Exploring User-centered Requirements Validation and Verification Techniques in a Social Inclusion Context

Emille Catarine Rodrigues Cançado[a], Ian Nery Bandeira[b], Pedro Henrique Teixeira Costa[c], José Fortes Neto, Daniel Carvalho Moreira, Luís Amaral[d] and Edna Dias Canedo[e]

Department of Computer Science, University of Brasília (UnB), P.O. Box 4466, Brasília, DF, Zip-cod 70910-900, Brazil

Keywords: Former Inmates, Social Inclusion, Requirements Verification and Validation, Privacy, Prototyping.

Abstract: The activities that comprise a requirements engineering process involve elicitation, modeling, validation, and verification of requirements, and these activities tend to be more communication and interaction-intensive than others during the software development process. This paper presents an experience report on requirements validation and verification techniques applied to a mobile application project developed for the Brazilian prison system's former inmates, aiming to support them in their resocialization process. Besides, it presents the decisions we made in agreement with the project stakeholders to guarantee the end-users data privacy. Our results show that even with the Covid-19 pandemic and social isolation restrictions, it was possible to apply the requirements validation techniques. Furthermore, the mobile application's acceptance tests with both stakeholders and the end-users demonstrate that the developers duly followed the privacy guidelines. Finally, all privacy requirements comply with the stakeholders and the application's end-users needs and are under the Brazilian General Data Protection Law (LGPD).

1 INTRODUCTION

A software's set of requirements must be validated to ensure that the stakeholders' needs are being addressed and that the defined quality criteria are being fulfilled (Lampa et al., 2017). Therefore, it is necessary to validate and verify the requirements to detect any errors or inconsistencies in the team's elicited and documented requirements for this activity (Bjarnason et al., 2014). Software requirements validation encompasses validation and verification activities (Nascimento et al., 2021). Verification is perceived as sufficing the precepts of correction and completeness of requirements, while validation is understood as meeting users' expectations (Fiorini et al., 1998; do Prado Leite and Freeman, 1991).

Validation and verification of elicited requirements are related to the quality of the software. Thus, as software systems become more complex, concerns about validating and verifying requirements increase since it is necessary to ensure that each error has been identified and corrected (McMillan et al., 2019). Therefore, the non-alignment of requirements engineering with verification and validation can lead to software deployment problems regarding the deadline, quality, and scope (Bjarnason et al., 2014).

The legal requirements of software also need to be verified and validated (Santos et al., 2017), especially in the current context. There is specific legislation to guarantee users' data privacy. In Brazil, the General Data Protection Law (LGPD) came into force in 2020, which establishes the rules for collecting, storage, treatment, and sharing personal data, imposing more protection and penalties for non-compliance with the principles of the law by organizations (Macedo, 2018; BRASIL, 2020). The LGPD covers all personal data collected, stored, and processed by public and private organizations with an international reach (Bax and Barbosa, 2020; Ferrão et al., 2021; Canedo et al., 2021; Canedo et al., 2020).

This paper presents a case study of requirements validation and verification techniques applied in software development to support the Brazilian prison system's former inmates. As it is a more vulnerable

[a] https://orcid.org/0000-0002-1283-8391
[b] https://orcid.org/0000-0002-8083-7278
[c] https://orcid.org/0000-0002-8385-8314
[d] https://orcid.org/0000-0003-1236-3119
[e] https://orcid.org/0000-0002-2159-339X

Cançado, E., Bandeira, I., Costa, P., Neto, J., Moreira, D., Amaral, L. and Canedo, E.
Exploring User-centered Requirements Validation and Verification Techniques in a Social Inclusion Context.
DOI: 10.5220/0010959900003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 1*, pages 85-92
ISBN: 978-989-758-569-2; ISSN: 2184-4992

group, our choices related to data privacy were reshaped so as not to be intimidating to the end-users, for instance, not saving end-users personal information and creating the possibility for former inmates to use all the application services, even if one chooses not to permit any of their device's native tools in the app, such as its location service. We also analyzed the impact of using these techniques on an actual software project amidst the period of social isolation caused by the Covid-19 pandemic.

