Information and Software Technology 56 (2014) 252–271

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Understanding the characteristics of quality for software engineering processes: A Grounded Theory investigation

Trent A. Kroeger [1], Neil J. Davidson *, Stephen C. Cook

Defence and Systems Institute, University of South Australia, Mawson Lakes, South Australia 5095, Australia

ARTICLE INFO

Article history:
Received 5 December 2012
Received in revised form 14 October 2013
Accepted 14 October 2013
Available online 24 October 2013

Keywords:
Software engineering processes
Quality attributes
Quality properties
Grounded Theory

ABSTRACT

Context: Software engineering organizations routinely define and implement processes to support, guide and control project execution. An assumption underlying this process-centric approach to business improvement is that the quality of the process will influence the quality, cost and time-to-release of the software produced. A critical question thus arises of what constitutes quality for software engineering processes.

Objective: To identify criteria used by experienced practitioners to judge the quality of software engineering processes and to understand how knowledge of these criteria and their relationships may be useful for those undertaking software process improvement activities.

Method: Interviews were conducted with 17 experienced software engineering practitioners from a range of geographies, roles and industry sectors. Published reports from 30 software process improvement case-studies were selected from multiple peer-reviewed software journals. A qualitative Grounded Theory-based methodology was employed to systematically analyze the collected data to synthesize a model of quality for software engineering processes.

Results: The synthesized model suggests that practitioners perceive the overall quality of a software engineering process based on four quality attributes: suitability, usability, manageability and evolvability. Furthermore, these judgments are influenced by key properties related to the semantic content, structure, representation and enactment of that process. The model indicates that these attributes correspond to particular organizational perspectives and that these differing views may explain role-based conflicts in the judgement of process quality.

Conclusion: Consensus exists amongst practitioners about which characteristics of software engineering process quality most influence project outcomes. The model presented provides a terminological framework that can facilitate precise discussion of software engineering process issues and a set of criteria that can support activities for software process definition, evaluation and improvement. The potential exists for further development of this model to facilitate optimization of process properties to match organizational needs.

1. Introduction

The influence of software in today's world is enormous, and pervades almost every aspect of modern society. Software has moved from being considered a highly-specialized product to being a common and essential commodity. This commoditization has resulted in the software industry coming under increasing pressure to develop and deliver greater volumes of high-quality products and services within cost and schedule constraints that are tighter than ever before [4].

Increasingly, software engineering organizations are looking to software engineering processes as a means to support, guide and, where necessary, control software project activities [20]. To this end, many companies have invested significant resources in improvement programs with the objective of implementing processes to support their business [42,48,46,45]. This investment has led to a proliferation of methods and tools for software engineering process definition, enactment, evaluation and improvement.

A fundamental principle of process-based approaches to software engineering management is that the quality of the process will influence the quality, cost and time-to-release of the software produced. For such approaches, a critical question arises of what

* Corresponding author. Tel.: +61 8 8302 5885.
E-mail addresses: trent.kroeger@adelaide.edu.au (T.A. Kroeger), Neil.Davidson@unisa.edu.au (N.J. Davidson).
URL: http://www.adelaide.edu.au/directory/trent.kroeger (T.A. Kroeger).
[1] Present address: School of Computer Science, University of Adelaide, Adelaide 5000, Australia.

0950-5849/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2013.10.003

constitutes quality for a software engineering process, both in general and within the context of a given business environment [13].

Attempts to answer this question are made significantly more difficult by the often divergent views held by software engineering process stakeholders of the attributes that constitute quality for a particular process. This presents a significant challenge to process improvement specialists who are tasked with developing and refining processes that balance the needs of those stakeholders. Further complicating the situation is that stakeholder views are rarely explicitly stated and that existing methods for software process evaluation are therefore forced to adopt either generic or very narrow views of process quality.

A small body of research has emerged to attempt to address this question. This research advances the argument that a multifaceted view of quality is required to enable the development of software engineering processes that will balance the needs of diverse stakeholder groups.

