Development, reuse, and repurposing of software artifacts in Digital Citizen Science. Are we reinventing the wheel?

Alejandra B. Lliteras[1,2], Diego Torres[1,2,3], César A. Collazos[4], Alejandro Fernandez[1,2]

[1] UNLP, Facultad de Informática, LIFIA. La Plata, Buenos Aires, Argentina.

[2] CICPBA, Buenos Aires, Argentina.

[3] UNQ, Dto. CyT.

[4] IDIS research group, Universidad del Cauca-Colombia.

{alejandra.lliteras, diego.torres, alejandro.fernandez }@lifia.info.unlp.edu.ar

ccollazo@unicauca.edu.co

Abstract. In the production of software artifacts, it is possible to start from scratch, reuse existing artifacts, or even repurpose artifacts produced with another purpose in mind. As an application domain matures, often developing from scratch and repurposing it leads to reuse. Reuse not only reduces time and costs but also acts as a mechanism to encapsulate and disseminate the knowledge of domain experts. With software being a central ingredient to mediate the participation of volunteers in digital citizen science, it would be expected to observe various developments with reusable devices. However, reuse is rare today. Through a systematic review, we study the software production strategies reported during the last decade in citizen science projects. We observe that there is still a high amount of development from scratch, so we open the debate on the usefulness of designing reuse processes focused on *reusers* to promote this strategy.

Keywords: Software Engineering, Development, Reuse, Repurpose, Digital Citizen Science

1 Introduction

Citizen Science is the way to carry out research projects involving "volunteers", "citizens" or "citizen scientists" as an important part of these projects, as indicated in [1]. Volunteers involved in Citizen Science projects form a community and carry out tasks that could not be carried out solely by experts or through computational methods [2]. In this way, human intelligence is intertwined with the resolving power of

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2747/paper16.pdf

computers. Although the participation of volunteers in science is not a new practice, it is the participation on the scale and in the forms proposed by modern Citizen Science with the support of ICTs that makes it interesting. Volunteers may have different motivations, profiles, and goals; diverse cultural contexts and languages. It is very important, according to Skarlatidou et al. [3] that these volunteers feel confident and satisfied with the technology they use.

It is possible to approach Citizen Science from different disciplines, such as Social Computing, Software Engineering and Human-Computer Interaction [4]. Also, according to Celino et al. [5] it is possible to approach it from different perspectives, such as social, socioeconomic and technological. In particular, when the CC adopts information and communication technology as a pillar, it is called Digital Citizen Science (DCC) [6]. From now on we will use the acronym DCC to denote Digital Citizen Science. Software plays a central role in the DCC.

In a DCC project, volunteers perform certain actions such as collecting, classifying and analyzing samples or solving new challenges. These actions are carried out with the support of information technologies. For example, taking and submitting photos relevant to the domain of interest (as done in the iNaturalist[1] project) is one way of **collecting**. Assigning a sample to predefined classes (as is done in GalaxyZoo[2]) is one way to **classify**. Looking closely at a sample to produce annotations that record characteristics (as is done in Worlds of Wonder[3]) is one way to **analyze**. Using a game to fold proteins (as is done in FoldIt[4]) or to map neurons (as is done in EyeWire[5]) are particular examples for the action of **solving new challenges**.

From the perspective of Software Engineering there are various mechanisms to reduce the effort of creating applications and improve their quality. Additionally, from the perspective of Human-Computer Interaction, various strategies are provided to consider during development, so that the user feels confident with the use of the proposed applications. In general, for Walton & Maiden [7], software reuse is a way of increasing productivity and reducing the time to obtain a technological solution.

In this work, we ask whether in DCC projects, more software artifacts are reused than those specifically developed. This is in accordance with the current trend in the growth of software reuse in general, as described in [8]. To answer this question, a survey and subsequent bibliographic analysis was carried out, which consisted of identifying, for each article, which of the software obtaining strategies was applied.

The present work is organized as follows: in Section 2, software is described as a relevant dimension in Citizen Science. In section 3, the framework for the survey of software artifacts that mediate the participation of volunteers in Citizen Science projects in relation to the strategies used to obtain them and the actions carried out by volunteers is presented. In Section 4, the results of the survey are shown, in Section 5 a Discussion is proposed and finally, in Section 6, Conclusions are presented.