2 VALIDATION AND VERIFICATION TECHNIQUES

The requirements validation and verification process consist of reviewing, analyzing, and testing the requirements to ensure that the designed system is compatible with the users' needs (Laplante, 2017). Several techniques in the literature address the validation and verification of requirements (Bertini et al., 2006; Kalinowski et al., 2007; Maalem and Zarour, 2016; Bhatti et al., 2015; Pohl, 2016):

1. **Review by a Specialist** – This technique is based on using a specialist's knowledge to check the quality of the requirements. The specialist reviews all requirements and looks for problems such as inconsistency or ambiguity. When the reviewer finds a problem, he/she comments on the documentation for errors and possible solutions to fix them.
2. **Inspection of Requirements** – This technique consists of a team with different responsibilities (Pohl, 2016): **Organizer:** Plans and controls the inspection process; **Moderator:** Lead the sessions, always maintaining the inspection follow-up as agreed in the initial stages; **Author:** Responsible for explaining the requirements to inspectors in the early stages of the project. The author is also responsible for correcting the errors identified in the process's steps; **Reader:** Member of the team that will guide inspectors in the face of the various requirements to be inspected. The reader's job is to take the inspector's responsibility away from interpreting what the author meant, and they can focus on the requirement itself; **Inspectors:** Responsible for detecting faults and reporting them to other members of the project team; **Secretary:** Responsible for taking the results of the session and compiling them, in addition to preparing the minutes of the session. Inspection within the context of requirements validation can be divided into the stages of planning, overview, fault detection, and fault collection: (Pohl, 2016):
   (a) **Planning:** In this step, one defines the objective of the inspection, the planning of the process, and the functions of each participant;
   (b) **Overview:** In this phase, the author details all the requirements for his/her team so that everyone is aware of the system in question and can clarify any doubts;
   (c) **Fault Detection:** The inspectors effectively examine the requirements and document the possible faults for the next step. Fault finding can be done either as a team or individually. The group search has the advantage of communication and consequent synergy between inspectors whilst the individual search makes everyone focused on the work;
   (d) **Fault Collection:** In this last step, all faults presented in the detection stage are reevaluated to avoid duplication or to remove faults classified incorrectly.
3. **Walkthrough** – It is a technique similar to inspection, but it is a more straightforward and less rigid version (Pohl, 2016). The division of the team is simplified, in which the same person can perform several functions. In this technique, the functions of at least one reviewer, author, secretary, and moderator are used. This technique aims to identify flaws in the requirements and share the project's understanding among the people involved in the inspection.
4. **Verification in Different Perspectives** – This technique validates the requirements through an interpretation of the system, taking into account different perspectives. The perspectives usually vary from system to system, but it is possible to elicit at least some types that usually occur in every project, and that can be adapted:
   (a) **Perspectives of Stakeholders** – User perspective: The user checks the requirements from his/her perspective and checks whether it matches the reality and expected quality; Developer perspective: The developer verifies that the requirements contain all the information and properties necessary for the project to be completed; Test perspective: The tester verifies whether it is possible to make test cases that satisfy the entire project based on the elicited requirements.
   (b) **Perspectives of System Quality** – Documentation perspective: From this perspective, it is ensured that the documentation

involving the requirements meets all the criteria that have been defined;

Content perspective: Perspective that deals with the content of the requirements itself and focuses on the quality of that content;

Perspective of agreement: In this perspective, the focus is on ensuring that everyone involved in the project is following the elicited requirements and that there is no conflict to be resolved.

