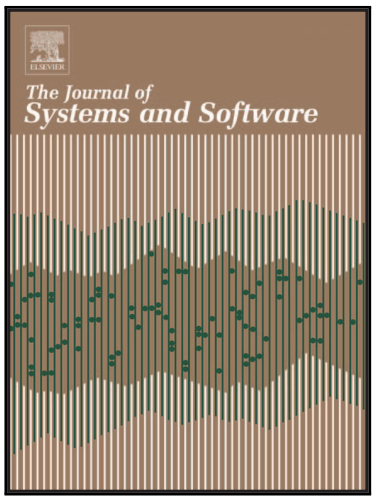

Grounded Requirements Engineering: An Approach to Use Case Driven Requirements Engineering

David Würfel, Rainer Lutz, Stephan Diehl

PII: S0164-1212(15)00227-7
DOI: 10.1016/j.jss.2015.10.024
Reference: JSS 9602

To appear in: *The Journal of Systems & Software*

Received date: 6 October 2014
Revised date: 16 July 2015
Accepted date: 12 October 2015

Please cite this article as: David Würfel, Rainer Lutz, Stephan Diehl, Grounded Requirements Engineering: An Approach to Use Case Driven Requirements Engineering, *The Journal of Systems & Software* (2015), doi: 10.1016/j.jss.2015.10.024

Highlights

- We present a systematic approach for eliciting, analyzing, and modelling requirements.
- It combines established methods from social sciences and software engineering.
- It prevents the three issues discussed in the paper: unawareness, early commitment, and ungroundedness.

Grounded Requirements Engineering: An Approach to Use Case Driven Requirements Engineering

David Würfel, Rainer Lutz, and Stephan Diehl

Department of Computer Science
University of Trier
54296 Trier, Germany
{s4dawuer, lutzr, diehl}@uni-trier.de

Abstract

Requirements engineering produces specifications of the needs or conditions to meet for a software product. These specifications may be vague and ungrounded, i.e. the relation of the requirements to the observations they are derived from may be unclear or not documented. Furthermore, stakeholders may be influenced by solutions of existing software without knowing if these actually suit the software to be developed.

To cope with the above issues, it is important to understand the complete task, before designing a software system to support the task. Thus, we developed a method called Grounded Requirements Engineering (GRE) that leverages the Grounded Theory method to observe and analyze processes and user activities in the real world.

GRE is an iterative process consisting of two steps. First, Grounded Theory methods are used to analyze user experiments or interviews. Second, the resulting abstract descriptions of the user behavior are transferred into use cases. GRE produces comprehensible and grounded requirements for the software system to be built, i.e. the requirements are traceable back to their origins.

In this paper, we provide an elaborate description of the GRE method and illustrate it by applying it to derive requirements for an interactive software tool for model merging. The development of this tool both served as a basis for the design of GRE as well as to test it.

Keywords: software engineering, requirements, elicitation methods, grounded theory

1. Introduction

Requirements engineering (RE) is a difficult and very essential step in software development. Regardless of which process model one follows—an agile or rather strict and traditional one—