[1] https://www.inaturalist.org/
[2] https://www.zooniverse.org/projects/zookeeper/galaxy-zoo/
[3] https://www.zooniverse.org/projects/lbeiermann/worlds-of-wonder
[4] https://fold.it/
[5] https://eyewire.org/explore

2 Software as a dimension in Digital Citizen Science projects

In Software Engineering there are strategies to reduce the effort of creating applications and improve their quality. The reuse of knowledge about the domain, in the form of design patterns or best practices, reduces design effort and improves the quality of solutions. Code reuse, in the form of libraries, services, and frameworks, reduces development effort. In general, and according to Walton & Maiden [7], software reuse is a way to increase productivity and quality, as well as to reduce the time to obtain a technological solution. Configurable and adaptable software, while more complex to build, multiplies reusability while eliminating the need to involve expert programmers. This saves time and resources, to invest efforts in specific aspects of the Citizen Science project rather than in the technology that supports it [9].

At DCC, software artifacts mediate actions taken by volunteers. In this work three strategies are identified to obtain these artifacts: "Development", "Reuse" and "Repurpose". Next, we describe each one of them.

The "Development" strategy involves the development of a new software artifact with the primary (and possibly only) intention of supporting a specific Citizen Science project. With this strategy, you could create a new software artifact from scratch or include what Taivalsaari et al. [8] mention it as a development from "ad hoc reuse". This refers to development using libraries or non-specific components of Citizen Science. This strategy includes extensive coding.

The "Reuse" strategy, in this work, includes the use (for example, as services) or the deployment (deploy) of artifacts already available that were conceived for another Citizen Science project. Luna et al. [10] mention the reuse of existing applications that require little customization, avoiding creating a new application from scratch. According to Varnell-Sarjeant & Andrews [11], there are several software reuse strategies.

On the other hand, the "Repurpose" strategy refers to the adaptation or application of general purpose software to provide support in some technological aspect of a Citizen Science project. That is, artifacts that were not specifically designed for DCC and that apply to a project. Let's discuss some examples.

Bagnolini et al. [12] present "BiodiveCity", an application that allows volunteers to take and send photographs to investigate biodiversity, in which the position of the sample is additionally captured. The goal is to register animals and plants on campus. BiodiverCity was developed specifically for this project.

Hsu et al. [13], propose a DCC project to address the problem of air pollution in a community. For this, the volunteers propose different hardware and software artifacts. Google Forms is one of the software artifacts adopted in the project for volunteers to upload smell reports. In this project, a more general-purpose application (Google Forms) is repurposed to support a particular DCC project.

On the other hand, Simpson et al. [14] present a web platform called Zooniverse, where volunteers analyze existing audio, image or video samples. With this platform, volunteers can identify, mark and tag or classify submitted samples. Zooniverse acts as a portfolio of DCC projects, which offers authoring tools to create classification and analysis projects, following a common methodology. In addition to the Web platform, the creators of Zooniverse provide the ability to access the project's source code under

an open source license. Zooniverse encourages reuse of both the authoring tool and its source code.

3 Survey on the Development, Reuse and Repurposing of software artifacts that mediate interaction

The objective of the survey in this work is to find computer science articles, written in English (at least their title, abstract and keywords) that have been published until May 2019.

Two main search terms were considered: "Citizen Science" and "Software Engineering". For "Software Engineering" the following first level derived terms were considered: "Software process", "Software design", "Software implementation", these derived terms are some of the activities mentioned for the software development process described in the ISO 12207. Finally, for each of the terms mentioned above, second-level derived terms were established.

The search string was formed using the AND operator between the terms and using the logical OR operator for the first and second level terms (synonyms), as proposed in [15]. The parentheses were also used to separate the logic of each level from the terms used in the search. The data sources considered were Scopus and IEEXplore. The articles that were included in this study come from conferences, journals, workshops and book chapters.

The articles found from the search strategy in the data sources described above were analyzed to determine their inclusion. The inclusion criterion applied refers to articles that use a software production strategy for Citizen Science digital projects and, in particular, those in which the volunteer performs some action using the produced software artifact (the artifact mediates the action of person).

Both the search string and the articles analyzed can be consulted in [16].

4 Results

To obtain the articles to be analyzed in this work, the following steps were carried out: 1) Search in bibliographic sources, 2) Elimination of duplicates, 3) Reading of the title, abstract and keywords, 4) Complete reading of the articles. The number of articles obtained in each of the steps mentioned above can be visualized in Fig. 1.

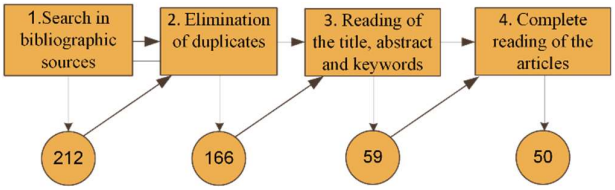

Fig. 1. Result set

This work considers articles from journals and conferences published until May 30, 2019. The number of publications in the area of Computer Science, related to the development of software to be used by a volunteer has varied throughout of the years. The first publication produced by the search appears in 2010. Fig. 2 shows the variation in the number of articles from 2010 to 2019. The quantities are discriminated between conference and journal articles for each year of publication.