5. **Prototyping Validation** – Prototype is a preliminary model of a complete system. Thus, what this technique does is to use this model in favor of requirements validation. Based on the project prototype, requirements can be tested and validated practically by the stakeholders, checking if they meet the expectations. Prototypes can be divided depending on their destination in disposable prototypes or evolutionary prototypes (Pohl, 2016). The disposables are only for the current test. They will not be put into production again since the evolutionary is a constantly evolving prototype as the project develops and requires more work to be done. For a prototype to be made, it is necessary to select the requirements that it will contemplate. This set does not typically cover all documented requirements as it would bring very high costs and complexities. Therefore, it is necessary to make a selection criterion among the requirements and determine the most suitable to be used in the prototype. Along with the prototype itself, three more documents are needed to validate the requirements (Pohl, 2016): a manual containing all the usage and application information; a document containing several scenarios for validating requirements; and a checklist for each of the requirements that will be validated. Scenarios that are not included in the documents can and should be tested after the entire process to verify the system in different situations and circumstances that were not planned initially. After the validation is completed, the results are compiled and discussed. Changing, removing, or adding requirements are consequences during this step. If the changes are very large or impactful, it is advisable to remodel the prototype to cover these changes, and it is possible to carry out a new validation (Pohl, 2016).
6. **Checklist Validation** – The checklist technique is used in complex projects and when the system has several issues that need to be taken into account. Checklists can be used in conjunction with many other techniques and are made up of questions or statements to identify errors in the requirements. The source for assembling the checklist can come from both stakeholders and requirements quality criteria. The list can constantly be updated as the project progresses. New features and requirements emerge, so removing ambiguous questions, editing, and adding information are always welcome as the project develops. The success of the technique will depend on the length and complexity of the checklist. A very long and complete list of details can make the process slower and unfeasible for the inspector. The same thing can happen if the list is very subjective, making the validation process slow and susceptible to failures.

3 SOFTWARE REQUIREMENTS

The Virtual Social Office (ESVirtual) purpose is to facilitate access to services and support the former inmates and their families (de Mendonça et al., 2020) The mobile application provides access to information on services and policies for assisting people from the Brazilian prison system by giving opportunities to take online courses, providing job advice, information about Health Care, proof of compliance with sentence conditionalities for people on semi-open and open prison regimes, among other features.

Among the requirements validation and verification techniques presented in Section 2, we used verification from different perspectives and validation by prototyping in software development in this research. In an effort to ensure that every possible necessity of both the public and the creators of the Virtual Social Office would be met, we opted to use verification from different perspectives. Along with the development, testing, and design leads, we analyzed the perspective of three different entities:

1. The Brazilian National Council of Justice (CNJ) administrative staff, who have overall knowledge of the resocialization process and which, despite not working directly with the app's target audience, have facilitating expertise in devising tools for the final product's deployment and dissemination;
2. Employees of the physical social office, who directly serve the app's target audience, since the physical social offices already provide assistance to former inmates in their resocialization process;
3. Potential end-users, who have experiential knowledge of their particular processes and might possess insight into critical issues and features not addressed by the other two parties.

Validation by prototyping was employed in an attempt to ensure ease of communication between the different project stakeholders since suggestions could be better understood and debated when embodied and evaluated in a prototype. In addition, prototyping might facilitate the communication process between the project's design and development teams to prevent the idealization of any functionality that is not feasible for development.

3.1 Verification in Different Perspectives

In the ESVirtual development, we carried out the validation and verification of requirements from the stakeholders' perspectives and the system's quality. Regarding the stakeholders' perspectives, those responsible for the project are Brazilian National Justice Council administrative staff, who regard this application as a tool to support the re-socialization process of former inmates. These executives have selected some physical Social Office members to validate the requirements along with three potential end-users.

The stakeholders' and users' validation was performed through online meetings. The project lead designer presented to the meeting participants the prototypes proposed for the application's functionalities. The validation and verification meetings were held one week after eliciting requirements for the respective functionality. In case an error was detected in the understanding of the requirement, we recorded the observations. In the following week, we presented the proposed prototype's updates to the stakeholders until there were a consensus and the validation of the requirement. Thus, the requirement was only forwarded to the software development team after its validation by the stakeholders.

From the test and developers' perspectives, we tested the functionalities using integrated technologies through programs that simulate the application's use from a user's perspective. We created automatic actions on the screen using command lines, and through the answers presented, it was possible to carry out a general analysis. Thus, we expected responsive activities on the screen according to the sequence of pre-established actions.

