The Use of Grounded Theory in Empirical Software Engineering

Jeffrey Carver

Empirical software engineering research has much in common with social science research (e.g. Cognitive Science, Psychology). Both types of research focus on understanding how people behave and react in different situations. An important research method for developing hypotheses and insight that is commonly used in these other fields is *grounded theory*. The basic principle behind grounded theory is that the hypotheses and theories emerge bottom-up from the data rather than top-down from existing theory. Using this approach, a researcher begins with an existing data set and abstracts a hypothesis or a theory that accurately describes that data. Then, as more data sets become available, the hypotheses and theories are refined so that they continue to accurately describe all of the extant data.

In empirical software engineering, grounded theory can be very helpful in building a body of knowledge about topics of interest. The grounded theory approach helps researchers develop the right set of questions and hypotheses for a new study or series of studies. This approach can be useful for both the exploratory and confirmatory stages of experimentation. By analyzing the data from existing studies and generating new or refined hypotheses from the bottom up, researches can get an idea both of which hypotheses will be likely to find support in a new study (confirmatory) as well as which hypotheses or research questions have not yet been addressed by the existing data (exploratory). These gaps in the data also provide a useful starting point for planning new studies that will extend the body of knowledge. We have successfully applied this approach to the large body of data on software inspections to generate hypotheses about the impact of an inspector's background and experience on their performance during an inspection.

V. Basili et al. (Eds.): Empirical Software Engineering Issues, LNCS 4336, p. 42, 2007.