
Proposal of risk management metrics for multiple project software development

Miguel Wanderleya,b, Júlio Menezes Jr.a,b*, Cristine Gusmãoa,c, Filipe Limaa,b

aSABER Tecnologias Educacionais e Sociais Research Group, Universidade Federal de Pernambuco, Recife, Brazil
bCentro de Informática, Universidade Federal de Pernambuco, Recife, Brazil
cDepartment of Biomedical Engineering, Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco, Recife, Brazil

Abstract

Multiple project management currently is a reality in software development environments. In tactical and strategic context, the management of only one single project does not exist, and it is necessary the execution of projects simultaneously in order to achieve organizational objectives. In the case of software projects, some characteristics are highlighted, such as, constant changes in levels of scope or product, software complexity and aspects related to human resources, such as technical knowledge and experience, among others. We may consider these characteristics as risk factors that should be managed. In this aspect, a tactical management requires the usage of better-structured information, which leads us to think about the usage of a metrics-based strategy as a support tool for multiple project managers with emphasis on risk factors. In this context, a metric called “Risk Point”. This work presents and discusses the metric “Risk Point”, identifying some points of adjustments. Also, this paper shows an application of the metrics in an environment of multiple projects of software development with the goal of analyzing its applicability and utility as support tool for decision-making and risk monitoring during project life cycle.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: risk management; metrics; multiple project management; software engineering.

* Corresponding author. Tel.: +55 81 2126 8426.
E-mail address: juliovenancio@gmail.com
1. Introduction

A project can be defined as a temporary endeavor undertaken to create a unique product or service. For example, software development projects are run by people, usually having limited resources, being planned, implemented, monitored and documented. Thus, a project environment will be a place that performs one or more projects, whether a company or any organization [1].

Software development projects, given their diverse and abstract nature, offer unique challenges and risks [2]. Projects are critical to the realization of performing organization's business strategy because projects are a means by which the strategy of the company is implemented. In this light projects are risky and managers have to take appropriate actions to prevent them from this dangerous status.

Risk Management depends on the perception and recognition of sources of risk in all phases of a project. The concept of risk is essential in a project. The common definition of risks, either by dictionaries or usages of the term risk, associates several different meanings. It can refer to a possibility of loss, injury, or destruction [3,4]. Consequently, managing risk in multiple project environments is very important to enrich and qualify the information for project manager decision-making.

Risks are events that could adversely affect the development of the projects or organization environment. Risk or Risk factor can damage critical factors such as budget, time or resources. Commonly it affects critical factors such as budget, time and costs. Risk management consists, basically, of identifying, analyzing, planning and controlling events that threat project environment, aiming to avoid or reduce the damage of these events in case of their occurrence [5]. Its management, however, does not guarantee the success of projects, but increase the probability of more effective achievements, respecting deadlines, inside the planned budget and meeting project goals [6].

According to [7] only 39% of software projects are completed on time and on budget. A significant part of the failures can be related to poor risk management. Risk management is one of the fields related to project management and practice becomes essential to measure the size and complexity of software grows [8].

Nevertheless, risk management is still little used in part for their application in software development environment is recent, another to depend on the manager's experience in analyzing qualitative factors which causes certain subjectivity.

The use of metrics can make management more efficient and may accurate risk, contributing to the risk analysis, risk response planning, risk monitoring, risk control, and reduce the degree of subjectivity that would facilitate the understanding of the risks and diffuse their application in the software development environment.

From this perspective this paper aims to present a proposal of Risk Management Metric and to do so after this introduction, Section 2 presents Risk Management background and concepts of Metric. Section 3 brings an analytical view of related works. Section 4 presents and discusses the process of Metric proposal. Some alternative Metrics are presented in Section 5. Section 6 discusses the Metric evaluation and outcomes. Finally in Section 7 we present some considerations about the lessons learned and future work.

2. Background

The increasing competition in the market and the expectations of the client’s requirements force the software developing organizations to manage closely their projects’ risks. The real challenge in the present software development environment is to deal with risk projects and improve the success-to-failure ratio of multiple software projects environments. In this section we present basic concepts of Risk Management and Software Metrics.

2.1 Risk management

Risk Management depends on the perception and recognition of sources of risk in all phases of a project. The concept of risk is essential in a project. From an overall business perspective, the success of many organizations is becoming increasingly dependent on the success or failure of the software they build. In this environment, managing risk is not only a sound development practice, but also a vital business practice [9, 10].
When risk management methods are used, they are often simplistic and users have little trust in the results of their risk analysis results. Given the increasing interest in risk management in the software industry, for applying risk management more widely, it is necessary to provide comprehensive support for risk management, guidelines for application, support communications between the stakeholders and be credible.

In software engineering literature there seems to be a consensus of the activities that compose risk management process [2, 9, 11]. It is important to notice that all activities are based and centered on communication, and is used by project teams to identify and handle the risk on their project. This is a cyclical and continuous process composed by:

**Risk Management Plan.** The goal of this activity is deciding how to plan the project’s risk management activities, resources allocations, teamwork, and documentation standards.

