

# Context-aware In-process Crowdworke Recommendation

## 上下文感知的过程中的众测人员推荐

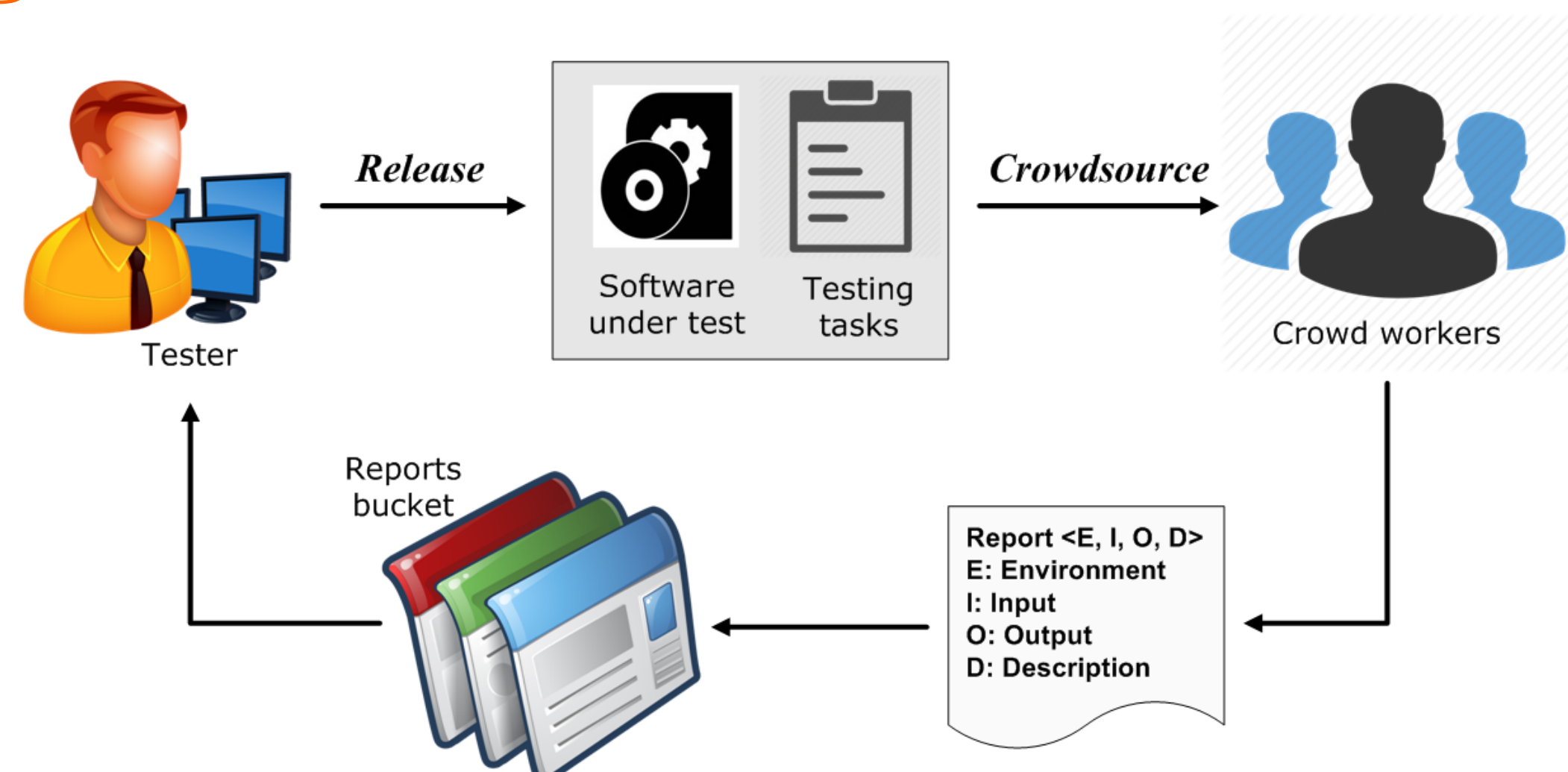
Junjie Wang, Ye Yang, Song Wang, Yuanzhe Hu, Dandan Wang, Qing Wang

In Proceedings of the 42st ACM/IEEE International Conference on Software Engineering (ICSE 2020)

