.

B. Metrics for the absolute irregularity

In this study, absolute irregularity is the complement of absolute uniformity. The metrics for the absolute irregularity is built for a pair (τ_t, τ_q) of US-UIDs. The construction of this metric is similar to the metrics for absolute uniformity, in that the metrics for absolute irregularity is sectioned by user inputs and system outputs. The absolute irregularity of the system outputs from a pair (τ_t, τ_q) of US-UIDs is calculated in relation only to system outputs belonging to τ_t. So, the system outputs belonging to τ_t that do not have a pair are counted as being irregular. The criteria for not forming a pair of system outputs is that a system output belongs to τ_t and that no other belongs to τ_q with identical data. The absolute irregularity of the system outputs of a pair (τ_t, τ_q) is calculated by the following formula:

$$NonUniformOutput_{(\tau_t,\tau_q)} = \sum_{i=1}^{k}\sum_{l=1}^{m} \begin{matrix} 1\ if\ o_{il} \notin \tau_q \\ 0\ otherwise \end{matrix}, \quad (\forall \delta_{ti} | \delta_{ti} \in \tau_t), (\forall o_{il} | o_{il} \in \delta_{ti}) \quad (5)$$

The expression $o_{il} \notin \tau_q$ means that the system output o_{il} does not belong to the system outputs among all states of interaction τ_q. The irregularity metrics for user inputs ($\text{NonUniformInput}_{(\tau_t,\tau_q)}$) is built in a similar way to the metrics for system outputs.

C. Metrics for the relative uniformity

The relative uniformity metric is computed from the absolute uniformity and absolute irregularity metrics. So, for the measure of the relative uniformity we have the following equation:

$$RelativeUnifomity_{(\tau_t,\tau_q)} = \frac{UniformInput_{(\tau_t,\tau_q)} + UniformOutput_{(\tau_t,\tau_q)}}{\begin{matrix} UniformInput_{(\tau_t,\tau_q)} \\ + UniformOutput_{(\tau_t,\tau_q)} \\ + NonUniformOutput_{(\tau_t,\tau_q)} \\ + NonUniformInput_{(\tau_t,\tau_q)} \end{matrix}} * 100 \quad (6)$$

The outcome measured from the relative uniformity takes the values in the range [0%, 100%]. For each comparison pair (τ_t, τ_q) the relative uniformity is calculated. This way, for a set of pairs of US-UIDS, the average of the relative uniformity can be used as a quantitative value that represents the general uniformity.

D. Computational implementation of the metrics

The uniformity metrics have been implemented in a computational version[1]. The computational version allows applying the metric in a large set of US-UIDs. The version has been implemented with the paradigm of object-oriented

[1] https://github.com/douglashiura/us-uid

programming and Java programming language. The implementation has been developed to measure the US-UIDs specified in the framework Sc3n4r10 [6]. It handles the US-UIDs in files in JSON format[2]. This way, the files in JSON formats are converted in Java objects and then applied to the metrics.

IV. EVALUATION OF METRICS

To evaluate the metrics, the US-UIDs of four projects are taken into account. In each project, the metrics are applied, and the outcomes compared. The projects are specified in different ways and have a uniformity gap between them. In this section, research issues are also shown and discussed for applicability of the metrics on the projects.

A. Project P1: 8-Puzzle

The 8-puzzle is a game that consists in a grade with three lines and three columns. The grade has a sequence of numbers from 1 to 8 and a blank. The purpose of the game is to start in a random state and put in order the sequence of numbers. This project is comprised by the US-UIDs from Longo and Vilain [4, 5] study that measured the completeness and the correctness of the requirements specified by non-technician users. In the original experiment, fourteen participants specified the winning state of the game. During the specification, each participant should consider at least one move in the grade. Essentially, the US-UIDs specified in this project are duplicated, showing only small differences, as all participants had to specify the same requirement. This way, presumably, the data from the US-UIDs should be 100% uniform.

B. Project P2: Web Application

The web application is a system that has been developed for evaluation and monitoring of the courses chain from e-Tec Brazil[3]. The system contains a database with surveys and evaluation outcomes about Brazilian Federal Education Institutes. A student evasion module has been developed with the specifications of the US-UIDs a priori. For the evasion module, four US-UIDs have been specified. The specification work was developed by two users and two experts in US-UIDs. After being specified, the US-UIDs were automated with tests and reviewed for quality improving, where the best data uniformity was considered. In this project, the US-UIDs were developed with the best correctness and completeness as possible.