We could analyze the application's response time, the availability of features, and errors that may arise without prior knowledge from the automated tests. Also, these tests could reveal bugs in the features resulting in a different procedure than expected, thus predicting future errors that could come to the end-user of the application. We tested all possible actions related to the application's functionality to expose, predict and correct possible errors that could occur through malfunction behavior. Error prevention allows preventing failures from remaining in the application and find possible improvements for the user experience. We use some tools to perform this type of tests, namely: Cucumber (Li et al., 2016; Fazzolino et al., 2018), tool developed in the Java language, used to describe all the scenarios and steps that were simulated in the application.

The Appium (Alotaibi and Qureshi, 2017), technology responsible for integrating other tools into the application and executing the steps defined in Cucumber, promoting the communication and execution of the commands to be executed on the screen automatically. For the most part, the operation of Cucumber occurs through references that are pre-labeled in the front-end project through identifying each feature that has an assigned reference name. It allows localization on screen and the necessary action of the feature to obtain the application's expected response. In addition to the identification property, others such as positions of elements using height and width references on the screen and the response time were considered in the execution of the test sessions.

We used Cucumber to describe the steps that, theoretically, would be performed by an actual user of the application, an emulator, or a physical device. The Cucumber was responsible for running the application, and the Appium tool was responsible for making the communication between Cucumber and the application on the device. Figure 1 shows one of the tests performed to verify the functioning of the functionality required for health facilities. This requirement is responsible for providing information on basic health units, emergency services, psychosocial care centers, and hospitals to former inmates from the prison system. We performed all the steps described in Cucumber in an actual use scenario. Besides, we corrected the found errors before the implementation phase, and the specifications of the Virtual Social Office requirements were updated.

```
#language: en
#@appium

Feature: Check the operation of screens related to health establishments

  Scenario: Access to "<ScreenTitle>"
    Given that I access the app and authorize the location of the device
    And I allow the GPS to be activated
    And I am at the "START" screen of the aplication
    When I click on the "Health" referent section
    Then I should be directed to the "Health" sector
    And the screen title should be "Health"
    When I scroll down the screen
    Then it should have a text about the difference in services and a button to understand more
    And it should have establishments in the health area to be clicked
    When I click the button for "<Button>"
    Then I should be directed to the "<ScreenTitle>" screen
    And the screen title should be "<ScreenTitle>"

    Examples:
    | Button                      | ScreenTitle                 |
    | BASIC HEALTH UNITS          | Basic Health Units          |
    | HOSPITALS                   | Hospitals                   |
    | PSYCHOSOCIAL CARE CENTER    | Psychosocial Care Center    |
    | EMERGENCY CARE UNITS        | Emergency care units        |
```

Figure 1: Verification of Health Facilities requirements.

The perspective of the system's quality was realized through the content, in which three researchers of the project initially validated the screens of the proto-

type, developed from the elicited requirements. Subsequently, we validated the prototypes again with the project stakeholders to guarantee that all requirements were in line with expectations and that there was no conflict between the decisions adopted.

Figure 2 (a) presents the proposed prototype for the specified requirement of the Health Facilities functionality, also validated through the testing perspective (Figure 1). All inconsistencies/errors detected in the verification from different perspectives were corrected, and we implemented suggestions for improvements in the prototypes before the next phase of the software development process.

4 PROTOTYPING VALIDATION

The prototyping of the requirements can be used as a validation technique, identifying the problems in the specification of requirements from a visual perspective of the software under development (Bilal et al., 2016). Aceituna et al. (Aceituna et al., 2011) stated that prototyping could prevent more errors than requirements validated in natural language. Prototyping allows us to include non-technical stakeholders in the inspection process and validate, with less time and cost, the hardware and software requirements of a system (Aceituna et al., 2011).

We prototyped all the ESVirtual features to validate with the stakeholders before going to the implementation phase. First, the elicitation of requirements occurred in face-to-face meetings at the National Council of Justice headquarters. Subsequently, due to the Covid-19 pandemic, virtual meetings were held using the Microsoft Teams tool [1]. We choose this tool because both institutions involved in the development of the project (National Council of Justice (CNJ) and University of Brasília (UnB)) adopted it.

