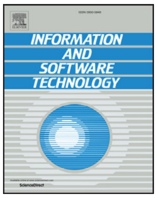
Information and Software Technology 73 (2016) 101–121

Contents lists available at ScienceDirect

Information and Software Technology

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



A systematic literature review on software measurement programs

Touseef Tahir[a], Ghulam Rasool[a,*], Cigdem Gencel[b]

[a] COMSATS Institute of Information Technology, Department of Computer Science, Lahore, Pakistan
[b] DEISER, Madrid, Spain

ARTICLE INFO

Article history:
Received 4 March 2015
Revised 20 January 2016
Accepted 22 January 2016
Available online 6 February 2016

Keywords:
Software measurement
Software measurement program
Software metrics
GQM
Systematic Literature Review

ABSTRACT

Context: Software measurement programs (MPs) are an important means for understanding, evaluating, managing, and improving software processes, products and resources. However, implementing successful MPs still remains a challenge.
Objectives: To make a comprehensive review of the studies on MPs for bringing into light the existing measurement planning models and tools used for implementing MPs,the accumulated knowledge on the success/failure factors of MPs and mitigation strategies to address their challenges.
Methods: A Systematic Literature Review (SLR) was conducted. In total, 65primary studies were reviewed and analyzed.
Results: We identified 35 measurement planning models and 11 associated tools, most of which either proposed extensions or improvements for goal based approaches. The identified success factors include (a) organizational adoption of MP, (b) integration of MP with SDLC, (c) synchronization of MP with SPI and (d) design of MP. The mostly mentioned mitigation strategies for addressing challenges are effective change management and measurement stakeholder management, automated tool support and incorporation of engineering mechanisms for designing sustainable, effective, scalable and extendible MPs, and measurement expertise and standards development.
Conclusion: Most of the success factors and mitigation strategies have interdependencies. Therefore, for successful MP implementation, software organizations should consider these factors in combination and make a feasibility study at the very beginning.

1. Introduction

The evolution of systems and software engineering has increased the demand of software intensive products. Software has become a key entity for survival of products, systems and services by providing competitive advantage, increased market-share, reworks reduction, and improved resource utilization [1]. Measurement, as in all other engineering fields, is vital in software engineering to characterize, evaluate or predict software entities such as processes, products and resources for assessing, monitoring, controlling and improving processes, product quality, estimation accuracy and productivity [2–4]. As Tom DeMarco [5] said “You cannot control what you cannot measure”, software measurement is pervasive as every single software process generates data and/or uses data generated about processes, products and resources [3,6–8].

ISO/IEC 15939:2007 [9] defines measurement process as “a process for establishing, planning, performing and evaluating measurement within an overall project, enterprise or organizational measurement structure”. Software companies implement Measurement Programs (MPs) as part of their measurement processes [3]. For planning a MP, organizations usually make use of some standards (e.g. ISO/IEC 15939:2007 [9], CMMI [10–11], ISO/IEC 25021 [12], ISO 9126 standard family [13–15], and ISO/IEC 14598-x [16]) procedures (e.g. based on experiences, heuristics and company policy), measurement planning models and tools andmeasurement stakeholders who input and/or use the data [3].

Briand et al. [4] reported in 1996 that implementation of successful MPs had been a challenge since last fifteen years. The results of this study show that successful implementation of MPs is still a challenge. MPs most of the time lack sustainability, clear objectives, correct measurement instruments, resources, time and budget e.g. [2,17,18]. In [81], it is reported that 80% of MPs implemented for 610 projects could not sustain more than two years and 78% of MPs were failed to achieve measurement objectives. The reasons of failure were reported as the lack of link between MP and business goals, organizational buy-in and commitment and

[*] Corresponding author: Tel.:+92 3334520196; fax: +92 4299203100.
E-mail address: grasool@ciitlahore.edu.pk (G. Rasool).

http://dx.doi.org/10.1016/j.infsof.2016.01.014
0950-5849/© 2016 Elsevier B.V. All rights reserved.

lack of synchronization between MP findings and corrective actions [19–21]. Mendonça and Basili [22,23] stated that soundness of metrics, completeness of the collected data, leanness in collection and consistency with the user goals are essential properties of a good MP. Kitchenham [24] mentioned the importance of standardization and validation of metrics. Niessink and van Vliet [25] stated that metrics should create value in a software company. Goal oriented measurement, reusability, measurement patterns and tool support are reported as success factors for implementation of MP [1,26–28].

For MP implementation, there are a number of measurement planning models, tools and practices in the literature. For example, Goal Question Metrics (GQM) [29–31], Goals Questions Indicators Measures (GQIM) [32], Measurement Information Model in ISO/IEC 15939:2007 [9] GQM+ Strategies [33], Goal Argument Metrics (GAM) [34], Measure Manage Paradigm (M3P) [35] and Balanced Scorecard [36]. A number of studies proposed enhancements and extensions to existing models such in [37–41] for addressing some of the weaknesses of these models in practice such as narrow process definition and guidance, lack of consistent terminology, poor traceability between GQM elements. According to [42], while goal oriented measurement has been repetitively applied in industry (e.g. NASA, Motorola, HP and AT&T), there exists no comprehensive analysis on measurement planning models, tools, success/failure factors, challenges, best practices and metrics selection methods for MPs.