联系人: 王俊杰 王青

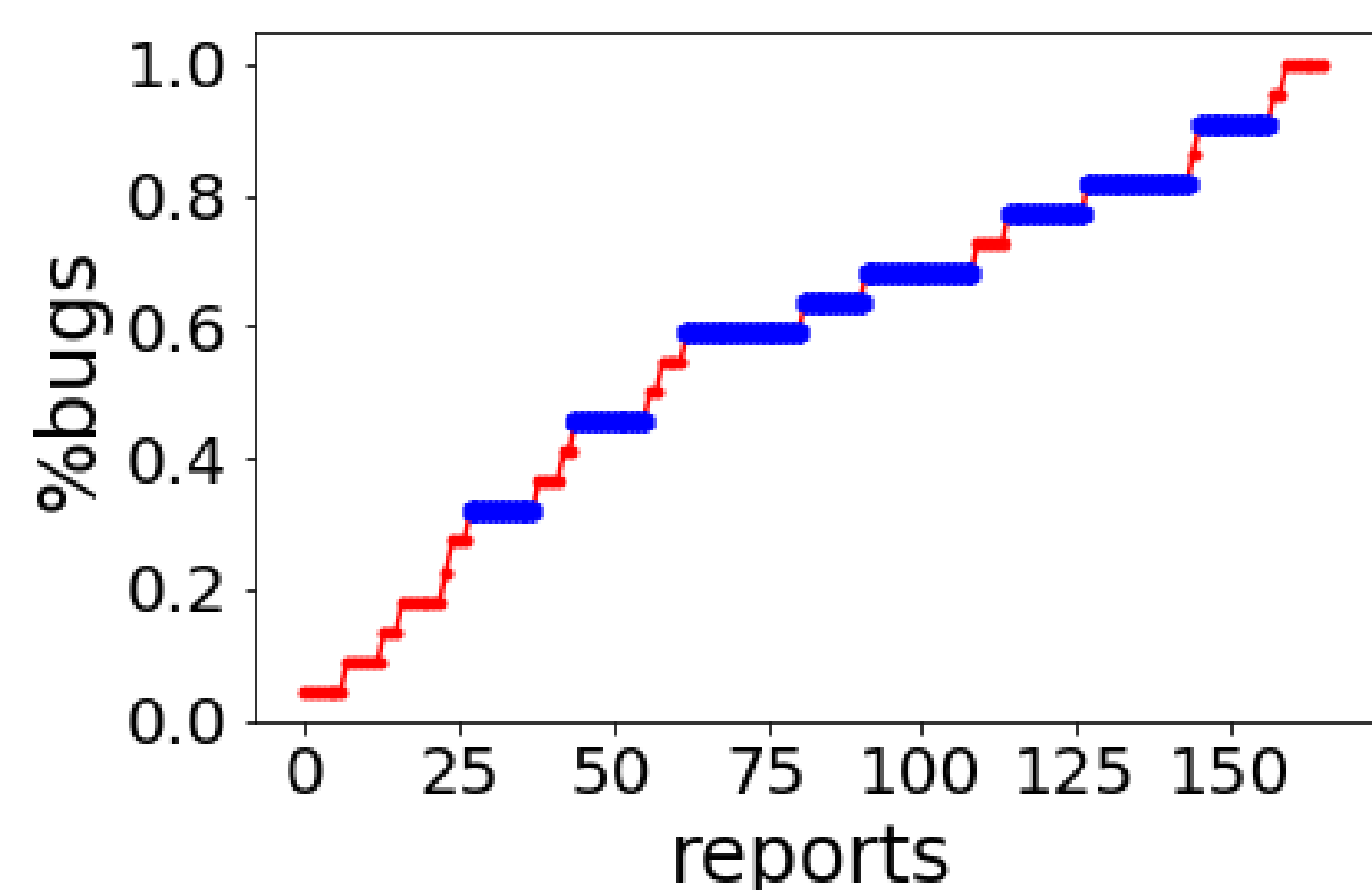
联系方式: {junjie, wq}@iscas.ac.cn

### Background

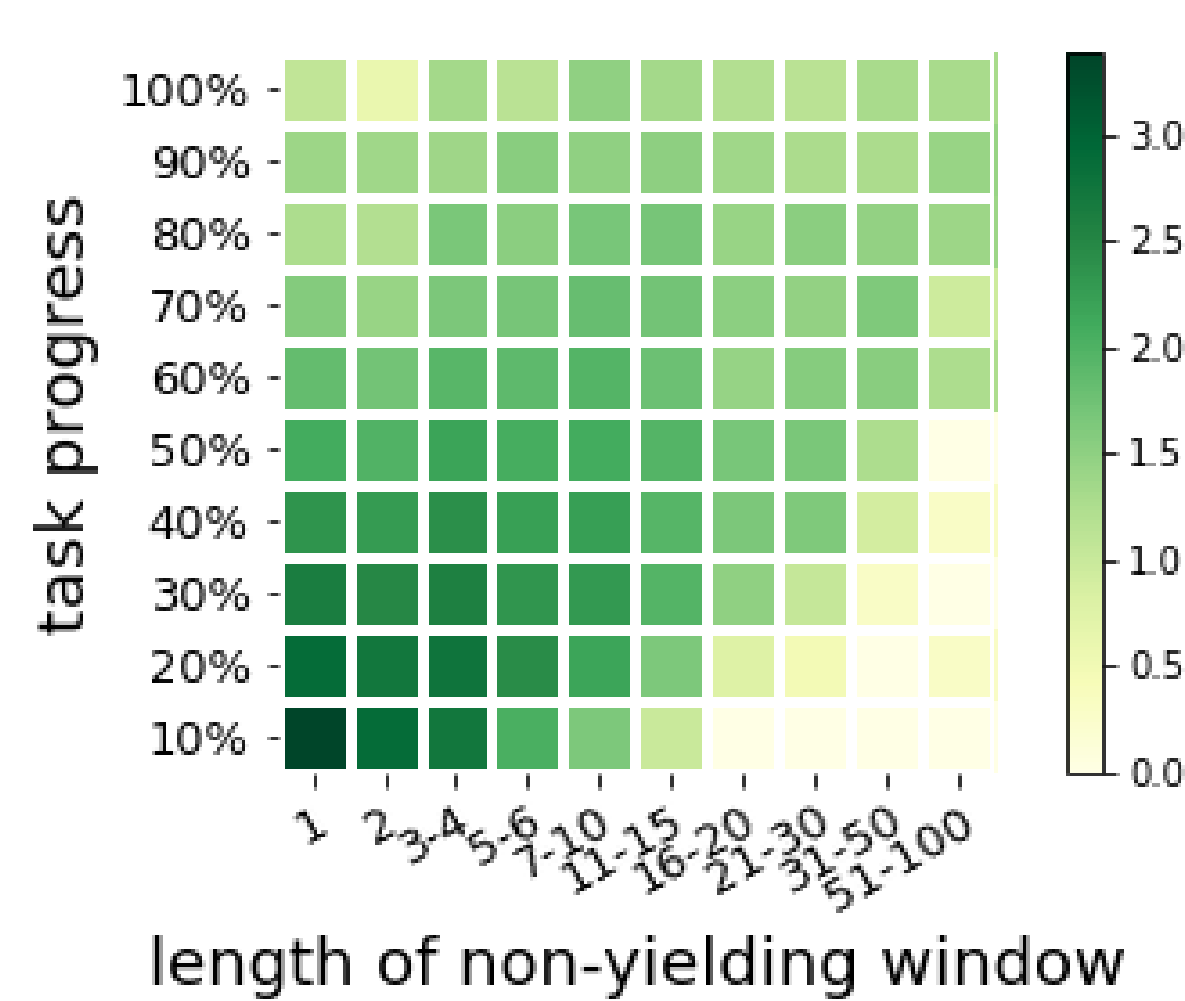


- Crowdtesting entrusts tasks to online crowdworkers whose diverse testing environments, background, and skill sets could significantly contribute to more reliable, cost-effective, and efficient testing results.
- Identifying and optimizing open participation is essential.

### Motivation



- **Long-sized non-yielding windows** (the flat segments), i.e., no new bugs are revealed in consecutive test reports during the process of a crowdtesting task.