**Risk Identification.** Determining the risks that might affect the project and documenting their characteristics. In this activity many techniques to collect risk are assigned in the literature.

**Risk Analysis.** Analyzing all project activities conditions (qualitatively) to determine and prioritize their impact on project objectives. As it as well, determining the probability (quantitatively) and risk consequences estimating their impact on projects objectives.

**Risk Response Plan.** Risk is always involved with loss, but also considers the possibility that the outcome of certain risks might be a gain. In this light, this activity determines how to enhance opportunities and minimize lost.

**Risk Monitoring.** This is a very important activity, as well as checks all risks identified and looks forward to new risks in the environment.

**Risk Control.** Executing and evaluating the effectiveness of risks responses plans. It is essential to notice that well-defined schedules are very important to the success of this activity.

**Risk Communicate.** The communication between software project teams and stakeholders is one of the most important factors for the successful accomplishment of risk management. Risks, problems and crises can appear, when the communication structure is weak in the organization environment [11].

### 2.2 Software metrics

Software metrics can be defined as the continuous application of measurement-based techniques to the software development process and its product to supply meaningful and timely management information, seeking for software and process improvement [12].

There are two types of metrics: base and derived. Base metrics is defined in terms of a single one attribute and the method for quantifying it. It is functionally independent of other metrics. The measurement method comprises the logical sequence of operators in order to quantify an attribute with respect to a specified scale. There are two types of measurement methods: subjective and objective. The first one quantifies an attribute through human judgment, whereas the objective method is only based on numerical rules.

Derived metrics is defined as function of two or more values of base metrics. The function to characterize a derived metric must be a mathematical function between two or more base metrics (measurement function). Derived metrics captures information about more than one attribute or the same attribute from multiple entities.

There are several reasons to measure in software engineering, either a tactical or a strategic fashion. Also, it is worth highlighting that measurement, if it is well done, gives information that generates knowledge for decision-making, as well as provides better control and tracking of the undertaken activities. Therefore, the measurement process cooperates to continuous software process improvement. There are four reasons to perform measurement in software engineering: i) to characterize for better understanding of processes, products, resources and environment, establishing reference models for further evaluations; ii) to evaluate the status according to what was planned; iii) to predict, in other words, get enough basis to estimate, minimizing risks, and; iv) to improve the product quality and the performance of process and project through concrete quantitative data, which allows to visualize improvement points more clearly [13].
3. Related works

In software engineering there are few references about the usage of metrics for project risk management. Barry Boehm [14] is considered a pioneer in the application of risk management in software engineering. He proposed a software risk management framework focused on risk analysis. The activity of risk analysis in his work is defined as Risk Exposure calculation, which is defined as the multiplication between Probability of Risk versus Loss or Impact of Risk. This analysis is only used for risk prioritization.

The work [15] proposes a way of to measure the risk level of a project through a metrics called Risk Point. According to the author, the objective of Risk point metrics is to define how risky is a software project based on number of identified risks and project complexity factors. We use this metrics as one of the indicators for this dissertation. However the author did not evaluate Risk Point in practice.

Another related work defines a quantitative approach where risk concepts of economics, specifically credit risk, are used to propose a method of risk assessment in software projects [16]. In this work, the author proposes a way to calculate how much capital a software development organization can gain or lose due to the risks of a selected set of projects. The adopted method allows the selection of projects’ sets that seeks to maximize the cost-benefit for an organization. The risk assessment method uses project characterization (size, duration cost and return) and a questionnaire to identify risks. However, this method was not evaluated in practice.

The use of the Goal-Question Metric paradigm to define software process metrics with the goal of monitoring risk factors is discussed on [17]. On the other hand, the proposal was not put in practice.

Some works used metrics for technical risks using Risk-Based Testing concept (RBT) [18, 19]. The objective of the metrics is to indicate information regarding test cases control through risk analysis and monitoring of system requirements. However, these metrics are not proposed as a tool for management of projects, providing only product risk view based on system requirements, architecture and coding analysis.

Another related work discusses the need of the usage of metrics for risk management, and shows examples of how they can be used [20]. For example, a risk factor related to team qualification – experience and knowledge level on certain technology. Hence, it is a data that could be quantified and followed through project life cycle. On the other hand, this paper does not present any practical application or assessment.

This paper approaches the evolution of the proposal presented by [15] because it shows a proposal of a metrics – Risk point, whose goal is to measure risks in the context of multiple project software management as support tool for project managers. Therefore, the rest of this paper presents Risk Point metrics in details as well as proposes improvements and previous assessment in a real environment.

