Information Sciences 180 (2010) 441–451

Contents lists available at ScienceDirect

Information Sciences

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

Choquet integral based aggregation approach to software development risk assessment

Gülçin Büyüközkan [a,*], Da Ruan [b,c]

[a] Galatasaray University, Department of Industrial Engineering, 34357 İstanbul, Turkey
[b] The Belgian Nuclear Research Centre (SCK·CEN), Boeretang 200, 2400 Mol, Belgium
[c] Department of Applied Math and Computer Science, Ghent University, 9000 Gent, Belgium

ARTICLE INFO

Article history:
Received 28 January 2009
Received in revised form 12 September 2009
Accepted 16 September 2009

Keywords:
Software development risks
Project risk assessment
Multi-attribute decision-making
Choquet integral
Aggregation

ABSTRACT

Software is a crucial component of today's business environment, and a superior risk management effort is required to adeptly steer software development projects. Software development risks are inherently dependent, in other words, mutually positive or negative assessments of some risks can influence the decision to accept or reject a project. This fact cannot be modeled with a traditional best compromise seeking method. Aggregation operations based on the family of fuzzy integrals include many operators and thus can express a variety of decision maker behaviors. This study proposes an integrated multi-criteria evaluation methodology for software development experts and managers to better enable them to position their projects in terms of the associated risks. The method relies on a special fuzzy operator, namely a two-additive Choquet integral that enables modeling various effects of importance and interactions among risks. The potential of the proposed methodology is exposed through a case study conducted in a Turkish software company.

1. Introduction

Software development projects often exceed time and budgetary constraints and potentially do not meet all user requirements [27]. While there are many factors that may result in software failure, inadequate risk control is undoubtedly one of the leading factors [2,35,43]. Software development risk applies to the uncertainties faced by a project manager in achieving a software development project's goals in terms of time, cost, and performance. Software engineering aims to improve the software development process, particularly in controlling the source of the project development risks and in producing quality software for both organizations and users [4,20]. Existing methods, such as simulation or option theory, may support risk management in software development projects [8,25]. To begin reducing the system development failure rate, researchers attempted to identify various risk factors while there was still time to take action. This involved looking into the future and considering the path chosen from a risk perspective [9,10,24]. Likewise, research on failed software projects showed that "their problems could have been avoided or strongly reduced if there had been an explicit early concern with identifying and resolving their high-risk elements" [4].

Since software development risks are numerous and sometimes conflicting, our assessment procedure is based on a multi-attribute decision-making (MADM) approach. In most of the MADM methods, partial evaluations of a solution are aggregated under the assumption of independent decision criteria. However, in software risk assessment and also in many practical applications, criteria present some interaction, and the discrete Choquet integral (CI) has proven to be a useful

* Corresponding author. Tel.: +90 212 227 4480x428; fax: +90 212 259 5557.
E-mail addresses: gulcin.buyukozkan@gmail.com, gbuyukozkan2003@yahoo.com (G. Büyüközkan), druan@sckcen.be, da.ruan@ugent.be (D. Ruan).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.09.009

aggregation operator to model those dependencies [21–23,32,46,49]. Recently, Tsai and Lu [44] applied this approach to evaluate the service quality of e-stores. In another study, Saad et al. [40] used CI to aggregate several attributes in the objective function of a flexible job-shop scheduling problem. Finally, Berrah et al. [3] defined an optimization model to identify the dimensions of improvement for an industrial performance measurement system based on CI while considering the allocated resources.

Recently, Narukawa and Torra [37] described the role of fuzzy measures and integrals in decision making and reviewed some computational aspects, which were important for building real applications. They also considered the computational cost of CI and showed that it was not much higher than the straightforward information fusion methods (e.g., the arithmetic mean and the weighted mean) usually implemented in most applications. This important outcome has also motivated this study.

While it is crucial to consider criteria dependence for decision making, it is also important to interpret or to give the reasons and drivers (the "how" and the "why") of the aggregated results [7]. In line with this requirement, Montmain et al. [34] used these elucidative capabilities to improve decision acceptability in their decision-making support system for project teams in charge of selecting technical solutions. They tried to justify the evaluation or the ranking of the rival alternatives. Denguir-Rekik et al. [15] applied this functionality in the framework of e-commerce website recommendation when presenting customers with the choice of a suitable e-retailer for their purchases. The deep analysis of e-commerce website performance criteria allowed evaluation of the impact of the uncertainty associated with each criterion in the decision-making process and explained how uncertainty inherent to a criterion can contribute to the final result. More recently, Feyzioğlu and Büyüközkan [19] used this approach in a benchmarking procedure to rate competing systems and to find dimensions that need improvement for new product development. The second part of the proposed framework focused on the elucidation of CI based aggregation.

This research aims to contribute to the risk assessment of software development projects by introducing an integrated methodology that addresses some important issues not investigated until now. More precisely:

- We propose an evaluation and rating method that models interdependencies among software development risk factors.
- We provide interpretation procedures to identify risk factors that mostly affect the final decision.

We also show that considering those points in project risk evaluation is highly significant in detecting the most appropriate alternative in a more realistic setting and in measuring the dimensions of improvement.

The paper is organized as follows: We briefly outline the proposed assessment approach in Section 2, and we present the details of the CI method and the associated elucidation procedure in Section 3. We extract software development risk factors and conduct a case study for a Turkish software company in Section 4. Finally, we provide our concluding remarks and future research directions in Section 5.

2. A framework for software development risk assessment

Risk management has played an important role in software project management over the last 30 years [33]. It is well accepted that the uncertainties faced by software projects should be taken into account when planning and controlling the development of software systems [2,5]. Several studies have identified and/or evaluated software development risk factors. For example, Engel and Barad [17] proposed a set of quantitative probabilistic models for estimating costs and risks of software verification, validation, and testing. Engel and Last [18] extended that research by modeling the software testing risks problem using the fuzzy logic paradigm. The proposed methodology provides management with a decision-support tool to evaluate proposed testing alternatives, thereby focusing resources on the project control.

Büyüközkan and Feyzioglu [8] proposed a three-step fuzzy-logic-based MADM evaluation methodology ranging from the selection of the most appropriate new software options to the selection of the least risky and most value added software project development strategy.

Barros et al. [2] described an approach to develop, retrieve, and reuse management knowledge and experience concerned with software development risks. They showed how risk archetypes and scenario models could represent reusable software development project management knowledge.

Costa et al. [13] presented a technique for evaluating risk levels in software projects through analogies with economic concepts. This technique allowed managers to estimate the probable distribution of earnings and losses incurred by an organization in relation to its software project portfolio. The proposed approach was calibrated by data collected in an empirical study, which provided information about the relative importance of risk factors in software projects.

Lee [28] builts a hierarchical structure model of aggregative risk in software development and rated aggregative risk in a fuzzy environment by fuzzy set theory. The author classified each risk item into two fuzzy sets with triangular membership functions: grades of risk, grades of importance, and rate of risk. Lee then evaluated the rate of each individual risk item and proposed to evaluate the rate of aggregative risk using a two-stage fuzzy assessment method. In succeeding studies, Lee [29] and his associates [30] proposed improved algorithms to find the aggregative risk in software development within a group decision making settings.

Almost all identified software development risk factors are assumed as independent in the above-mentioned and other existing studies. In this study, we introduce a risk evaluation approach based on a two-additive CI to aggregate partial