Requirements elicitation meetings were held weekly with various project stakeholders and researchers and lasted an average of 02 hours. During the meetings, the functionalities were discussed, and the stakeholders passed on the necessary information. Then the researcher Designer of the project later performed the prototyping of the ESVirtual's functionalities. We prototyped all the functionalities of the application using the tool Figma (Figma, 2021).

We carefully reviewed all functionalities with the stakeholders in the prototyping phase, aiming to provide a solution according to the former inmates' needs and with a friendly interface because it is an application that serves a fragile audience. After the prototyping of the functionalities, we presented the newly proposed solution to the stakeholders. During the presentation, the meeting participants discuss doubts, and in some situations, there were suggestions for improvements and reformulation of the proposed solution. These reviews allow the ESVirtual to be an ally to the end-user searching for important information for their re-socialization process.

As an example, it is worth mentioning the proposed solution for the requirement " Food ". The purpose of this functionality in the ESVirtual application is to indicate to the user (former inmates) food facilities to find meals at a popular price. If the user permits the application to access their location, they will have access to a list of places that offer food, ordered by proximity, as shown in Figure 2 (b). Ordering by proximity helps them to take the shortest route/path to reach a community restaurant.

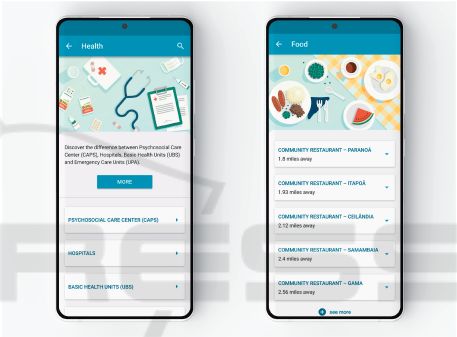

Figure 2: (a) Validation of the prototype of the health facilities requirements and (b) "Food" functionality with access to the user's location.

However, sharing the location can be sensitive information for many users. Even if the application does not save the location information, some may choose not to provide this data. In this way, if the users choose the application not to access their location, they can indicate the state and city to consult and access the complete list of community restaurants in alphabetical order. Figure 3 presents the proposed prototype for the requirement specified for this functionality. Although the user does not know the distance of these restaurants concerning his current location, he is not deprived of accessing the application's information.

All errors found in the validation activity in the requirements elicitation phase were corrected, and we implemented some suggestions for improvements in the functionalities. Thus, we can conclude that using the techniques of validation and verification of requirements was positive since all users could validate the requirements elicited through different perspectives and the prototypes proposed for each func-

[1] https://teams.microsoft.com/

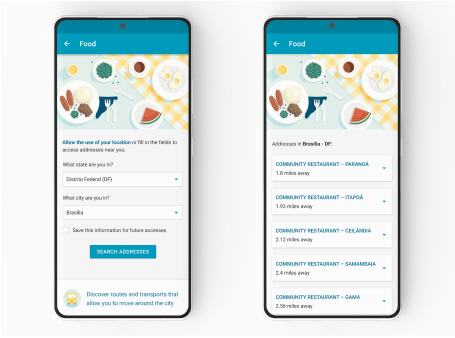

Figure 3: "Food" functionality without access to the user's location.

tionality of the application.

In addition to using the techniques to perform the validation and verification of requirements in the initial phases of the project, we also carried out a validation with the project stakeholders after implementing all software requirements. The stakeholders and users performed the validation by downloading the application from Play Store stores (for Android platform users) and Apple Store stores (for iOS users). Participants in the validation activity performed all the features provided and made available in the ESVirtual. We did not find impacting errors that could harm the evolution of the ESVirtual. Thus, we can conclude that the verifications and validations before the implementation phase were essential for the delivered project's quality.