This study presents a Systematic Literature Review (SLR) on software MPs. In our previous study, we made a SLR on measurement process in general and with a limited analysis [43]. This paper is an extension of the previous study covering the studies published between 1997 and 2014 and providing a comprehensive review also answering additional research questions (RQs). The RQ for this SLR are:

RQ1: Which measurement planning models, tools and practices are discussed in the literature?

RQ2: Which techniques/methods/models are developed for metrics selection when implementing MPs?

RQ3: What are the success and failure factors for MPs implementation?

RQ4: Which mitigation strategies are discussed for MP implementation?

This SLR has two main contributions to researchers and practitioners: (1) to identify further improvement opportunities for MP implementation and (2) to help in implementing successful MPs through accumulated experience and knowledge in the literature. Answers to RQ1 would help to build an understanding of existing MP planning models, tools and practices as well as their use in different cases with respect to level of MP implementation (e.g. project, organization), which software entities (e.g. processes, products, resourcesthey focusedon and main measurement purposes (characterize, evaluate, predict and improve). In particular, we aim to reveal current practices. RQ2 would further answer which metrics selection methods and models are discussed in the literature and help in identifying their specific challenges in practice. Answers to RQ3 and RQ4 present a comprensive review and analysis of success/failure factors of MPs and mitigation strategies to address the challenges met by companies, which can help companies building pro-active strategies against potential risks and researchers to develop methods and tools to address the challenges.

This paper is structured as follows: Section 2 presents a summary of the related work. The details of the SLR (plan, conduct and analysis and results) are presented in Section 3. The conclusions are given in Section 4.

2. Related work

Unterkalmsteiner et al. [2] made a SLR to analyze types of metrics collected for software process improvement (SPI) and evaluation and measurement strategies. In total, they analyzed 148 primary studies that were published between 1991 and 2008. The studies were classified with respect to their focus, measurement perspective, process quality, estimation accuracy, productivity, product quality, effort, defects, cost, time-to-market, return-on-investment, customer satisfaction and 'other' qualitative success indicators. The results of the SLR showed that 39% of the studies focused on measurement of quality, 38% on estimation accuracy and 35% on productivity. The papers are also classified according to ISO-9126-1 quality attributes i.e. reliability, maintainability, reusability, usability, portability, efficiency, functionality. Three categories of measurement are identified i.e. project, product and organization. The projects and project-product perspectives are identified as highly discussed perspectives in measurement research. The Capability Maturity Model (CMM) is the mostly studies model in the SPI domain. The problems related to SPI studies such as the primary studies did not provide the complete context of case studies; the confounding factors (threats) to case studies that can affect the implementation and evaluation of SPI are not discussed completely. The challenges are such as consistent, unambiguous definition of metrics; appropriate metrics selection for improvement evaluation. They concluded that the scope of measurement before and after evaluation for SPI should be defined and long-term effects such as customer satisfaction should bemeasured. The RQs answered in this studydo not discuss the role of MPs for SPI, but rather the role of measurement for SPI. Our study provides an analysisof the MPs with respect to characterization, evaluation, improvement and prediction in the context of MPs.

Kitchenham [24] made a SLR to analyze studies published on software metrics and investigated the possibility of aggregation of results. She identified 25 primary studies that were published between 2000 and 2005. She found out that journal papers impact software measurement community more as compared to conference papers, and that research on software MPs is one of the most studied topics. The study concluded that there is need of aggregation and comparison of results reported in research. There is also a need of providing the reference to context and using industrial datasets to highlight and address the problems faced regarding software measurement in industry.

Catal et al. [44] made a SLR that analyzed the role of metrics in fault prediction studies. The study included 74 primary studies published between 1990 and 2007. They classified the primary studies according to methods used for fault prediction, i.e. machine learning methods/algorithms, 'statistical and machine learning' methods and expert judgment. The machine learning and statistics are found to be the most widely used methods for software measurement. Furthermore, fault prediction metrics were classified with respect to method, class, component, file, process and quantitative values levels.

Gómez et al. [45] made a SLR to answer what to measure, how to measure and when to measure. The study included 78 primary studies. They classified the identified metrics according to the type of entity measured: project, process, or product. Furthermore, they also identified whether the attributes measured were internal or external attributes. They reported that 79% of the primary studies discuss product metrics, 12% project metrics and 9% process metrics. The complexity and size attributes were found to be the mostly measured attributes. The identified metrics are also mapped according to project life cycle. 48% of the primary studies were found to be focused on the initial phase, 36% on the intermediate phase and 16% on the final phase. They also found that