Reproducing BugsInPy

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—We present our experience on replicating a bug dataset for the Python programming language. The bug dataset provides some information about the software environment, but this environment decays quickly into something uninstallable. We assess the reproducibility over time and improve the reproducibility of the dataset.**

***Index Terms*—reproducibility, bug database, python**

I. Introduction

What is BugsInPy [1].

Why does it exist.

What information does the dataset contain.

II. Methodology

Explain the bugsinpy-testall script.

Explain how Conda packaging works.

Explain how Pip and PyPI works.

III. Results

Table containing results with Debian 11 and results with Conda

IV. Discussion

What makes our reproduction easy.

What makes our reproduction hard.

A. Recommendations to BugsInPy users

B. Recommendations to artifact authors

C. Threats to validity

Is it possible that our reproduction is not working when it should be?

Is it possible that our reproduction won't be reproducible by others?

V. Conclusion

Does our study support the conclusion?

Future directions for research?

- Improve reproducibility of BugsInPy
- Improve description of software environments.
- Make reproducibility tools easier to use.

References

References

[1] R. Widyasari, S. Q. Sim, C. Lok, *et al.*, "BugsInPy: A database of existing bugs in python programs to enable controlled testing and debugging studies," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2020, New York, NY, USA: Association for Computing Machinery, Nov. 8, 2020, pp. 1556–1560, ISBN: 978-1-4503-7043-1. DOI: 10.1145/3368089.3417943. [Online]. Available: https://doi.org/10.1145/3368089.3417943 (visited on 07/08/2023).

Appendix
Code, Data, and Reproducing

A snapshot of the latest state of this code can be found at: ...(ZenodoDOI).

A rolling release of the code can be found at: ...(GitHub).

In the rolling release or snapshot:

- `data` holds a machine-readable view of the data, split across several files.
- `spack/spack.lock` contains the Spack environment in which this experiment was run.

To reproduce this paper, run:

```
#␣command
output
```

After which, the results will be here:

- `reports/main.pdf` This is the actual paper.
- `raw_data` This is the raw data.