5 DATA PRIVACY

We developed the ESVirtual for a vulnerable population that seeks to re-socialize in society. Thus, we elicited the software requirements to keep the end-users' information confidential to avoid embarrassment on the former inmate or situations considered difficult and/or embarrassing by them. Therefore, the requirements elicitation ensure that users' data were not requested and maintained when using the application's functionalities. For this reason, according to the ESVirtual specification, the system developed will only attend queries to public data provided by different government agencies. There is no need to keep a register with users' information since they carry out the consultations anonymously to public data. As the system does not store user information, no actions were necessary to guarantee the user's data privacy. The functionality called "Legal Situation" allows the former inmates to monitor their processes' execution through a private consultation using a personal access key.

The project development team opted to use the POST request method, from the HTTP (Tanenbaum and Wetherall, 2011) protocol, to pass the query key to the processes so that this data was not part of the query URL. Besides, the HTTPS protocol was used, which implements the HTTP protocol with an additional layer of security over Secure Socket Layer (SSL), so that all requests are encrypted, thus guaranteeing the security of the execution of the functionality. The prototype of the "Legal Situation" functionality is shown in Figure 4.

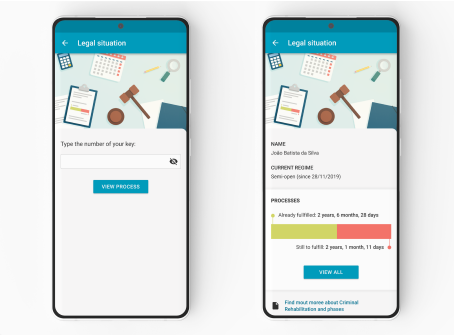

Figure 4: "Legal Situation" functionality.

6 DISCUSSIONS

The Virtual Social Office application's proposed requirements stemmed from discussions and interviews held with the National Council of Justice (CNJ) stakeholders, prison system former inmates, and Social Office of Espírito Santo employees. These requirements guided the layout and usability solutions proposed by the project designer, and the prototyping of the solutions on the Figma Design Tool (Figma, 2021) facilitated the process of monitoring and validation of the entire team and stakeholders, and through the visual concept rendered by the prototype, stimulated further discussions involving the gathering of new requirements and the validation of requirements already proposed.

Using the validation technique involving multiple perspectives enabled all parties involved in the project to validate the requirements, whether from the stakeholders', end-users', developers', or project researchers' perspectives. The physical Social Office employees and the former inmates who participated in our meetings fulfilled essential development roles. Their inputs and insights, which are from people closely intertwined with the re-socialization process, aligned the development and design teams for features that effectively reach the application's target audience. Such interaction among the contributors allowed for interesting exchanges related to user expe-

rience, solution usability, and possible technologies which might be employed to support the application's acceptance by the end-users.

The validation by prototyping proved to be highly adequate and enabled the project's design team's involvement in the software release schedule and the requirements prioritization. The prototypes were submitted to the Scrum master and the project development team right after the stakeholder validation to ensure that any concerns and features were explained to the team. Each solution proposed by the design team is discussed with the project's lead engineer to ensure the proposed approach's feasibility. Errors found during the validation and requirements verification phase and after the application was published online were negligible and could be addressed quickly. Furthermore, the use of prototypes proved to be extremely important for the idealization of the application's functionalities by the entire team during the Covid-19 pandemic, in which only non-presential meetings were held. This graphic resource allowed easier comprehension of all the proposed ideas, filtered and represented by the design team, and then discussed, reviewed, and validated by the entire team.

This process, along with the interaction between the project members, reduced the risks of errors, usability failures, or wrongly implemented requirements by the development team. Besides, the finished prototypes were of high graphic quality, facilitating the appraisal from every member involved and accurately representing every detail of what was expected as an outcome from the development team. As a result, this eased the entire process of implementing new features. Also due to the social isolation period resulting from the Covid-19 pandemic, meetings between the research team and the stakeholders became more limited. This limitation resulted in the research team having to make critical decisions to develop the app without the stakeholders' direct involvement. The prototyping and requirements validation enabled all such decisions to be easily identified, understood, and validated before the app's release in the stores.

Furthermore, the usability tests with end-users were also impaired due to social isolation, especially among sensitive groups with low education levels or little to no familiarity with mobile devices. However, the app's alpha version allowed virtual usability tests with three former inmates, which were able to consolidate the proposed solutions and point out not only constructive criticism towards what had already been implemented but also requirements that they deemed relevant and that had not been addressed in the app so far.