C. Projects P3 and P4: Messaging System

The message system is an experiment where the requirements were chosen by the participants themselves. The participants were requested to think about the requirements for an application similar to WhatsApp, Telegram, Hangout or Messenger and, then, specify the requirements as US-UIDs. Two projects were performed in the same experiment. In the project P3, a participant, along with an expert, specified the US-UIDs. The expert reviewed the US-UIDs produced in order to minimize uniformity problems. The project P4 was performed with four participants and without the expert's help. In the project P4, we tried to simulate the situation of shared specification where the irregularity of the data of the US-UIDs occurs. For both of these projects, the controlled factor is the aid from expert. By the practical knowledge, we know that the aid from expert is significantly important for a better quality of the uniformity.

D. Uniformity a priori of the projects

The four projects were selected with characteristics that contribute to a gap in the uniformity of US-UID data among the projects. The unevenness on the uniformity of each project is considered for subsequent evaluation of the sensitivity of the proposed metric. Table II shows the classification a priori of the uniformity of the projects. The classification was made by an expert on US-UIDs. For the classification, the characteristics of each project and the manual review from an expert were considered. The manual identification of the uniformity problems is complex when the amount of US-UIDs increases. Other factor that complicates the manual evaluation is the number of elements belonging to the US-UIDs evaluated. For example, when there are lots of elements as states of interaction in the US-UIDs, the evaluation becomes complex as well. For the manual evaluation of the US-UIDs from an expert, the number of elements should be small, around four US-UIDs, because, above this amount, lots of doubts may be raised for the manual evaluation. The projects were evaluated in a general way by the specialist, i.e., an evaluation was performed to each project.

TABLE II. CLASSIFICATION A PRIORI OF THE PROJECT UNIFORMITY

Project	Classification of the uniformity
P1	Duplication
P2 and P3	Uniformity
P4	Irregularity

The characteristics of the three classes are:

- Duplication: occurs when the same requirement is specified in lots of US-UIDs, although small peculiarities occur in each specification.
- Uniformity: occurs when the diagrams are being specified carefully and taking into account the domain data. In projects P2 and P3, uniformity was maintained with the help of experts.
- Irregularity: occurs when the diagrams are specified by different people and each person considers only the data of their own knowledge and not of the general domain. In project P4, it was observed that uniformity was not maintained since each participant individually specified the US-UIDs and without the expert's help.

[2] https://www.json.org/
[3] http://saas.etec.ufsc.br/

V. RESULTS OF THE METRICS EVALUATION

This section shows the results and the analysis of applicability of the measures of uniformity from projects.

A. Relative Uniformity

The relative uniformity was calculated to all pairs of US-UID for the four projects. Figure 3 shows a graphic with the box plot of the relative uniformity of each project.

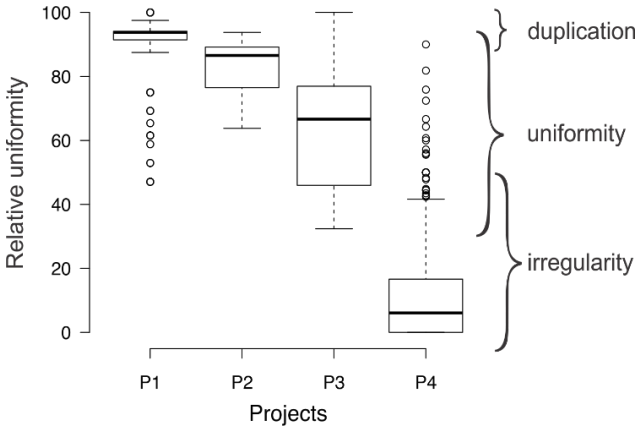

Figure 3. Visual comparison of the distribution of relative uniformity of each pair of US-UIDs for each project

On the right side of the chart there are three intervals for the three classifications of uniformity defined by the expert to each project. The relative uniformity interval for the duplication class is [47.1% to 100%]. The relative uniformity interval for the uniformity class is [32.4% to 100%]. The relative uniformity interval for the irregular class is [0% to 90%]. Ideally, the intervals should be continuous with no overlapping, however, the projects keep the real characteristics, i.e., they are not projects with all controlled data, removing discrepancies. This way, to classify the measures of relative uniformity, it is more suitable to consider a scale of classes of continuous intervals and with no overlapping. The suitable scale can be built using intervals closer to medians, but after analyzing the sensitivity of the metric.