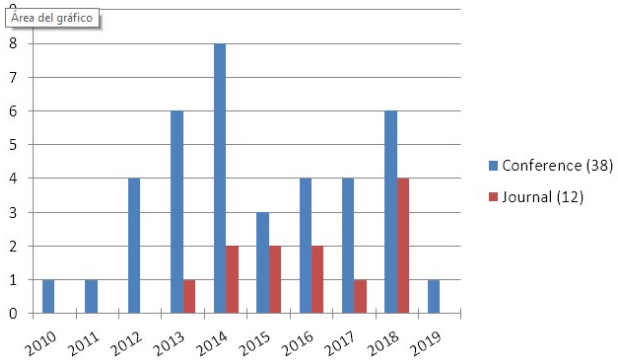

Fig. 2: Distribución de los artículos en los años relevados

Once the articles were quantified by class and by their year of publication, the following question was answered:

- *In what quantity is each identified strategy presented?*

To answer the question previously introduced, for each of the three strategies previously described (Development, Repurpose and Reuse), the corresponding number of articles was determined. Fig. 3 shows the numbers obtained for each of them, considering that, in some cases, the software artifacts combine strategies.

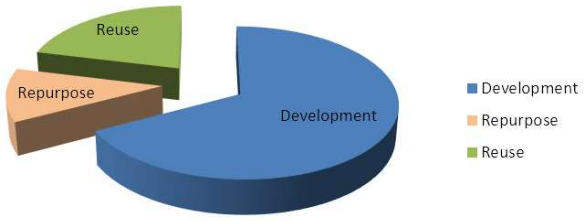

Fig. 3: Distribution of strategies for obtaining software artifacts

When analyzing the graph presented in Fig. 3, it is visualized that the most used strategy to obtain software artifacts is development. As a result of the visualization of the graph (Fig. 3), it was decided to analyze the behavior, over time, of the use of each of the strategies, since an emerging *hypothesis* is that *the greatest amount of*

development of these artifacts occurred in the first years and that, as the years passed, a greater maturity was achieved, increasing the strategies of repurpose and reuse.

In Fig. 4, the graph with the distribution of the separate articles in the years covered by the study is shown.

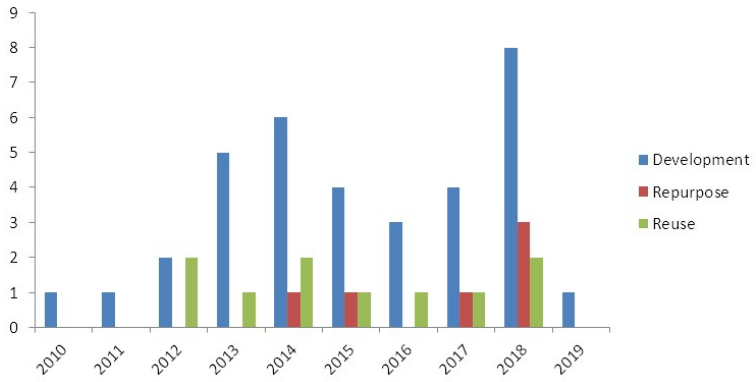

Fig. 4: Evolution of strategies over time

Fig. 4 shows that the development strategy currently prevails over the other two. The observed phenomenon contradicts the original hypothesis in which a greater amount of development of these artifacts was expected during the first years. When analyzing the reuse strategy, it can be seen that it did not increase over the years, but was maintained. These values are surprising because they do not follow the trend of reuse of software artifacts as indicated by Taivalsaari et al. [8]. In particular, reuse is manifested in works [10], [17], [18] and [19].

Finally, regarding the repurpose, a slight growth is detected in the last year. The artifacts are SMS [20], [21], Twitter [22], [23], Facebook [24] and Google documents [13].

Lastly, but not least, it was decided to *analyze the artifacts according to the actions carried out by the volunteers*. Fig. 5 shows the results.

Fig. 5: Actions carried out by volunteers and mediated by software artifacts

Fig. 5 shows the great prevalence of artifacts where the volunteer collects data as an action.

5 Discussion

Reusing software artifacts allows you to take advantage of the knowledge that these software artifacts encapsulate.

Additionally, when developing artifacts for later reuse, on the one hand a high level of knowledge of the domain for which they are conceived is necessary [25] and, on the other, the knowledge of technology experts (for example, software engineers) to propose a coherent software solution is required [26]. When developing a new software artifact, it is also necessary to consider aspects of the action that the artifacts will mediate with people (interaction of people with the artifact) and its usability [27].