7 CONCLUSIONS

In this paper, we provide an experience report on the use of software requirements validation and verification techniques applied to real-world software development. We conducted these processes using verification from different perspectives and validation by prototyping. These results have shown how the techniques have mitigated the likelihood of errors during the requirements engineering process. Furthermore, performing the validation made it possible to pinpoint possible errors in advance of the beta testing period, as all members involved could assess the proposed solutions in both functioning prototypes and the application's alpha version.

The requirements validation also allowed the stakeholders and end-users to ensure that the privacy of the user's data was being guaranteed in the development process, as per the requirements elicitation performed, and whether it complied with the LGPD standards. For future projects, we intend to work towards the requirements elicitation for new application functionalities based on interviews and discussions with the CNJ stakeholders, former inmates of the prison system, and employees of physical Social Offices across the country. New tests will also be conducted with stakeholders and former inmates, hopefully with more participants with varying age, education, and gender profiles. Also, we will perform a usability evaluation on the ESVirtual application employing the set of usability heuristics for quality assessment of mobile applications on smartphones, proposed by Ruyther et al. (da Costa et al., 2019). Simultaneously, we will investigate the application's impact on the re-socialization process of Brazilian prison system former inmates.

ACKNOWLEDGMENTS

We want to thank the National Council of Justice for supporting this research. This work has been partially supported by FAP-DF (the Brazilian Federal District Research Foundation).

REFERENCES

Aceituna, D., Do, H., and Lee, S. (2011). Interactive requirements validation for reactive systems through virtual requirements prototype. In *First Model-Driven Requirements Engineering Workshop, MoDRE 2011, Trento, Italy, August 29, 2011*, pages 1–10. IEEE Computer Society.

Alotaibi, A. A. and Qureshi, R. J. (2017). Novel framework for automation testing of mobile applications using appium. *International Journal of Modern Education & Computer Science*, 9(2).

Bax, M. P. and Barbosa, J. L. S. (2020). Proposta de mecanismo de consentimento na lei geral de proteção a dados - LGPD (consent mechanism proposal in LGPD). In *ONTOBRAS*, volume 2728 of *CEUR Workshop Proceedings*, pages 316–321, http://ceur-ws.org/Vol-2728/doctorate4.pdf. CEUR-WS.org.

Bertini, L. A., Kirner, T. G., Montebelo, M. I., and Lara, I. A. R. (2006). Técnicas de inspeção de documentos de requisitos de software: um estudo comparativo. In Alencar, F. M. R., Sánchez, J., and Werneck, V., editors, *Workshop on Requirements Engineering (WER), Rio de Janeiro, RJ, Brasil, Julho 13-14, 2006*, pages 67–74.

Bhatti, S. N., Usman, M., and Jadi, A. A. (2015). Validation to the requirement elicitation framework via metrics. *ACM SIGSOFT Software Engineering Notes*, 40(5):1–7.

Bilal, H. A., Ilyas, M., Tariq, Q., and Hummayun, M. (2016). Requirements validation techniques: An empirical study. *International Journal of Computer Applications*, 148(14).

Bjarnason, E., Runeson, P., Borg, M., Unterkalmsteiner, M., Engström, E., Regnell, B., Sabaliauskaite, G., Loconsole, A., Gorschek, T., and Feldt, R. (2014). Challenges and practices in aligning requirements with verification and validation: a case study of six companies. *Empir. Softw. Eng.*, 19(6):1809–1855.

BRASIL (2020). Guia de boas práticas – lei geral de proteção de dados (lgpd). *Comitê Central de Governança de Dados. Secretaria de Governo Digital*, 1–65.

Canedo, E. D., Calazans, A. T. S., Masson, E. T. S., Costa, P. H. T., and Lima, F. (2020). Perceptions of ICT practitioners regarding software privacy. *Entropy*, 22(4):429.

