Challenges and Future Research Direction for Microtask Programming in Industry

Masanari Kondo
kondo@ait.kyushu-u.ac.jp
Kyushu University, Japan

Shinobu Saito, Yukako Iimura
{shinobu.saitou.cm,yukako.iimura.vr}@hco.ntt.co.jp
NTT Computer and Data Science Laboratories, Japan

Eunjong Choi, Osamu Mizuno
{echoi,o-mizuno}@kit.ac.jp
Kyoto Institute of Technology, Japan

Yasutaka Kamei, Naoyasu Ubayashi
{kamei,ubayashi}@ait.kyushu-u.ac.jp
Kyushu University, Japan

ACM Reference Format:
Masanari Kondo, Shinobu Saito, Yukako Iimura, Eunjong Choi, Osamu Mizuno, and Yasutaka Kamei, Naoyasu Ubayashi. 2022. Challenges and Future Research Direction for Microtask Programming in Industry. In *19th International Conference on Mining Software Repositories (MSR '22), May 23–24, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3524842.3528511

Microtask programming [4] is a solution to promote distributed development in industry. The key idea of microtask programming is to reduce face-to-face communication across developers by splitting the development task of software into independent *microtasks*. Such microtasks can be completed by *crowd workers* who work remotely and at their preferable time such as early morning. *Dedicated developers* who have the responsibility for the progress of development split the task into microtasks, and distribute them to crowd workers. Hence, microtask programming has these two actors. Our research team reported that microtask programming has potential benefits such as the fluidity of project assignments in industrial companies [4]. However, we suppose it still has challenges. In addition, it is still unclear what are future research direction to support both actors in microtask programming, though our research team has conducted three studies for microtask programming so far [2–4].

We found three key challenges that lie ahead to employ microtask programming in industry by our interview: *well-being*, *encouragement*, and *responsibility*. We interviewed two developers (i. e., crowd worker and dedicated developer), who had participated in the microtask programming project in [4], for 1.5 hours to clarify these challenges and highlight future research direction. Our presentation will describe our interview, these challenges, and future research direction. We outline the challenges and the direction as follows.

Challenge 1: Well-being. Microtask programming currently affects actors' well-being. For example, dedicated developers are under pressures. Crowd workers ask them all questions and concerns in microtask programming. An interviewee, who was a dedicated developer, said if they receive a question from crowd workers, they are under a pressure to respond to the question quickly. This is because 1) they believed crowd workers wait for quick responses and 2) if they delayed responding to questions, the progress of development would be delayed. This pressure is not a serious problem in face-to-face communication because dedicated developers can observe the status of crowd workers and easily realize they wait for responses or leave the desk and go to lunch. Hence, telling the status of crowd workers to dedicated developers is important to mitigate the pressure and achieve dedicated developers' well-being. We believe that the status should include not only active/inactive but what time they will start the remaining microtasks again.

Challenge 2: Encouragement. It is also an important challenge to encourage crowd workers. For example, in remote work, developers in industry feel that communication takes a lot of time because of asynchronous communication [1]. Indeed, the interviewees said that 1) crowd workers want dedicated developers to respond quickly, 2) even the response times in popular remote work situations (i. e., open source software communities) are insufficient for crowd workers in industry. Such communication would discourage crowd workers and have them give up finishing the assigned task. This tendency would lose the time that crowd workers spared for conducting microtasks and delay the progress of the development.

Challenge 3: Responsibility. The responsibility in microtask programming is currently unequally distributed. The important step in microtask programming is to split the task into microtasks. Each microtask is independent and can be completed individually. In other words, crowd workers do not know the overview of the development and have no responsibility for the progress of development. On the other hand, dedicated developers have all responsibility for the development. Indeed, the interviewee (i. e., dedicated developer) said that they were nervous because of the over-concentration of responsibility during all times of the project period. Hence, it is important to distribute not only microtasks but the responsibility.

Future Research Direction. These challenges are important in industrial companies for developers' mental. We believe that researchers in the MSR community can address them. For example, we envision researchers can address the first challenge by implementing a bot that is an intermediate between two actors. Specifically, the bot delivers the appropriate number of questions at the appropriate time to dedicated developers depending on when crowd workers need the response. Also, the bot can be used to show the status of the crowd workers. For example, crowd workers go to lunch and come back in 1 hour or come back soon.

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9303-4/22/05…$15.00
https://doi.org/10.1145/3524842.3528511

REFERENCES

[1] Denae Ford, Margaret-Anne Storey, Thomas Zimmermann, Christian Bird, Sonia Jaffe, Chandra Maddila, Jenna L. Butler, Brian Houck, and Nachiappan Nagappan. [n.d.]. A Tale of Two Cities: Software Developers Working from Home during the COVID-19 Pandemic. *ACM Trans. Softw. Eng. Methodol.*, Article 27 ([n. d.]), 37 pages. accepted, to appear.

[2] Shinobu Saito and Yukako Iimura. 2020. Hybrid Sourcing: Novel Combination of Crowdsourcing and Inner-Sourcing for Software Developments. In *Proceedings of the 15th International Conference on Global Software Engineering (ICGSE)*. ACM, 81–85.

[3] Shinobu Saito and Yukako Iimura. 2021. Toward Understanding of Employee Motivation for Software InnerSourcing : Industrial Experience Report. In *Proceedings of the 2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*. IEEE/ACM, 33–38.

[4] Shinobu Saito, Yukako Iimura, Emad Aghayi, and Thomas D. LaToza. 2020. Can Microtask Programming Work in Industry?. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM, 1263–1273.