QUALITATIVE STUDY

To study the quality of the refactored microservices and their benefits to the developers, we conducted a survey through two software engineers and one researcher with program analysis background. On an average, the participants had an industrial experience of 13 years in different software engineering roles. All three participants had prior working experience with Java programming language and microservices. It took participants around four weeks to understand Daytrader along with their daily work. The participants were provided with a high-level summary of our work including the original application and our refactoring; no further instructions were given to avoid bias. The effort needed to understand an application along with necessary skill sets to assess output limited our study to few participants and datasets.

Study Instructions

With each participant we explained our code refactoring approach using examples from DayTrader and Plantsbywebsphere and all the figures in this paper. They examined the codes, cross questioned us, manually refactored the code on their own, and then answered the following questions :

- How much development effort does this refactored output reduce? a) Less(<20%) b) Moderate(20–40%) c) 40-60%(Significant) d) >60%(Completely)
- How much would you modify before deploying the refactored work? a) No change (Use the code as such) b) Low (change <20%) c) Moderate (20-40%) d) Significantly (>40%)
- Please list the type of changes you would do.
- Optionally, please explain your inputs.

Results

All three participants said that compared to manual refactoring, automation reduces their effort in generating microservice projects and APIs by 40-60%. A participant mentioned that he avoids making manual changes and relies on automatic refactoring to prevent the introduction of new errors. The other two participants said they would make moderate changes as they felt the number of APIs are too many; they may group the exposed APIs. All three participants mentioned that they would introduce the following changes 1) Modify API and variable names, 2) improve candidate clusters by introducing new endpoints more aligned with their business domain model, 3) remediate static fields, etc. Note that (2) is out of our scope as we do not perform clustering but expose functions of the given clusters as REST APIs. We believe this study shows that our output accelerates their development task, but more business-centric changes are still required.