The development of software to support a DCC project can be viewed from two dimensions. On the one hand, support for the *methodological* aspects and, on the other, support for the *community* of people in the project. From a methodological perspective, a software artifact incorporates knowledge, for example, about how to collect samplers for a specific domain and how to guide the user to make a valuable contribution to the project. Regarding the community perspective, the software artifact incorporates the knowledge of how to carry out recruitment, training and retention [22]. In this way, technological support crosses both dimensions.

Under the premise that software components incorporate knowledge, the implications (advantages and disadvantages) of building them from scratch must be analyzed. For example, it is possible to wonder what it would take to develop a communication tool (for example, a social network) from scratch to support communication between members of a community. It is known that there are many tools for this purpose, which are widely used by the general public, and which are not only highly proven, but also conceived by multidisciplinary teams of experts who contributed their knowledge. In this way, by adopting one of these general-purpose tools, you take advantage of all that built-in knowledge and it is also very likely that the people who join the project will know how to use them. On the contrary, by not adopting a pre-existing one, as an advantage it can be thought that flexibility and customization are added through ad-hoc development for the domain of the project, but as a disadvantage the loss of knowledge already incorporated in this type of tools.

When carrying out specific software developments for each project, the loss of interoperability must be analyzed, and the weakness when integrating with other projects to generate a possible collaboration network in DCC. Additionally, another problem that can emerge with a new software development is the technological and economic solvency to store the large volumes of data for each project and the recovery techniques for these data, which end up being limited to the particular design and making it impossible it reuses in other projects.

On the other hand, the analysis shows that the type of action mediated by the software artifacts carried out by the volunteers is data collection. As a result, it is valid to ask yourself some questions, such as, for example, what happens when the same volunteer or community of volunteers participate in more than one project at the same time? In this context, will the volunteer have to learn to use a different interface for each project in which he participates? How does the use of multiple applications and multiple styles of carrying out the same task affect your personal confidence? Another aspect of this massive action by people in DCC projects is what is the real level of

volunteer participation? Why is it limited to collecting samples, when it could participate more actively in the project by performing more actions?

Another question that emerges, as a result of the little reuse detected, is whether Software Engineering and HCI should join forces to propose processes that allow the construction of reusable software artifacts with a focus on "reusers" (volunteers and scientists who wish to propose their own Projects, people who do not necessarily have knowledge of software development). Considering also that, on the scale of the volunteer participation, there is multiculturalism, different legislation on the data that is generated and analyzed, as well as technological limitations and various individual and group motivations. How can this collective knowledge multiply and serve other emerging communities? Could it be that with each new software development, is the wheel being reinvented?

It is considered very relevant to open the debate on the usefulness of proposing a reuse process focused on reusers, which is simple, and usable for people who are not experts in software development. Furthermore, taking into account the multidisciplinary nature of Digital Citizen Science projects.

6 Conclusions

In this work, a survey and analysis of articles that refer to software artifacts to support Digital Citizen Science projects was presented. Those artifacts were classified according to their acquisition strategy: Development, Reuse, and Repurpose. Additionally, the distribution of the articles was analyzed according to the actions carried out by the volunteers (actions mediated by the artifacts).

Currently most of the software artifacts are obtained through the "Development" strategy and are for actions of "Collect" data.

In order to promote the reuse strategy, it is planned to work on proposing a reuse process focused on "reusers" (people who do not necessarily have knowledge of software development).

References

1. Cohn, J. P. (2008). Citizen science: Can volunteers do real research?. BioScience, 58(3), 192-197.
2. Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., ... & Murray, P. (2008). Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. Monthly Notices of the Royal Astronomical Society, 389(3), 1179-1189.
3. Skarlatidou, A., Hamilton, A., Vitos, M., & Haklay, M. (2019). What do volunteers want from citizen science technologies? A systematic literature review and best practice guidelines. JCOM: Journal of Science Communication, 18(1).
4. Preece, J. (2016). Citizen science: New research challenges for human–computer interaction. International Journal of Human-Computer Interaction, 32(8), 585-612.