1) Sensitivity of the metrics

The result of the Kruskal-Wallis ANOVA statistical analysis for uniformity measure of the projects shows the statistical measure (H=422.3), with degrees of freedom equal to (3) and probability of significance (P_{value}= 0.0000). This way, for a level of significance of 5% (α=0.05), the test suggests that there is a difference in measures of uniformity between the projects, therefore, the alternative hypothesis (H1) is asserted. Thus, a *post hoc* test is needed to identify among which projects there is a difference in the measure of relative uniformity. Table III shows a comparison of the measures of relative uniformity among the projects.

TABLE III. COMPARISON POST HOC AMONG THE AVERAGES OF RELATIVE UNIFORMITY AS PER THE PROJECTS

Projects	P_{value}	Statistical Decision ($\alpha = 0.05$)
P1 x P2	1.0000	H0: Insensitive
P1 x P3	0.1265	H0: Insensitive
P1 x P4	0.0000	H1: Sensitive
P2 x P3	1.0000	H0: Insensitive
P2 x P4	0.0000	H1: Sensitive
P3 x P4	0.0000	H1: Sensitive

The statistical analysis suggests that there is no significant difference among the measures of uniformity in the projects P1, P2 and P3, but there is a significant difference for the measures of relative uniformity between the projects P1 and P4, P2 and P4 and P3 and P4. This way, we can notice that the classes of duplication and uniformity that are classified manually by the expert are not sensitive to the metric, i.e., the metric is not able to classify between duplication and uniformity. However, the metric is sensitive and can classify among the classes of uniformity and irregularity as per the evaluation from expert. For the projects P2 and P3 that are classified as uniform, the metric is insensitive, i.e., the metric does not measure differences, because, in fact, there are no significant differences in the measure of relative uniformity between the projects P2 and P3.

B. Absolut Uniformity and Irregularity

The absolute uniformity is the count of the pairs of user inputs and system outputs sectioned in uniform and irregulars. Figure 4 shows the box plots for visual comparison of the absolute uniformity and irregularity of each project. The first characteristic that we can notice is that the amount of pairs of system outputs is greater than the amount of pairs of user inputs in all projects. This first characteristic is compatible with the fact that in the US-UIDs there are more system outputs than user inputs. The second characteristic is that there are more uniform system outputs in the projects that were classified as duplicate and uniform (P1, P2 and P3). Adversely, for the project classified as irregular (P4), there are more irregular system outputs. The third characteristic that we can notice is that there are less uniform user inputs than irregular ones in all projects. However, this third characteristic is less accentuated in project P3 if compared with the project P4, where we can conclude that the system outputs also influence the uniformity, but it is more difficult to analyze and keep the uniformity during the specification of the US-UIDs. The fourth characteristic is about the influence on the automation process of the tests in the uniformity. During the test automation process, US-UIDs are reviewed to improve quality. Project P2 considers this process. Also, project P2 has more uniform system outputs than others projects. In practice, the testing automation process is done by the programmers and, in this process, with the help from the stakeholders, the US-UIDs are corrected and implemented. In general, programmers are more rigorous with the review process and tend to review US-UIDs to improve uniformity.