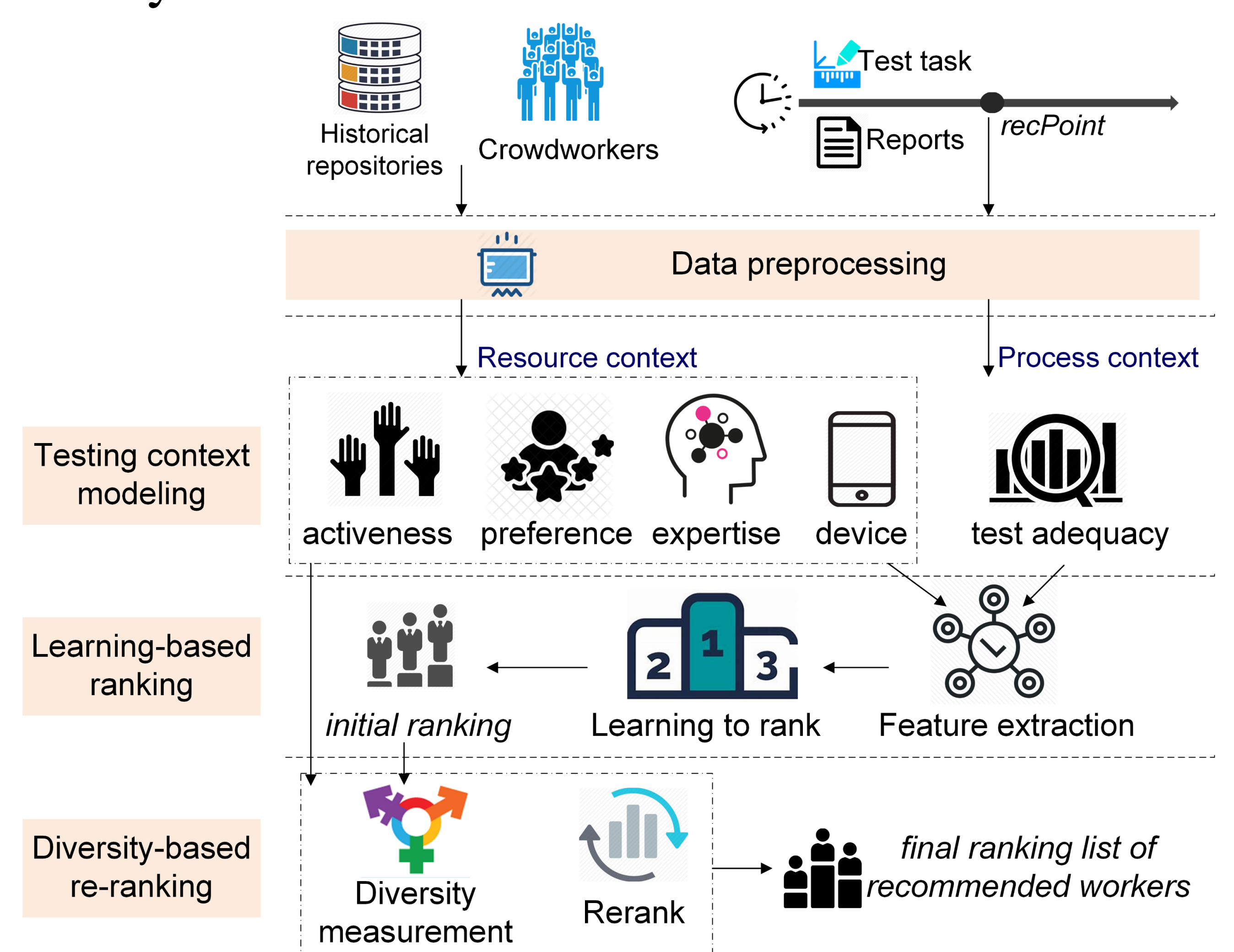


- 85% tasks have 10- or longer-sized non-yielding window.
- 39% cost is wasted.

- Current workers possibly have similar bug detection capability with previous workers on the same task.
- Unsuitability of existing one-time worker recommendation.
- The need for in-process crowdworker recommendation by learning from the dynamic underlying contextual information to mitigate the non-yielding window.

