



## Cost, benefits and quality of software development documentation: A systematic mapping



Junji Zhi<sup>a</sup>, Vahid Garousi-Yusifoglu<sup>b,c,\*</sup>, Bo Sun<sup>d,e</sup>, Golar Garousi<sup>c,f</sup>,  
Shawn Shahnewaz<sup>c</sup>, Guenther Ruhe<sup>c,d</sup>

<sup>a</sup> Department of Computer Science, University of Toronto, Ontario, Canada

<sup>b</sup> System and Software Quality Engineering Research Group (SySoQual), Department of Software Engineering, Atilim University, İncek, Ankara, Turkey

<sup>c</sup> Department of Electrical and Computer Engineering, University of Calgary, Alberta, Canada

<sup>d</sup> Department of Computer Science, University of Calgary, Alberta, Canada

<sup>e</sup> iSolutions Inc., Calgary, Alberta, Canada

<sup>f</sup> geoLOGIC Systems Ltd., Calgary, Alberta, Canada

### ARTICLE INFO

#### Article history:

Received 5 September 2012

Received in revised form

28 September 2014

Accepted 28 September 2014

Available online 22 October 2014

#### Keywords:

Software documentation

Documentation benefit

Systematic mapping

### ABSTRACT

**Context:** Software documentation is an integral part of any software development process. Researchers and practitioners have expressed concerns about costs, benefits and quality of software documentation in practice. On the one hand, there is a lack of a comprehensive model to evaluate the quality of documentation. On the other hand, researchers and practitioners need to assess whether documentation cost outweighs its benefit.

**Objectives:** In this study, we aim to summarize the existing literature and provide an overview of the field of software documentation cost, benefit and quality.

**Method:** We use the systematic-mapping methodology to map the existing body of knowledge related to software documentation cost, benefit and quality. To achieve our objectives, 11 Research Questions (RQ) are raised. The primary papers are carefully selected. After applying the inclusion and exclusion criteria, our study pool included a set of 69 papers from 1971 to 2011. A systematic map is developed and refined iteratively.

**Results:** We present the results of a systematic mapping covering different research aspects related to software documentation cost, benefit and quality (RQ 1–11). Key findings include: (1) validation research papers are dominating (27 papers), followed by solution proposals (21 papers). (2) Most papers (61 out of 69) do not mention the development life-cycle model explicitly. Agile development is only mentioned in 6 papers. (3) Most papers include only one “System under Study” (SUS) which is mostly academic prototype. The average number of participants in survey-based papers is 106, the highest one having approximately 1000 participants. (4) In terms of focus of papers, 50 papers focused on documentation quality, followed by 37 papers on benefit, and 12 papers on documentation cost. (5) The quality attributes of documentation that appear in most papers are, in order: completeness, consistency and accessibility. Additionally, improved meta-models for documentation cost, benefit and quality are also presented. Furthermore, we have created an online paper repository of the primary papers analyzed and mapped during this study.

**Conclusion:** Our study results show that this research area is emerging but far from mature. Firstly, documentation cost aspect seems to have been neglected in the existing literature and there are no systematic methods or models to measure cost. Also, despite a substantial number of solutions proposed during the last 40 years, more and stronger empirical evidences are still needed to enhance our understanding of this area. In particular, what we expect includes (1) more validation or evaluation studies; (2) studies involving large-scale development projects, or from large number of study participants of various organizations; (3) more industry-academia collaborations; (4) more estimation models or methods to assess documentation quality, benefit and, especially, cost.

© 2014 Elsevier Inc. All rights reserved.

\* Corresponding author at: System and Software Quality Engineering Research Group (SySoQual), Department of Software Engineering, Atilim University, İncek, Ankara, Turkey. Tel.: +90 (312) 586-8774.

E-mail addresses: [zhij@cs.toronto.edu](mailto:zhij@cs.toronto.edu) (J. Zhi), [vahid.garousi@atilim.edu.tr](mailto:vahid.garousi@atilim.edu.tr) (V. Garousi-Yusifoglu), [sbo@ucalgary.ca](mailto:sbo@ucalgary.ca) (B. Sun), [golar.garousi@gmail.com](mailto:golar.garousi@gmail.com) (G. Garousi), [smshahne@ucalgary.ca](mailto:smshahne@ucalgary.ca) (S. Shahnewaz), [ruhe@ucalgary.ca](mailto:ruhe@ucalgary.ca) (G. Ruhe).

## 1. Introduction

Software documentation is an integral part of any software development process (Bayer and Muthig, 2006). In fact, software documentation has become a popular sub-domain in software engineering (Cook and Visconti, 1994) to the extent that there are special interest groups such as the ACM Special Interest Group on Design of Communication (SIGDOC).

A literature search in the beginning of this study (Fall 2013) yielded 500+ papers on software documentation. A large portion of this set proposes various types of documentation management systems or formats. Another portion of the paper set focuses on cost, benefits and quality of documentation, the subjects which we focus on in this study.