5. Celino, I., Corcho, O., Hölker, F., & Simperl, E. (2018). Citizen science: design and engagement (dagstuhl seminar 17272). In Dagstuhl Reports (Vol. 7, No. 7). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
6. Nov, O., Arazy, O., & Anderson, D. (2011, February). Dusting for science: motivation and participation of digital citizen science volunteers. In Proceedings of the 2011 iConference (pp. 68-74). ACM.
7. Walton, P., & Maiden, N. (Eds.). (2019). Integrated software reuse: management and techniques. Routledge.
8. Taivalsaari, A., Mikkonen, T., & Mäkitalo, N. (2019, August). Programming the Tip of the Iceberg: Software Reuse in the 21st Century. In 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 108-112). IEEE.
9. Tangmunarunkit, H., Hsieh, C. K., Longstaff, B., Nolen, S., Jenkins, J., Ketcham, C., ... & Khalapyan, Z. (2015). Ohmage: A general and extensible end-to-end participatory sensing platform. ACM Transactions on Intelligent Systems and Technology (TIST), 6(3), 38.
10. Luna, S., Gold, M., Albert, A., Ceccaroni, L., Claramunt, B., Danylo, O., ... & Radicchi, A. (2018). Developing mobile applications for environmental and biodiversity citizen science: considerations and recommendations. In Multimedia Tools and Applications for Environmental & Biodiversity Informatics (pp. 9-30). Springer, Cham.
11. Varnell-Sarjeant, J., & Andrews, A. A. (2015). Comparing reuse strategies in different development environments. In Advances in Computers (Vol. 97, pp. 1-47). Elsevier. https://doi.org/10.1016/bs.adcom.2014.10.002
12. Bagnolini, Guillaume and Da Costa, Georges and Gerino, Magalie and Roth, Mathias and Trân, Cécile Multidisciplinarity for biodiversity management on campus through citizen sciences. In: 2nd Workshop on Smart and Sustainable City (WSSC 2017) in conjunction with 2017 IEEE Smart World Conference, 4 August 2017 (San Francisco, United States) (2017).
13. Hsu, Y. C., Dille, P., Cross, J., Dias, B., Sargent, R., & Nourbakhsh, I. (2017, May). Community-empowered air quality monitoring system. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (pp. 1607-1619).
14. Simpson, R., Page, K. R., & De Roure, D. (2014, April). Zooniverse: observing the world's largest citizen science platform. In Proceedings of the 23rd international conference on world wide web (pp. 1049-1054). ACM.
15. Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. Journal of systems and software, 80(4), 571-583.
16. Lliteras, A.B., Fernandez A., Torres D. (2020) Result Set_Desarrollo, reuso, y resignificación de artefactos de software en Ciencia Ciudadana. ¿Reinventando la rueda?. https://doi.org/10.5281/zenodo.3968740
17. Sheppard, S. A., Wiggins, A., & Terveen, L. (2014, February). Capturing quality: retaining provenance for curated volunteer monitoring data. In Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing (pp. 1234-1245). ACM.
18. Brovelli, M. A., Minghini, M., & Zamboni, G. (2016). Public participation in GIS via mobile applications. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 306-315.
19. Chen, L. J., Ho, Y. H., Lee, H. C., Wu, H. C., Liu, H. M., Hsieh, H. H., ... & Lung, S. C. C. (2017). An open framework for participatory PM2. 5 monitoring in smart cities. IEEE Access, 5, 14441-14454.
20. Martinelli, M., & Moroni, D. (2018). Volunteered geographic information for enhanced marine environment monitoring. Applied Sciences, 8(10), 1743.

21. Beza, E., Reidsma, P., Poortvliet, P. M., Belay, M. M., Bijen, B. S., & Kooistra, L. (2018). Exploring farmers' intentions to adopt mobile Short Message Service (SMS) for citizen science in agriculture. Computers and Electronics in Agriculture, 151, 295-310.
22. Tapia, A. H., LaLone, N. J., MacDonald, E., Priedhorsky, R., & Hall, M. (2014, January). Crowdsourcing rare events: Using curiosity to draw participants into science and early warning systems. In ISCRAM.
23. II, R. T. B., Lundgren, L., Crippen, K. J., & MacFadden, B. J. (2018, June). Designing for Public Participation in Paleontology Through the Development of an App. In ECSM 2018 5th European Conference on Social Media (p. 462). Academic Conferences and publishing limited.
24. Jambeck, J. R., & Johnsen, K. (2015). Citizen-based litter and marine debris data collection and mapping. Computing in Science & Engineering, 17(4), 20-26.
25. Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. Research policy, 32(7), 1217-1241.
26. Annaiahshetty, K., & Prasad, N. (2013, April). Expert System for Multiple Domain Experts Knowledge Acquisition in Software Design and Development. In 2013 UKSim 15th International Conference on Computer Modelling and Simulation (pp. 196-201). IEEE.
27. Calp, M. H., & Akcayol, M. A. (2019). The importance of human computer interaction in the development process of software projects. arXiv preprint arXiv:1902.02757.