This paper presents the results from research to investigate practitioner perspectives of quality with respect to software engineering processes. The objectives of this research were to better understand the criteria that practitioners use to judge the quality of software engineering processes and to show how an awareness of these criteria may be used to enhance methods for software engineering process definition, enactment, evaluation and improvement. To this end, a Grounded Theory qualitative research methodology was applied that involved the collection and analysis of source data from thirty published software process related case study reports and seventeen interviews with experienced software engineering practitioners. The result of this analysis is a perspective-based model of quality for software engineering processes that consists of distinct quality attributes, quality properties, contextual factors and relationships.

The remainder of the paper is structured as follows. Section 2 sets the context for the discussion by examining the endeavor of software engineering from a systemic perspective, with a particular focus on the role played by defined software engineering processes within such a system. It then considers software engineering process research with respect to the concept of quality. Section 3 describes the design of the research project that was conducted. Firstly, specific research questions are defined that were used to guide the research project. It then describes the theoretical perspective taken throughout the project and the qualitative research methodology applied to obtain answers to the research questions. Section 4 presents the main results of the paper, which are centered around a perspective-based model of quality for software engineering processes. Section 5 provides a discussion of these results with respect to related literature, identifies potential areas of application and suggests areas for further development of the model. Finally, Section 6 summarizes the key findings and contributions from the research.

2. Literature review

Considerable research has been conducted with respect to the definition and implementation of software engineering processes. A common theme throughout this research is the recognition that personal characteristics and systemic interactions are of vital importance within software engineering. This section explores the field of software engineering from these perspectives. The section begins by discussing the systemic nature of software engineering, and the primary role played by people within the system. Implications of this systemic view are then explored with respect to methods for the evaluation of software engineering process quality. Finally, several multi-attribute models of quality for software engineering processes are critically reviewed to establish the justification for the research described herein.

2.1. Systemic nature of software engineering

Software engineering is a complex endeavor that requires the interaction of people, technology and processes in order to develop and maintain a software product. This section describes the nature of software engineering within a systemic context, with a particular focus on the roles of people and processes within such a system. The section begins by establishing the nature of software engineering as a human-activity system; it then highlights the role that processes play within such a system and concludes with a discussion of the implications that a systemic view has on the task of software process engineering.

2.1.1. Software engineering as a human activity system

Fig. 1 depicts processes as static elements within a software engineering system. Within such a system, people selectively interpret processes and apply technologies to transform customer needs into software products. Often, the same software engineering system may produce multiple software products concurrently from different sets of customer needs. Furthermore, the system operates within a particular – and perhaps changing – business environment and is consequently influenced by a range of economic, social, technological and political factors.

Although it is sometimes useful to view the act of software engineering from a technological or process perspective, it is important to note the primacy of people within such a system. Processes and technology serve a purpose within the system, but it is

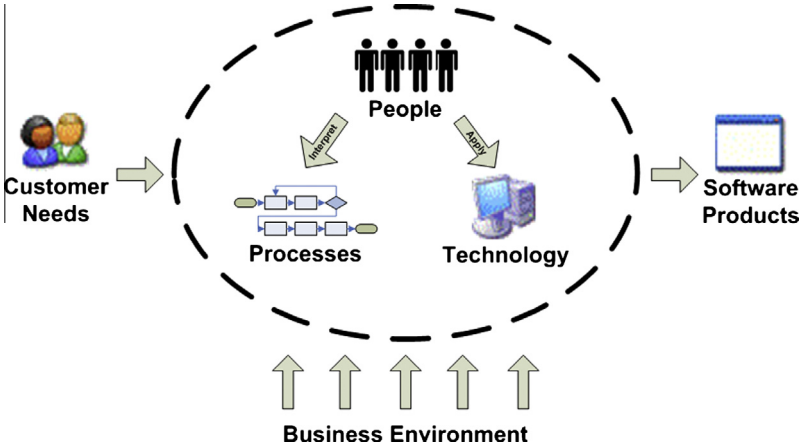

Fig. 1. A model of a software engineering system.