In our study, we target the documents that are software development related. We call them *technical* and refer to those documents that (1) are produced during the software development lifecycle and (2) whose target audience(s) are software developer(s). The types of documents within the scope of our investigation typically include requirement, design, implementation and test documents as well as code comments. Product or user manuals may also be produced during development lifecycle, but are excluded in our investigation because it violates the second criterion, i.e., their target audiences are not software developers. We define the term *cost* as the value of effort or time that has been used to produce a software artifact (e.g., code, or documentation).

A considerable share of software projects' costs are spent on documentation, e.g., a ratio of 11% was reported in (Sanchez-Rosado et al., 2009). This indicates that the effort consumed in documentation is one significant cost drivers during software development processes. It is natural and expected that, when cost is spent in developing an artifact, that artifact should be used and provides *benefit* at some point in the development or maintenance phase (Mira, 2005; Tilley and Huang, 2003; Lethbridge et al., 2003). The benefits could be reflected in many aspects, e.g., shortened task duration, improved code quality, higher productivity, or any other improvements related to software development. In terms of documentation *quality*, we define it as the character of documents with respect to fineness which is often influenced by how much time/effort is spent on and affects the benefits practitioners get from the documents. Therefore, the aspect of document quality is also included in our scope of study.

On the other hand, the traditional view of software documentation is undergoing the challenge of Agile development methods (Ambler, 2011; Rubin and Rubin, 2011; Rueping, 2003; Stettina and Heijstek, 2011). As the *Agile manifesto* (Beck et al., 2012) points out: "Working software [is valued] over comprehensive documentation". The manifesto also mentions that, while there is value in the items on the right (i.e., documentation), we value the items on the left (i.e., working software) more. Does this mean documentation is no longer important (Stettina and Heijstek, 2011)? Practitioners start to question whether the cost of creating and maintaining documentation outweighs its potential benefit (Rueping, 2003; Stettina and Heijstek, 2011). To answer such a question, one needs to be able to quantitatively measure the cost and benefit of documentation.

During the past three to four decades, researchers, in increasing numbers, have proposed different techniques for analyzing cost, benefit and quality of documentation. As the research area matures and the number of related papers increases, we feel it is important to summarize the current state-of-the-art and provide an overview of the trends in this specialized field. To address that goal, we present in this paper a *systematic mapping* of the literature in this area.

According to Petersen et al. (2008), a systematic mapping (SM) is a method to review, classify, and structure papers related to

a specific research field in software engineering. According to Kitchenham et al. (2011): "mapping papers can save time and effort for researchers and provide baselines to assist new research efforts". The goal is to obtain an overview of existing approaches, outlining the coverage of the research field in different facets of the classification scheme that we develop in this paper. Identified gaps in the field serve as a valuable basis for future research directions. Using an empirical study, Kitchenham et al. (2010) reported that SM papers also have educational values and would provide young researchers and students with useful and transferable research skills and are a useful first step for postgraduate PhD candidates.

Unlike a Systematic Literature Review (SLR) (Kitchenham and Charters, 2007), finding evidence for impact of a proposed approach is not the main focus in a systematic mapping (Petersen et al., 2008). However, the two methods have many overlaps and the results of a systematic mapping can be fed into a more rigorous systematic review study to support evidence-based software engineering (Kitchenham and Charters, 2007).

Systematic mapping papers generally consist of five steps including: (1) a definition of research questions, (2) conducting the search for relevant papers, (3) screening of papers, (4) key-wording of abstracts, and (5) data extraction and mapping (Petersen et al., 2008), which we follow in this paper.

As far as we are concerned, we have not been able to find any study to synthesize or to systematically map the existing papers on software documentation cost, benefit and quality. Our study aims to survey the existing literature for purpose of identifying research trends. We hope that this paper contributes a summary of the area that could be useful for follow-up future papers. Also, the need for this SM was motivated in the context of a multi-year industrial collaborative research and development project in which the authors are involved in, which aims to minimize the cost and amount of documentation across the software development life-cycle for one of our industrial partners.

The main questions we intend to answer in this study are:

- (1) How do researchers assess the quality of documentation?
- (2) What are the cost-related attributes of software documentation?
- (3) What benefit does documentation bring to software practitioners?

During our SM study, we have extracted the attributes or metrics to measure these three aspects. For document quality aspect, we extracted more than 13 attributes that cover different aspects of document quality, including up-to-date-ness, completeness, etc. For benefit aspect, we also gathered three main categories (e.g., development aid, maintenance aid, etc.) and two metrics (e.g., task time reduction, etc.). In terms of document cost, we also extracted two main categories (i.e., production or maintenance cost, etc.) and one quantitative metric (i.e., document size). The results are presented in detail in Sections 6.8–6.10.

The main contributions of this paper are two-fold:

- A unified meta-model for documentation quality incorporating and consolidating all the individual and partial parts proposed by previous researchers, and also a meta-model for documentation usage process and benefit (Section 5.2).
- A systematic map (Section 5) developed for the area of documentation cost, benefit, and quality and consequently the systematic mapping of the existing research in this area (Section 6).

Also, we published an online paper repository which has been created during this systematic study (Zhi et al., 2012). Future researchers or practitioners can find related works in the area

Download English Version:

<https://daneshyari.com/en/article/461044>

Download Persian Version:

<https://daneshyari.com/article/461044>

[Daneshyari.com](https://daneshyari.com)