

Chapter 100

Developing a Glossary for Software Projects

Tamer Abdou

Suez Canal University, Egypt

Pankaj Kamthan

Concordia University, Canada

Nazlie Shahmir

WestJet Airlines Limited, Canada

ABSTRACT

The success of a software project depends intrinsically on effective communication among stakeholders. The purpose of a glossary is to ensure that the knowledge of the domain underlying a software project be communicated properly to all the stakeholders of that project. This chapter provides the context, the rationale, and the means for developing a glossary for software projects. In doing so, it proposes a process for developing a glossary. This process is independent of any particular application domain, software development methodology, and information technology. The approaches for representing and presenting a glossary, for the consumption of humans as well as that of machines, are discussed.

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).

DOI: 10.4018/978-1-5225-7598-6.ch100

Developing a Glossary for Software Projects

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 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.

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.

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/developing-a-glossary-for-software-projects/214706

Related Content

LEAP Enhanced: A Lightweight Symmetric Cryptography Scheme for Identifying Compromised Node in WSN

Maleh Yassine and Abdellah Ezzati (2016). *International Journal of Mobile Computing and Multimedia Communications* (pp. 42-66).

www.irma-international.org/article/leap-enhanced/171627

Integrating Mobile Technologies in Enterprise Architecture with a Focus on Global Supply Chain Management Systems

Bhuvan Unhelkar, Ming-Chien Wu and Abbass Ghanbary (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 2368-2390).

www.irma-international.org/chapter/integrating-mobile-technologies-enterprise-architecture/26669

Analysis and Modeling of H.264 Unconstrained VBR Video Traffic

Harilaos Koumaras, Charalampos Skianis and Anastasios Kourtis (2009). *International Journal of Mobile Computing and Multimedia Communications* (pp. 14-31).

www.irma-international.org/article/analysis-modeling-264-unconstrained-vbr/37453

Collaborative Access to Ancient Documents: Towards a Distributed Comparison of Pre-Processing Approaches

Ines Ben Messaoud Ben Arbia, Haikal El Abed, Volker Märgner and Hamid Amiri (2012). *International Journal of Mobile Computing and Multimedia Communications* (pp. 34-53).

www.irma-international.org/article/collaborative-access-ancient-documents/69532

Mobile Learning for Social Change: Democratizing Education and Civic Engagement

Tseday Alehegn and Dominic Mentor (2016). *Handbook of Research on Mobile Learning in Contemporary Classrooms* (pp. 363-377).

www.irma-international.org/chapter/mobile-learning-for-social-change/157989