Developing a Glossary for Software Projects

S

Tamer Abdou
Suez Canal University, Egypt

Pankaj Kamthan
Concordia University, Canada

Nazlie Shahmir
WestJet Airlines Limited, Canada

INTRODUCTION

There is increasingly significant role played by software in society. This has led to attention on the practices of developing and maintaining software that aim to be successful for all the stakeholders involved.

The discipline of software engineering advocates a systematic and disciplined approach towards the development and evolution of software systems. There is a domain underlying every software project. For a software project to be successful, it is imperative that the domain knowledge be understood and communicated properly to all the stakeholders of that project (Schneider, 2009). Indeed, lack of adequate understanding of the domain has been cited as one of the reasons for software project failures (Kliem, 2007).

The purpose of this chapter is to draw attention to one of the initial steps towards building shareable domain knowledge for a software project, namely a glossary. In doing so, it aims to be relevant to professional as well as pedagogical settings.

The rest of the chapter is organized as follows. First, a motivation and background on glossary are presented, and discussion of relevant previous work is outlined. This is followed by details of a process for developing a glossary. Next, directions for future research are highlighted. Finally, concluding remarks are given.

BACKGROUND

In this chapter, the terms 'software project' and 'project' are considered synonymous, unless otherwise stated. The term 'project' is used to emphasize the fact that the notion of glossary is applicable to a variety of projects, including, but not limited to, software projects. A software project may be about development or about maintenance of a software product.

The following definitions are essential for the rest of the chapter. A *domain* is an area of interest (or the universe of discourse). A *glossary* is a list of terms in a particular domain of knowledge with the definitions for those terms. A *stakeholder* is an individual, group, and/or organization, having an interest in a project.

Glossary in Context

The history of use of glossary in software projects goes back to mid-to-late-1960s, and is therefore is almost as old as the discipline of software engineering itself.

A glossary is similar to, but different from, a dictionary, lexicon, and thesaurus. A comparison can be made using the criteria of goal and scope.

DOI: 10.4018/978-1-5225-2255-3.ch644

Goal

A glossary, like a dictionary, presents its terms (and corresponding definitions) in a lexicographical (alphabetical) order. Also, a glossary, like a thesaurus, may include synonyms of its terms, but does not include antonyms of any terms. For example, in a *Glossary of Requirements Engineering Terminology* (Glinz, 2014), bug, defect, and fault are considered synonymous. However, unlike a lexicon, a glossary usually does not point to etymology of a term.

Scope

A glossary is specific to the scope of a project, while dictionary, lexicon, and thesaurus are more general in scope as implied by the type of information they include.

Motivation for a Glossary

There are a number of (not necessarily mutually exclusive) reasons for having a glossary for any project.

Support for Knowledge Engineering

It has been acknowledged for some time that software engineering is a knowledge-intensive discipline (Robillard, 1999). For example, there is acquisition of knowledge during all stages of software development. The development of a glossary is an initial step towards knowledge engineering (Schneider, 2009).

The purpose of a glossary is to help make implicit domain expert knowledge *explicit* (Dalkir, 2011), help avoid incurring *analysis debt* due to inconsistent terminology, and to ensure that the knowledge of the domain underlying a software project be communicated properly to all the stakeholders of that project.

Communication Among Stakeholders

It has been pointed out in a number of contexts that the quality of communication among stakeholders is crucial for the success of a project (Kliem, 2007). There are several dimensions of the quality of communication, one of which is common understanding of the domain underlying the project. It is therefore crucial to create a placeholder of commonly-agreed upon domain knowledge (Kovitz, 1999).

A project-related discourse, whether expressed verbally or in writing, consists of sentences in some language (usually, natural language). A sentence can include terms that may or may not be understood universally. A glossary makes a discourse meaningful by ensuring that all the stakeholders involved in the discourse are on the "same page". A glossary is a placeholder for commonly-agreed terms (Gottesdiener, 2005).

Figure 1 illustrates the pairwise relationships that can arise between stakeholders and a project artifact, between stakeholders, and the role of a glossary in enabling these relationships.

The need for a glossary is especially acute in organizations where, historically, there has been compartmentalization of departments with more conflict than cooperation between them, but effort is being made to change the trend. For example, *DevOps* is a recent initiative towards a better alignment and working relationship between *development* and *operations* (Hüttermann, 2012). To accomplish that, DevOps extends lessons learned from agile software development and aims to foster close collaboration, culture of shared goals and values, and commitment to measurement and monitoring (Lwakatare, Kuvaja, & Oivo, 2015). For these to manifest successfully, the presence of a common language of communication is a necessary requisite, which a glossary can help provide.