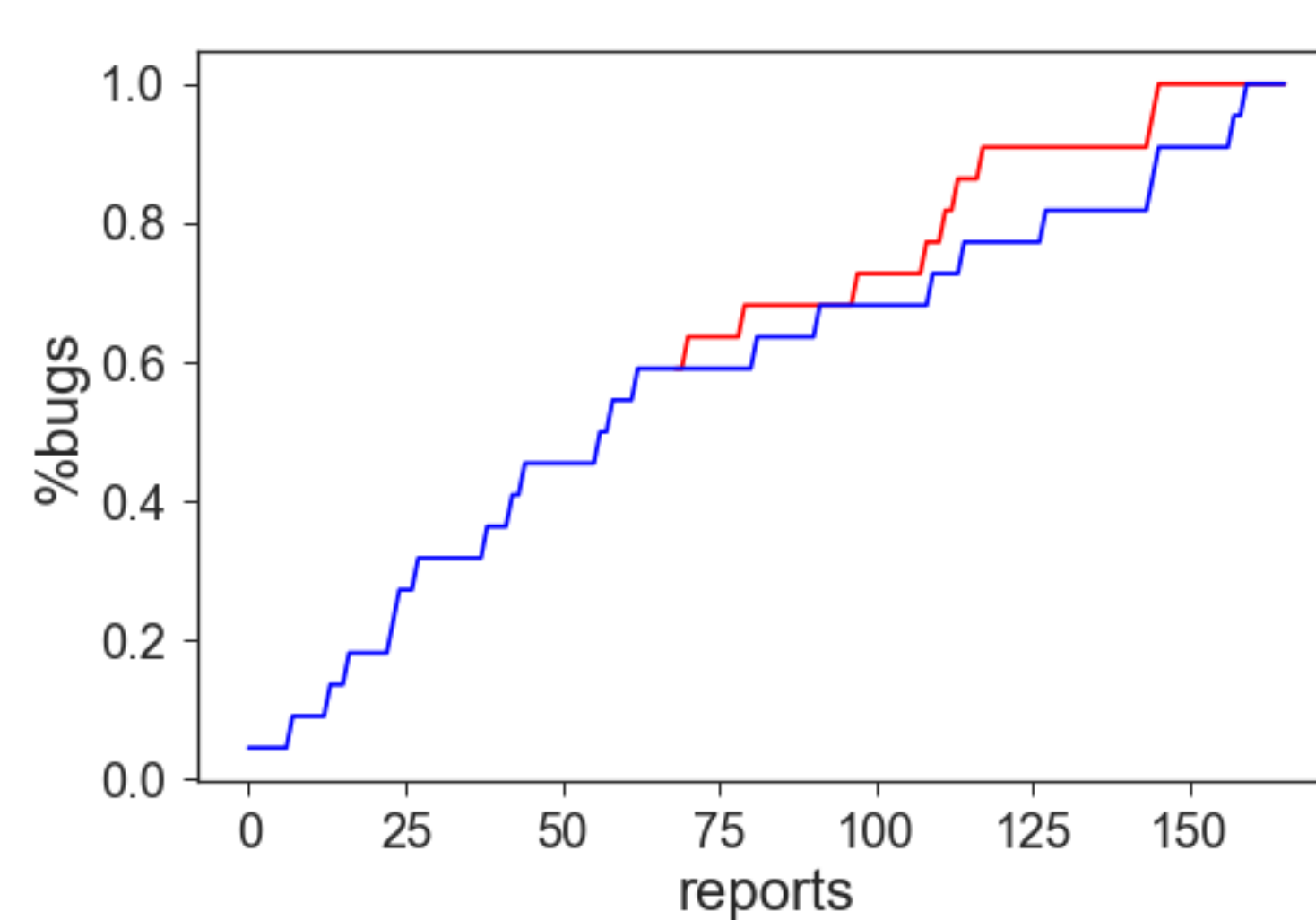
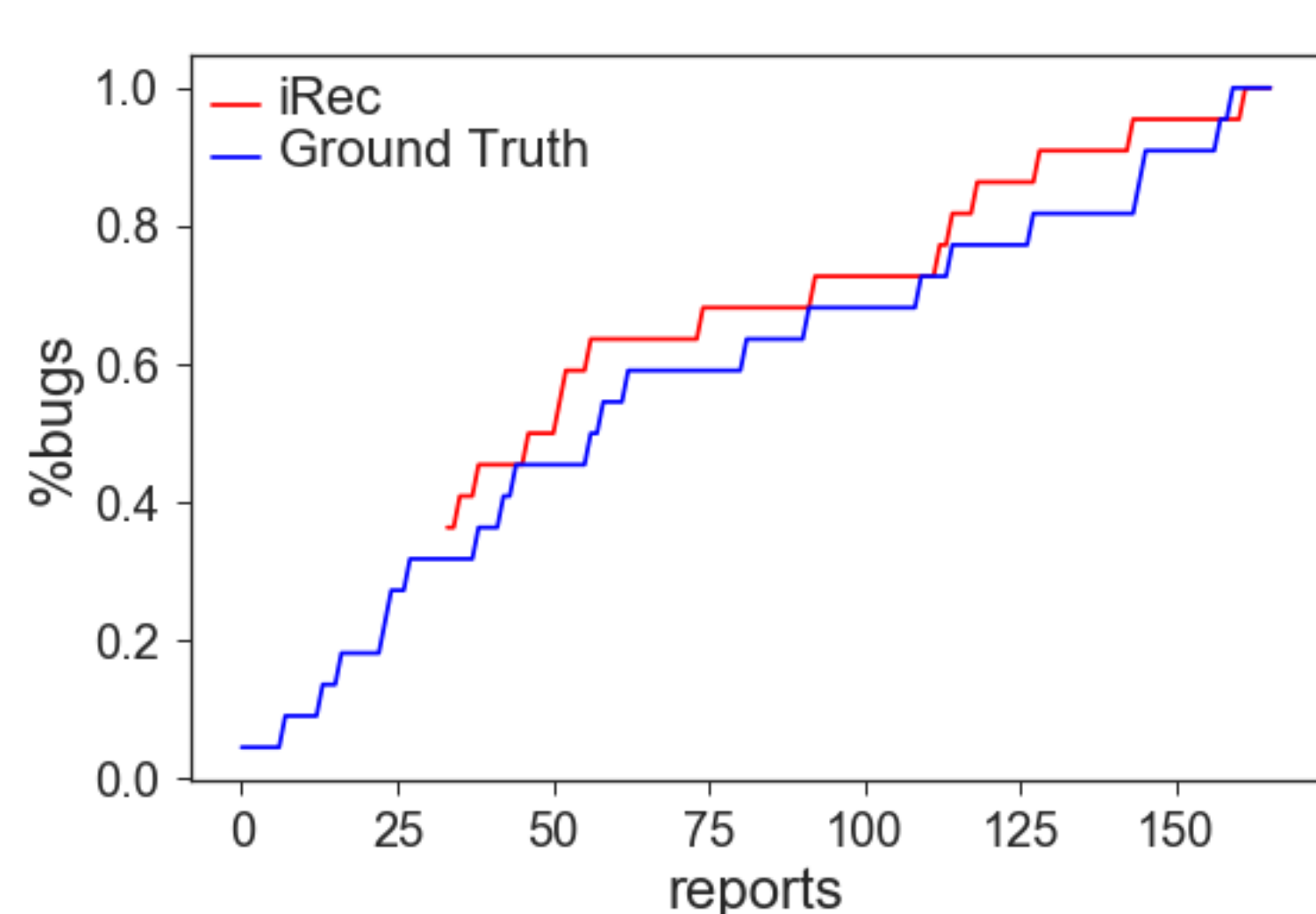
### Approach

Context-aware in-process crowdworker recommendation approach (iRec) to dynamically recommend a diverse set of capable crowdworkers based on various contextual information at a specific point of crowdtesting process, aiming at shortening the non-yielding window and improving bug detection efficiency.



- 1) **Test context modeling:**
  - *Process context:* process-oriented information related to the crowdtesting progress of current task.
  - *Resource context:* availability and capability factors concerning the competing crowdworker resources.
- 2) **Learning-based ranking:** extract 26 features from both process context and resource context, and learn the workers with the greatest potential to detect bugs abstracted from historical tasks.
- 3) **Diversity-based re-ranking:** adjust the ranked list of recommended workers by optimizing the worker diversity to reduce duplicate bugs.

### Experiment



- A median of 50% remaining bugs can be detected with first 10 recommended workers by iRec, 400% improvement compared with current practice
- With recommended workers, bug arrival curve can rise quickly
- Reduce an average of 8%-12% cost