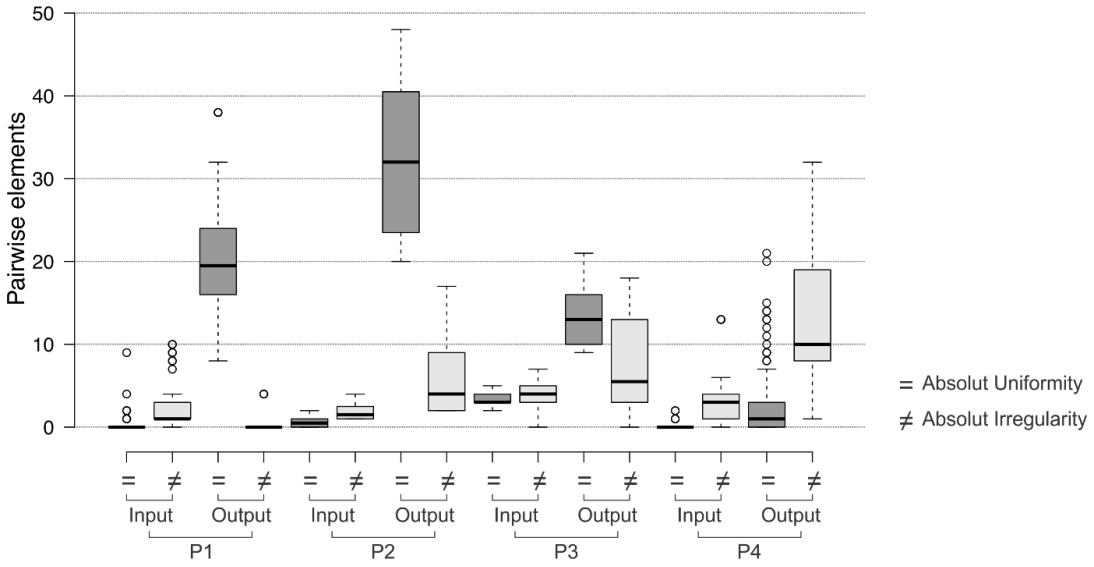

Figure 4. Visual comparison of the absolute uniformity of the user inputs and system output.

VI. CONCLUSIONS

This paper shows three metrics for measuring data uniformity of the US-UIDs. The metric of relative uniformity, based on metrics of uniformity and absolute irregularity, is important for measuring and classifying the US-UIDs. The metric takes values from measure of uniformity in the interval [0% to 100%], where 0% is irregular and 100% is uniform.

In order to evaluate the metric, four empirically pre-evaluated projects were used by an expert. The evaluation from expert considered three classes for classifying the uniformity: duplication, uniformity, irregularity. With the results of the applicability of the metrics, it was concluded that, through the measure of relative uniformity, only two classes are suitable. The metric of relative uniformity is not sensitive for three classes of the projects. Based on the results, the suitable classes are uniformity and irregularity. The class of irregularity takes the interval [0% to 45%] from the measure of relative uniformity. The class of uniformity takes the interval [45% to 100%] from the measure of relative uniformity. The value 45% of relative uniformity is the boundary between both classes, but in both projects classified as uniform and irregular, a measure overlapping has occurred, where this point was arbitrary defined as the most suitable for boundary between both classes. So, this classification is suitable to evaluate the US-UIDs during the process of specification and reviewing them, if necessary, before starting the testing automation process.

The measures of absolute uniformity and irregularity are complementary and can be used to evaluate and compare the types of elements in the US-UIDs. The elements of type of system outputs are more present in the US-UIDs and are also more uniform, however, it was unable to create a classification like the relative uniformity one. However, through the projects, the elements of system outputs have weights of uniformity different than the user inputs.

With the specifications of uniform data of the US-UIDs, it's expected avoid reworking and improve the communication between users and developers. However, the evaluation of quality criteria not always has significant results in practice [8], so, it also should be investigated how to apply the metrics during the specification of the US-UIDs and how to guide the users in uniformity troubleshooting.

The main contributions of this work are the proposed metrics, the computational implementation of the metrics and the evaluation of four projects. Moreover, the US-UIDs used in the evaluation are available (https://github.com/douglashiura/us-uid-uniformity) for future investigations of this acceptance testing format for the software development.

REFERENCES

[1] BECK, Kent, "Test-driven development: by example," Addison-Wesley Professional, 2003.

[2] Gärtner, Markus, "ATDD by example: a practical guide to acceptance test-driven development," Addison-Wesley, 2012.

[3] Hendrickson, Elisabeth, "Driving development with tests: ATDD and TDD," STARWest 2008, 2008.

[4] Longo, Douglas Hiura, and Patrícia Vilain, "Creating User Scenarios through User Interaction Diagrams by Non-Technical Customers," SEKE. 2015, pp. 330-335.

[5] Longo, Douglas Hiura, and Patricia Vilain, "User scenarios through user interaction diagrams," International Journal of Software Engineering and Knowledge Engineering 25.09n10, 2015, pp.1771-1775.

[6] Longo, D. H., Vilain, P., da Silva, L. P., & Mello, R. D. S, "A web framework for test automation: user scenarios through user interaction diagrams," In Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services. ACM, 2016 pp. 458-467.

[7] Vilain, P., Schwabe, D., de Souza, C., "A diagrammatic tool for representing user interaction in UML," <<UML>> 2000- The Unified Modeling Language. Springer, 2000, pp.133-147.

[8] Lucassen, G., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S., "Improving user story practice with the Grimm Method: A multiple case study in the software industry," In International Working Conference on Requirements Engineering: Foundation for Software Quality.). Springer, Cham, 2017, pp. 235-252.