Explaining Pair Programming Session Dynamics from Knowledge Gaps

Franz Zieris¹ Lutz Prechelt²

This is an extended abstract of the paper with the same title [ZP20a] which was presented at the 42nd International Conference on Software Engineering (2020).

Keywords: pair programming; qualitative analysis; grounded theory methodology

1 Background, Data, and Research Method

Pair programming (PP) has many purported benefits, including higher code quality, faster progress, and knowledge transfer between developers. Despite a lot of research on the effectiveness of PP, the question when it is useful or less useful remains unsettled: A meta-analysis found mere tendencies and a lot of between-study variance [Ha09]; a large controlled experiment could not determine consistent moderating effects of task complexity and developer expertise [Ar07]. Even though the feasible experimental setups tend to be highly unrealistic, there have been only few qualitative studies which looked at the actual PP process in industrial contexts (e.g., [Pl15]).

We follow Straussian Grounded Theory Methodology [SC90] to understand how pair programmers actually transfer knowledge. We analyze 26 recordings of industrial PP sessions from 9 companies which we selected in the manner of theoretical sampling from the PP-ind session repository [ZP20b]. For open coding, we build on our own prior work [ZP14; ZP16] that identified various phenomena related to within-session knowledge build-up and transfer. We validate our findings with practitioners from four companies.

2 Results

We identify two different types of required knowledge and explain how different constellations of knowledge gaps in these two respects lead to different session dynamics:

- Industrial pairs mostly deal with gaps in project-specific system understanding, or S knowledge. They address any differences in their respective system understanding first before building up new system understanding together.

¹ Freie Universität Berlin, Institut für Informatik, Takustr. 9, 14195 Berlin, Deutschland zieris@inf.fu-berlin.de
² Freie Universität Berlin, Institut für Informatik, Takustr. 9, 14195 Berlin, Deutschland prechelt@inf.fu-berlin.de

doi:10.18420/SE2021_48
• Differences in *general software development knowledge*, or G knowledge, hardly hamper the PP process. Rather, such a difference is an opportunity to transfer knowledge—which pairs only do after they dealt with their S knowledge gaps.

• Building up lacking G knowledge together in a PP session appears to be difficult.

• Pair constellations with *complementary knowledge* allow both partners to contribute S knowledge and G knowledge, respectively, which makes pair programming a particularly effective practice.

• Software developers may use our findings when forming pairs (e.g., by choosing a partner and/or amending the goal of the session such that differences in their knowledge levels play out favorably) or as a means of reflecting after a session (e.g., whether the *right* knowledge gaps were addressed or which were newly identified).

References


