

When Grounded Theory is Not Enough: Additions for Video-Based Analyses of Software Engineering Process Phenomena

A Research Methods Discussion

Franz Zieris¹

Abstract: I discuss how the practices of the Grounded Theory Methodology are not enough to conduct qualitative research on process phenomena of software development based on video recordings. I present five extensions and auxiliary practices to fill in the gaps.

Keywords: Grounded Theory Methodology; Qualitative Research

1 Introduction and Background

My research group has been investigating the development practice of pair programming (PP) since 2004. To understand how PP actually works, we record sessions (consisting of screencast, webcam, and audio) of industrial developers working in pairs on their everyday tasks. Our ultimate goal is to provide practitioners with actionable advice.

We tried to apply Grounded Theory Methodology (GTM) in its Straussian form [SC90], in particular open, axial, and selective coding, and made observations of two kinds: (1) the GTM coding practices are not readily applicable to our kind of data; (2) the Straussian GTM alone (theoretical sampling and applying the coding practices until theoretical saturation) does not lead to a qualitative study with the desirable properties of naturalistic and holistic inquiry, rich and dense data, and active consideration of the researcher role. Here, I present five additions of the GTM from our group (some already reported in [SPP08]), which I believe to be helpful tools for other researchers considering the GTM to study SE process phenomena.

2 Additions: Extensions and Auxiliary Practices

Data Collection (1): Observations & Interviews. Both Straussian advice on open coding and qualitative SE research in general are often focused on (non-naturalistic) *interviews*.

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

Observations, in contrast, are not limited to what subjects are aware of and able to express. Our research combines both: We record PP sessions as our main data source (observation) and afterwards reflect with the developers on their recent session (interview) which combines capturing their personal accounts and checking findings for resonance. Others may have reversed roles: Rely on interviews supported by observation.

Data Collection (2): Stay in the Field & Blend. The researcher role in the field is not discussed in Straussian GTM. Instead of collecting data on single events, we stay at industrial partners multiple days or weeks on end. In assuming a hybrid role of researcher and software engineer, we also talk to developers on a technical level about their everyday issues, thereby establishing trust and collecting *context* information valuable for interpretation later on.

Sampling: Repository. Establishing trust with a company and collecting data costs time and effort. Theoretical sampling in the sense of *collecting* new data when needed is often impractical. A *repository* built over years organizing raw data (here: session recordings) and metadata is a source for ready-to-analyze data from multiple companies for different studies.

Analysis (1): Filter for Data. Open coding and theoretical sensitivity alone lead to too many concepts when the phenomenon is non-trivial and data is rich. Our case is both: Video recordings of two developers working on a task from their domain at their own speed without the need to explain it to an interviewer are *dense*. Defining a *filter* for what phenomena to look out for and which epistemological stance to take (e.g., to only analyze knowledge transfer based on observable explicit behavior) is necessary to not drown in concepts.

Analysis (2): Reusable Concepts. The GTM requires all concepts to be grounded in data, but studying complex phenomena in one take leads to shallow concepts. To resolve this contradiction, we propose to (1) develop grounded concepts in a reusable way including naming schemes and operationalization rules, and (2) reuse such concepts where fit to avoid reinventing the wheel. For our research, we developed a set of base activities of a PP process to capture developers' primary intentions [SP13], which proved valuable for jump-starting focused analyses (e.g., on knowledge transfer) multiple times.

References

- [SC90] Strauss, A. L.; Corbin, J. M.: Basics of Qualitative Research: Grounded Theory Procedures and Techniques. SAGE Publications, Inc, 1990, ISBN: 0-8039-3251-0.
- [SP13] Salinger, S.; Prechelt, L.: Understanding Pair Programming: The Base Layer. BoD, Norderstedt, Germany, 2013, ISBN: 978-3-7322-8193-0.
- [SPP08] Salinger, S.; Plonka, L.; Prechelt, L.: A Coding Scheme Development Methodology Using Grounded Theory for Qualitative Analysis of Pair Programming. Human Technology: An Interdisciplinary Journal on Humans in ICT Environments 4/1, pp. 9–25, 2008, ISSN: 1795-6889, URL: <http://urn.fi/URN:NBN:fi:jyu-200804151350>.