Chapter 2
A Review on Software Project Management Ontologies

Omiros Iatrellis
Department of Computer Science and Engineering. University of Applied Sciences of Thessaly, Larissa, Greece

Panos Fitsilis
Business Administration Department, University of Applied Sciences of Thessaly, Larissa, Greece

ABSTRACT

This article aims to provide the reader with a comprehensive background for understanding current knowledge and research works on ontologies for software project management (SPM). It constitutes a systematic literature review behind key objectives of the potential adoption of ontologies in PM. Ontology development and engineering could facilitate substantially the software development process and improve knowledge management, software and artifacts reusability, internal consistency within project management processes of various phases of software life cycle. The authors examined the literature focusing on software project management ontologies and analyzed the findings of these published papers and categorized them accordingly. They used qualitative methods to evaluate and interpret findings of the collected studies. The literature review, among others, has highlighted lack of standardization in terminology and concepts, lack of systematic domain modeling and use of ontologies mainly in prototype ontology systems that address rather limited aspects of software project management processes.

INTRODUCTION

Successful Project Management (PM) is widely accepted as an important success factor for the development of large engineering projects. Over time, it proved its value as an important management tool in business development and success. A wide variety of PM methodologies, frameworks and approaches has evolved over the years each with its own recognized strengths and weaknesses. In this context a wide variety of PM frameworks, methodologies and approaches have been developed over the past few decades. Among of the most popular is the "Project Management Body Of Knowledge" (PMBOK) from Project

DOI: 10.4018/978-1-7998-3016-0.ch002

Management Institute (PMI, 2017), the “IPMA Competence Baseline” (ICB) from International Project Management Association (IPMA, 2015) and the “Projects IN a Controlled Environment” (PRINCE2) from the Office of Government Commerce in UK (Office of Government Commerce, 2009).

The principle aim of software engineering is to identify precisely what are the repeatable and reusable procedures in software development, and to support, regulate and automate as many as possible while leaving as little as possible for mental-intensive work. The quality of the software is key feature for success of the any system and the quality product is developed by using different kind of methodologies like Waterfall, Prototyping, RAD (Rapid Application Development), Incremental, Spiral, UP (Unified Process) XP (Extreme Programming), Scrum, etc. (Sommerville, 2013) whereas the selection of the methodology depends on what type of the scenario in which product is being developed. However, it is well established that a significant number of software projects still fail to deliver on time and within target costs and specifications. Due to cost overruns, schedule delays, unfilled requirements and poor quality many IT projects are perceived to be successful (Carroll, 2013; Charette, 2005).

Srikantaiah, Srikantaiah, Koenig, & Al-Hawamdeh, (2010) discussed the benefits of managing knowledge in projects and argued that knowledge management has become a fundamental necessity in PM. Aiming at representing project knowledge explicitly and formally, and sharing and reusing this knowledge among multidisciplinary engineering teams several works build upon ontological engineering as a foundation for capturing implicit knowledge and as a basis of knowledge systematization. In 1995, Gruber defined an ontology as a “formal specification of a shared conceptualization” (Gruber, 1995). This definition required that the conceptualization should express a shared view in a (formal) machine-readable format between several parties, a consensus rather than an individual view. The ontology engineer analyzes relevant entities and organizes them into concepts and relations, being represented, respectively, by unary and binary predicates. The backbone of an ontology consists of a generalization/specialization hierarchy of concepts, i.e., a taxonomy.

An overview of Ontology Engineering, mentioning the most outstanding and used methodologies, languages, and tools for building ontologies can be found in (Suárez-Figueroa, García-Castro, Villazón-Terrazas, & Gómez-Pérez, 2011). Devedzić (2002) surveyed several application classes that benefit from using ontologies, including natural language processing, intelligent information retrieval (especially from the Internet), virtual organizations, and simulation and modeling. Protégé[1] is one of the most famous and widely used ontology construction tools for building intelligent systems, while the most prominent language is the OWL2 (Web Ontology Language) (Motik, Patel-Schneider, & Parsia, 2009), which is also a W3C recommendation.

This paper applies systematic review techniques to survey available literature to identify the most mature studies that address PM as well as, ontologies modeling specific software development lifecycles or addressing specific problem areas of software projects from the software project management perspective.

The rest of the paper is organized as follows; Inn the next section we provide an overview of PM methodologies and of ontological engineering and defines the context of this survey. In the Section after we describe the review process and the protocol underlining this systematic review. We present and discuss the review results and findings and finally we conclude and discuss future work.