Chapter XV
Examining the Quality of Evaluation Frameworks and Metamodeling Paradigms of Information Systems Development Methodologies

Eleni Berki
University of Jyväskylä, Finland

ABSTRACT

Information systems development methodologies and associated CASE tools have been considered as cornerstones for building quality in an information system. The construction and evaluation of methodologies are usually carried out by evaluation frameworks and metamodels - both considered as meta-methodologies. This chapter investigates and reviews representative metamodels and evaluation frameworks for assessing the capability of methodologies to contribute to high-quality outcomes. It presents a summary of their quality features, strengths and weaknesses. The chapter ultimately leads to a comparison and discussion of the functional and formal quality properties that traditional meta-methodologies and method evaluation paradigms offer. The discussion emphasizes the limitations of both methods and meta-methods to model and evaluate software quality properties such as computability and implementability, testing, dynamic semantics capture, and people's involvement. This analysis along with the comparison of the philosophy, assumptions, and quality perceptions of different process methods used in information systems development, provides the basis for recommendations about the need for future research in this area.

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

In traditional software engineering, the information systems development (ISD) process is defined as a series of activities performed at different stages of the system lifecycle in conformance with a suitable process model (method or methodology). In the fields of Information Systems and Software Engineering, the terms methodology and method are often used interchangeably (Nielsen, 1990; Berki et al., 2004). Increasingly, new methods, techniques and automated tools have been applied in Software Engineering (SE) to assist in the construction of software-intensive information systems. Quality frameworks and metamodels are mainly concerned with the evaluation of the quality of both the process itself and the resulting product at each stage of the life cycle including the final product (the information system).

Professional bodies such as IEEE and ISO have established quality standards and software process management instruments such as Software Process Improvement and Capacity dEtermination (SPICE) (Dorling, 1993) and Capability Maturity Model (CMM) (Paulk et al., 1993) have focused on the quality properties that the ISD process should demonstrate in order to produce a quality information system (Siakas et al., 1997). However, software quality assurance issues (Ince, 1995) such as reliability (Kopetz, 1979) and predictability, measurement and application of software reliability in particular (Myers, 1976; Musa et al., 1987), have long preoccupied software engineers, even before quality standards.

IS quality improvement can be achieved through the identification of the controllable and uncontrollable factors in software development (Georgiadou et al., 2003). ISD methodologies and associated tools can be considered as conceptual and scientific ways to provide prediction and control; their adoption and deployment, though, by people and organizations (Iivari & Huisman, 2001) can generate many uncontrollable factors. During the last thirty-five years, several methodologies, techniques and tools have been adopted in the ISD process to advance software quality assurance and reliability. A comprehensive and detailed coverage of existing information systems development methodologies (ISDMs) has been carried out by Avison & Fitzgerald (1995), with detailed descriptions of the techniques and tools used by each method to provide quality in ISD.

Several ISDMs exist. Berki et al., (2004) classified them into families, highlighting their role as quality assurance instruments for the software development process. They have been characterized as hard (technically oriented), soft (human-centered), hybrid (a combination of hard and soft), and specialized (application-oriented) (Berki et al., 2004). Examples of each include:

- Hard methods - object-orientated techniques, and formal and structured families of methods;
- Soft methods - Soft Systems Method (SSM) and Effective Technical and Human Implementation for Computer-based Systems (ETHICS)
- Hybrid methods - Multiview methodology, which is a mixture of hard and soft techniques;
- Specialized methods - KADS, extreme programming (XP) and other agile methods.

The contribution of these methods to the quality of the ISD process has been a subject of controversy; particularly so because of the different scope, assumptions, philosophies of the various methods and the varied application domains they serve. For example, it is believed that the human role in ISD bears significantly on the perception of the appropriateness of a method (Rantapuska et al., 1999); however, usability definitions in ISO standards are limited (Abran et al., 2003). There is empirical support for the notion that a methodology is as strong as the user involvement it supports (Berki et al., 1997).

16 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/examining-quality-evaluation-frameworks-metamodeling/23795

Related Content

Towards a Conceptual Framework for Security Requirements Work in Agile Software Development
Inger Anne Tøndeland Martin Gilje Jaatun (2022). *Research Anthology on Agile Software, Software Development, and Testing (pp. 247-279).*
www.irma-international.org/chapter/towards-a-conceptual-framework-for-security-requirements-work-in-agile-software-development/294467

Machine Learning-Based Electricity Load Forecast for the Agriculture Sector
Megha Sharma, Namita Mittal, Anukram Mishraand Arun Gupta (2023). *International Journal of Software Innovation (pp. 1-21).*
www.irma-international.org/article/machine-learning-based-electricity-load-forecast-for-the-agriculture-sector/315735

Analyzing Impacts on Software Enhancement Caused by Security Design Alternatives with Patterns
Takao Okubo, Haruhiko Kaiyaand Nobukazu Yoshioka (2012). *International Journal of Secure Software Engineering (pp. 37-61).*
www.irma-international.org/article/analyzing-impacts-software-enhancement-caused/64194

Engineering a Service-Oriented Architecture in E-Government
Marijn Janssen (2005). *Service-Oriented Software System Engineering: Challenges and Practices (pp. 340-352).*
www.irma-international.org/chapter/engineering-service-oriented-architecture-government/28962

Performance Evaluation of Different Machine Learning Algorithms Using Credit Scoring Model
Amrit Singh, Harisankar Mahapatra, Anil Kumar Biswal, Milan Samantarayand Debabrata Singh (2023). *The Software Principles of Design for Data Modeling (pp. 248-260).*
www.irma-international.org/chapter/performance-evaluation-of-different-machine-learning-algorithms-using-credit-scoring-model/330500