Canedo, E. D., Cerqueira, A. J., Gravina, R. M., Ribeiro, V. C., Camões, R., dos Reis, V. E., de Mendonça, F. L. L., and de Sousa Jr., R. T. (2021). Proposal of an implementation process for the brazilian general data protection law (LGPD). In *ICEIS (1)*, pages 19–30. SCITEPRESS.

da Costa, R. P., Canedo, E. D., de Sousa Júnior, R. T., de Oliveira Albuquerque, R., and García-Villalba, L. J. (2019). Set of usability heuristics for quality assessment of mobile applications on smartphones. *IEEE Access*, 7:116145–116161.

de Mendonça, W. L. M., Costa, P. H. T., Cançado, E. C. R., Lima, F., Canedo, E. D., Bonifácio, R., and Amaral, L. H. V. (2020). From dusk till dawn: Reflections on the impact of COVID-19 on the development practices of a r&d project. In *SBES*, pages 596–605. ACM.

do Prado Leite, J. C. S. and Freeman, P. (1991). Requirements validation through viewpoint resolution. *IEEE Trans. Software Eng.*, 17(12):1253–1269.

Fazzolino, R., de Faria, H. M., Amaral, L. H. V., Canedo, E. D., Rodrigues, G. N., and Bonifácio, R. (2018). Assessing agile testing practices for enterprise systems: A survey approach. In *SAST*, pages 29–38. ACM.

Ferrão, S. É. R., Carvalho, A. P., Canedo, E. D., Mota, A. P. B., Costa, P. H. T., and Cerqueira, A. J. (2021). Diagnostic of data processing by brazilian organizations - A low compliance issue. *Inf.*, 12(4):168.

Figma (2021). The collaborative interface design tool. *Figma, accessed on march 5, 2021*.

Fiorini, S. T., do Prado Leite, J. C. S., and de Lucena, C. J. P. (1998). Organizando processos de requisitos. In *Workshop on Requirements Engineering(WER)*, pages 1–8.

Kalinowski, M., Spínola, R. O., Dias-Neto, A., Bott, A., and Travassos, G. H. (2007). Inspeções de requisitos de software em desenvolvimento incremental: Uma experiência prática. *VI Simpósio Brasileiro de Qualidade Software (SBQS), Porto de Galinhas, Brazil*.

Lampa, I. L., de Godoi Contessoto, A., Amorim, A. R., Zafalon, G. F. D., Valêncio, C. R., and de Souza, R. C. G. (2017). Project scope management: A strategy oriented to the requirements engineering. In *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 2: ICEIS,*, pages 370–378. INSTICC, SciTePress.

Laplante, P. A. (2017). *Requirements engineering for software and systems*. CRC Press.

Li, N., Escalona, A., and Kamal, T. (2016). Skyfire: Model-based testing with cucumber. In *ICST*, pages 393–400. IEEE Computer Society.

Maalem, S. and Zarour, N. (2016). Challenge of validation in requirements engineering. *J. Innov. Digit. Ecosyst.*, 3(1):15–21.

Macedo, P. N. (2018). Brazilian general data protection law (lgpd). *Brazilian National, accessed on January 18, 2021*, 1(1):1–16.

McMillan, C., Crapo, A., Durling, M., Li, M., Moitra, A., Manolios, P., Stephens, M., and Russell, D. (2019). Increasing development assurance for system and software development with validation and verification using assert™. Technical Report 2019-01-1370, SAE Technical Paper.

Nascimento, R., Souza, L., Targino, P., Sizílio, G., Kulesza, U., and Lucena, M. (2021). Investigating information about software requirements in projects that use continuous integration or not: An exploratory study. In *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 2: ICEIS,*, pages 303–312. INSTICC, SciTePress.

Pohl, K. (2016). *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant*. Rocky Nook, Inc.

Santos, I., Miranda, E., and Lucena, M. (2017). Rastreamento e gerenciamento de requisitos em busca da conformidade legal. In *Workshop on Requirements Engineering(WER)*.

Tanenbaum, A. S. and Wetherall, D. (2011). *Computer Networks*. Prentice hall, 5 edition.