W

Web Service in Knowledge Management for Global Software Development

Kamalendu Pal
https://orcid.org/0000-0001-7158-6481
City, University of London, UK

INTRODUCTION

Software system development has transitioned from a solo activity of designing standalone development activities to a distributed and collaborative production that needs a team-based software developer's contribution. Many software project staff now contribute to multiple projects. Due to this work practice, project boundaries blur, not just in terms of their work and how they design and develop software but also their communication channels and knowledge management practice. This way, software development is a knowledge-intensive practice. For example, people work in software development teams to bundle the man-powers and use the systematic approach to share system design knowledge. This collaborative knowledge sharing mechanism is known as '*knowledge management'* in software industries.

Modern software systems play a vital role in shaping significant social challenges (Jansen et al., 2009) (Pal, 2019). Software is an essential value-adding component of most consumer products (e.g., mobile phones, digital music systems, automobiles). Moreover, software systems are also heavily used in the aerospace industry, industrial business process automation, and control systems (Pal, 2020) (Pal & Karakostas, 2020). In these software applications, malfunction or error can cause loss of life or injury, and error-free software is crucial to the safety and wellbeing of people and businesses. Hence, there is an increasing requirement for applying strict engineering discipline to the development of software systems (Pal, 2021a) (Pal, 2021b); and requirements are context sensitive in which the system need-to-be operate (Sawyer et al., 2010) (Sharp et el., 1999).

Consequently, software products must be verified against their requirements throughout the development process like any engineering product. Different software development process models (e.g., Waterfall, Spiral, V-Model) have evolved over the decades to accommodate challenges in software development practice. This way, software development process models play a crucial role to provide a systematic and organized approach to software development (Sommerville, 2019). According to Kevin Roebuck (2012), a traditional Software Development Life Cycle (SDLC) provides the framework for planning and controlling the development or modification of software products, along with the methodologies and process models used for software development.

The design and development of software is a complex process consisting of many interdependent activities that involve many stages such as inception, initial design, detailed design and development, implementation and testing, operation, maintenance, and retirement. This includes requirements analysis, technical development, project management, quality assurance, and customer support activities. In addition, requirement analysis has always been with any human act of design, so it may seem strange

DOI: 10.4018/978-1-7998-9220-5.ch041

that they have been singled out for study in computer science and created a subject area known as requirement engineering.

Requirement engineering is a term used to describe the business processes involved in eliciting, documenting, and maintaining requirements for a software system. It is about discovering what the users need the system to do for them. For example, one can define a requirement as "a specification of what should be implemented". There are two types of requirements: (i) functional requirements – what behaviour the system should offer; and (ii) non-functional requirements – a specific property related to quality assurance, time, or cost of development related issues. They are a statement of *what* the software system should do and not *how* it should.

When making software requirements elicitation and specification, requirement engineers share their experiences; these experiences (i.e., knowledge) need to be used in the software development process. In addition, knowledge in software development results from perception, realization, rational thinking, experience, or innate reasoning ability. Knowledge is intuitive and exists within people, part of human reasoning and decision making. Therefore, knowledge is hard to capture in words or understand entirely logically. Knowledge is a framework for "evaluating and incorporating new experiences and information". It is the basis for the process, which continues along time (Davenport & Prusak, 1998).

It is vital to have a digital infrastructure to manage software development knowledge, which will help the software development team effectively do their work. This chapter will focus on software requirement engineering (SRE) related knowledge management issues. In addition, there are several research works (Wouters, Deridder & Van Paesschen, 2000) (Mayank, Kositsyna & Austin, 2004) (Lasheras et al., 2009) (Kaiya & Saeki, 2006) published in recent years that relates to knowledge management in software requirement engineering.

Requirement engineering business activities typically involve people from at least two distinct fields: (1) business consulting area (e.g., clients and other stakeholders); and (2) software development area (e.g., business engineers, system architects, software project managers). These diverse groups of people often produce information flows and knowledge exchange that need to be captured in an automated way by which global participants can interact with this newly formed system. This automated system needs to accommodate the different UML (Unified Modelling Language) models of proposed software development and to critique these models at different business meetings. In other words, this automated environment will help gather, analyze, and document software requirements. In addition, it will help the global software development actors manage collaboratively and share their knowledge through this global digital platform.

This global software platform will use the Internet, Intranet, or any other computer network to connect with the different stakeholders for software development purposes. Its central constituent part is web service-based technology, and web service can strengthen communication and information exchange within a community. Different web service-based 'web portal' infrastructures appeared to provide an open and effective communication forum for their members. A web portal collects and presents relevant information for the community in a simplistic sense, and users can publish and access events or information to the community.

This new semantic enhanced web service is known as semantic web. Ontologies are the backbone technology for the Semantic Web and, more generally – for managing formalized knowledge in the context of distributed systems. In an ontology-based system, information is made better understandable for the computer application, thus assisting end-users to search, extract, interpret and process information efficiently. Therefore, semantic web technologies can improve the information sharing process by overcoming the problems of standard web portals. Finally, this chapter presents the key features of an

Related Content

Designing a Real-Time Dashboard for Pandemic Management: COVID-19 Using Qlik Sense
Rahul Rai (2021). *Machine Learning and Data Analytics for Predicting, Managing, and Monitoring Disease (pp. 190-203).*
www.irma-international.org/chapter/designing-a-real-time-dashboard-for-pandemic-management/286252

Unravelling Climate Dynamics: Leveraging Machine Learning for Predicting Changes in Climate Behavior
Neha Guptaand Priya Tanwar (2024). *Reshaping Environmental Science Through Machine Learning and IoT (pp. 230-250).*
www.irma-international.org/chapter/unravelling-climate-dynamics/346579

Development of a Charge Estimator for Piezoelectric Actuators: A Radial Basis Function Approach
Morteza Mohammadzaheri, Mohammadreza Emadi, Mojtaba Ghodsi, Issam M. Bahadur, Musaab Zarogand Ashraf Saleem (2020). *International Journal of Artificial Intelligence and Machine Learning (pp. 31-44).*
www.irma-international.org/article/development-of-a-charge-estimator-for-piezoelectric-actuators/249251

Introduction to Bioinformatics and Machine Learning
Rakhi Chauhan (2024). *Applying Machine Learning Techniques to Bioinformatics: Few-Shot and Zero-Shot Methods (pp. 317-332).*
www.irma-international.org/chapter/introduction-to-bioinformatics-and-machine-learning/342731

Multi-Objective Materialized View Selection Using Improved Strength Pareto Evolutionary Algorithm
Jay Prakashand T. V. Vijay Kumar (2019). *International Journal of Artificial Intelligence and Machine Learning (pp. 1-21).*
www.irma-international.org/article/multi-objective-materialized-view-selection-using-improved-strength-pareto-evolutionary-algorithm/238125