International Journal of Natural Computing Research
Volume 9 • Issue 1 • January-March 2020

A Pragmatic Assessment of Approaches and Paradigms in Software Risk Management Frameworks

Alankrita Aggarwal, IKG Punjab Technical University, Jallandhar, India
https://orcid.org/0000-0002-0931-1118

Kanwalvir Singh Dhindsa, Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib, India
https://orcid.org/0000-0002-7911-9244

P.K. Suri, Kurukshetra University, Kurukshetra, India

ABSTRACT

Software risk management involves the process of prior recognition and the assessment of vulnerabilities with the classification approach so that the risk avoidance mechanism can be implemented. It includes one of the key factors in software project management with the goal to improve quality as well as the avoidance of vulnerabilities. The term defect refers to the imperfection that may arise because of reasons including programmers' skills, lack of suitable testing strategies, and many others. When there is difference in actual and expected result or meeting the wrong requirement it is called a defect and it forms the basis of risk escalation in the software project, which is obviously not accepted in any type of deployment. To make software reliable, the software should be risk-free from any type of vulnerability factor. Along with reliability, another issue that has arisen is software quality in which the associated factor is with software risk management. The quality of software is to reduce the occurrence of risks and defects with the objective to produce effective valued software.

KEYWORDS

Software Defects and Risks, Software Risk Management, Software Risk Management Mechanisms, Software Risk Management Models

1. INTRODUCTION

Software Risk Management (Boehm et al., 1997) is one of the prominent domains of research in software engineering which includes the prior identification, processing and management of risks and vulnerabilities associated in the software development.

Software Defect Prediction (Fenton, 1999) in software engineering used to predict the deformity in the software module. Numbers of defect are present during the development or after the delivery of software module. To obtain high quality software the prediction process is followed to predict to the defects. The need of obtaining high quality software is to gain customer loyalty (Offutt, 2002).

Few big organizations are using this prediction process as they release their software and software versions frequently and they have less time so instead of manually predicting the defects they use software deformity process (Figure 1).

DOI: 10.4018/IJNCR.2020010102

Tables 1 and 2 present an assortment of attributes of the software risk with the details on the real-world projects with the associated features.

Impact values:

1. Catastrophic
2. Critical
3. Marginal
4. Negligible

Based on the risk profile in the software projects (Figure 2), McFarlan classifies the project into eight types (Table 3).

Figure 1. Software project risks (Wallace et al., 2004)

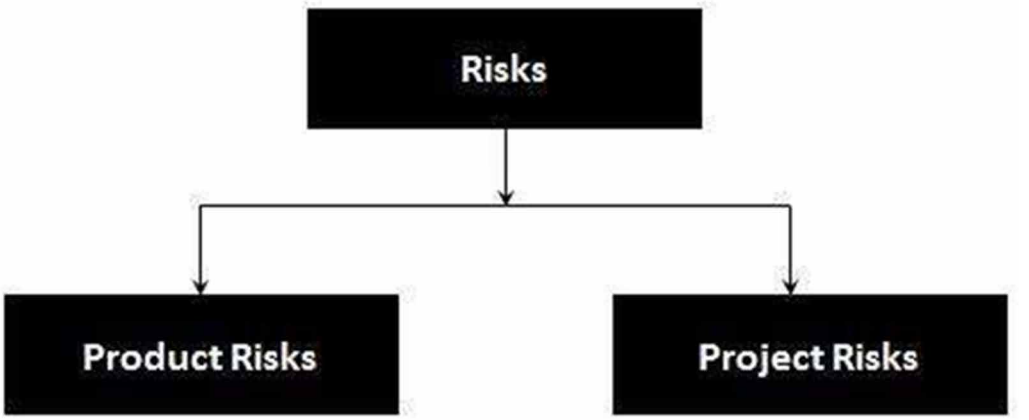

Table 1. Software risk attributes (Keil et al., 1998)

Risk Attribute	Perspective
Product size	Risks associated with the overall size of the software to be built or modified.
Business impact	Risks associated with constraints imposed by management or the marketplace.
Customer characteristics	Risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
Process definition	Risks associated with the degree to which the software process has been defined and is followed by the development organization.
Development environment	Risks associated with the availability and quality of the tools to be used to build the product.
Technology to be built	Risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
Staff size and experience	Risks associated with the overall technical and project experience of the software engineers who will do the work.