# Chapter 2.5
# Multimodal Modeling, Analysis, and Validation of Open Source Software Development Processes*

**Walt Scacchi**
*University of California, Irvine, USA*

**Chris Jensen**
*University of California, Irvine, USA*

**John Noll**
*Santa Clara University, USA*

**Margaret Elliott**
*University of California, Irvine, USA*

## ABSTRACT

Understanding the context, structure, activities, and content of software development processes found in practice has been and remains a challenging problem. In the world of free/open source software development (F/OSSD), discovering and understanding what processes are used in particular projects is important in determining how they are similar to or different from those advocated by the software engineering community. Prior studies have revealed that development processes in F/OSSD projects are different in a number of ways. In this article, we describe how a variety of modeling perspectives and techniques are used to elicit, analyze, and validate software development processes found in F/OSSD projects, with examples drawn from studies of the software requirements process found in the NetBeans.org project.

## INTRODUCTION

In the world of globally dispersed, free/open source software development (F/OSSD), discovering and understanding what processes are used in particular projects is important in determining how they are similar to or different from those advocated by the software engineering community. For

example, in our studies of software requirements engineering processes in F/OSSD projects across domains like Internet infrastructure, astrophysics, networked computer games, and software design systems (Scacchi, 2002, 2004, 2005), we generally find there are no explicit software requirements specifications or documents. However, we readily find numerous examples of sustained, successful, and apparently high-quality F/OSS systems being deployed on a worldwide basis. Thus, the process of software requirements engineering in F/OSSD projects must be different than the standard model of requirements elicitation, specification, modeling, analysis, communication, and management (Nuseibeh & Easterbrook, 2000). But if the process is different, how is it different, or more directly, how can we best observe and discover the context, structure, activities, and content software requirements processes in F/OSSD projects? This is the question addressed here.

Our approach to answering this question uses multimodal modeling of the observed processes, artifacts, and other evidence composed as an ethnographic hypermedia that provides a set of informal and formal models of the software development processes we observe, codify, and document. Why? First, our research question spans two realms of activity in software engineering, namely, software development and software process modeling. So we will need to address multiple perspectives or viewpoints, yet provide a traceable basis of evidence and analysis that supports model validation. Second, given there are already thousands of self-declared F/OSSD projects affiliated with OSS portals like SourceForge.net, Freshmeat.net, and Savannah.gnu.org, then our answer will be constrained and limited in scope to the particular F/OSSD projects examined. Producing a more generalized model of the F/OSS development process studied requires multiple, comparative project case studies, so our approach should be compatible with such a goal (Scacchi, 2002). Last, we want an approach to process modeling that is open to independent analysis, validation, communication, and evolution, yet be traceable to the source data materials that serve as evidence of the discovered process in the F/OSSD projects examined (cf. Kitchenham, Dyba, & Jorgensen, 2004).

Accordingly, to reveal how we use our proposed multimodal approach to model requirements processes in F/OSSD projects, we first review related research to provide the foundational basis for our approach. Second, we describe and provide examples of the modeling modes we use to elicit and analyze the processes under study. Last, we examine what each modeling mode is good for, and what kind of analysis and reasoning it supports.

## RELATED RESEARCH AND APPROACH

There is growing recognition that software requirements engineering can effectively incorporate multiple viewpoints (Finkelstein, Gabbay, Hunter, & Nuseibeh, 1994; Leite & Freeman, 1991; Nuseibeh & Easterbrook, 2000) and ethnographic techniques (Nuseibeh & Easterbrook, 2000; Viller & Sommerville, 2000) for eliciting, analyzing, and validating functional and nonfunctional software system *product* requirements. However, it appears that many in the software engineering community treat the *process* of requirements engineering as transparent and prescriptive, though perhaps difficult to practice successfully. However, we do not know how large distributed F/OSSD projects perform their development processes (cf. Curtis, Krasner, & Iscoe, 1998).

Initial studies of requirements development across multiple types of F/OSSD projects (Scacchi, 2002, 2004) find that OSS product requirements are continuously emerging (Gans, Jarke, Kethers, & Lakemeyer, 2003; Gasser, Scacchi, Penne, & Sandusky, 2003; Truex, Baskerville, & Klein, 1999) and asserted after they have been implemented, rather than relatively stable and