4. Risk Point Metric

The Risk Point (RP) metric was developed aiming a simple objective, in just one value, represent the overall risk exposure level of a project. Basically, the metric is defined in terms of the amount of identified risks, where these risks are defined in terms of its probability and estimated impact, as the concept of Risk Exposure (RE) [21], that is defined as:

\[ RE(Risk) = \text{Probability} (Risk) \times \text{Impact} (Risk) \]  

As a starting point for defining the Risk Point metric, it was used the same idea that was applied in the Use Case Points metric (UCP). That technique proposes a way to quantify the size of the whole project based on the defined Use Cases and other technical and environmental factors [22, 23]. Summarizing, by using the UCP, it is possible to define a single value to represent the project, based on the Use Cases and others technical and environmental factors of the project. The weights used in Risk Point were defined based on interviews with students of a software project management course and others software management professionals [15].

Similarly, the Risk Point metric allows quantifying the project in terms of its identified risks. It is necessary to estimate the Risk Exposure value, i.e. Probability versus Impact, for each identified risk, so, for a specific data collection about the current risks of a project, it is possible to determine a value of Risk Point (RP), as follows:

\[ RP = PCF \times URPW \]
Where, PCF is the Project Characteristics Factor and URPW means Unadjusted Risk Point Weight. PCF is a value for giving the project a weight and adjust the metric final value based on technical and environmental factors [24]. This value is defined through the answers of a questionnaire, which was developed from an empirical study with software project managers and management students, as mentioned [24]. Then, PCF is defined as:

\[
PCF = 1.05 + (0.015 \times CF)
\]

(3)

CF means Characteristic Factor, it’s determined by answering the 8 questions of the questionnaire with scores between 0 and 4, and then this answer is multiplied by the defined weighted value for each question. Finally, these 8 products are summed, resulting in the CF value:

\[
CF = \sum_{i=1}^{8} (Question_{i} \times Weight_{i})
\]

(4)

Once the CF value is defined in the interval [0,39.48], it means that the PCF occurs in the interval [1.05, 1.6422].

URPW is the Unadjusted Risk Point Weight, composed by the identified risks during the data collection, in terms of their Risk Exposure as defined in Equation (1). In this study, the estimation adopted was values in \( \{0.1, 0.2, \ldots, 0.9\} \).

The Unadjusted Risk Point Weight (URPW) value is formed by the summation of the Weights of each identified risk, being this Weight defined according the Risk Exposure value, as can be seen in the following table.

<table>
<thead>
<tr>
<th>Classification</th>
<th>RE(Risk)</th>
<th>Weight(Risk)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>[0.0, 0.2)</td>
<td>1</td>
</tr>
<tr>
<td>Low</td>
<td>[0.2, 0.4)</td>
<td>2</td>
</tr>
<tr>
<td>Average</td>
<td>[0.4, 0.6)</td>
<td>3</td>
</tr>
<tr>
<td>High</td>
<td>[0.6, 0.8)</td>
<td>4</td>
</tr>
<tr>
<td>Very High</td>
<td>[0.8, 1.0]</td>
<td>5</td>
</tr>
</tbody>
</table>

Thus, for \( n \) identified risks, the URPW value follows this rule:

\[
URPW = \sum_{i=1}^{n} Weight(Risk_{i}) \mid n = \text{number of identified risks}
\]

(5)

Briefly, a given data collection (even in a subjective way, with values in a 5 levels scale for Probability and Impact) about the current risks of a project yields a value which represents the overall evaluation concerning the known risks of a project in some specific moment in its life cycle. This value allows a broad risk assessment about the risk exposition level of a project in different moments, and also allows a way to compare between different projects based on their identified risks.

5. Alternative metrics development

Just changing the weights for the Risk Exposures classification, showed in Equation (5), two new alternative metrics were defined. Note that by changing the weights values we can create many other metrics, but the ones presented here focus on the concept, taken as the most important, inside these changes.

5.1 Pure Risk Point (PRP)
In this alternative metric, all the weights are defined as 1. Therefore, the URPW value composition becomes a simple summation of all identified risks, without distinguishing the different Risk Exposure values of each risk. Thus, the PRP metrics prioritize the assessment of the amount of different risks identified during some data collection.

5.2 Exponential Risk Point (ERP)

This metric presents the weights in a base 2 exponential growth, i.e. \{1,2,4,8,16\}. Therefore, ERP is even higher for the highest occurrences of Risk Exposure levels. The URPW receives higher values for “Average” or upper levels of Risk Exposure, making then this metric more sensible to elevated levels of risk.

For a better understanding of the changes in Risk Point metrics, the complete weight table for the different risk classifications, and a posterior metric composition, are presented at Table 2 below.

Table 2. Weights of PR, PRP and ERP.

<table>
<thead>
<tr>
<th>Classification</th>
<th>RE(Risk)</th>
<th>Weight (PR)</th>
<th>Weight (PRP)</th>
<th>Weight (ERP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>[0.0, 0.2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Low</td>
<td>[0.2, 0.4)</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Average</td>
<td>[0.4, 0.6)</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>High</td>
<td>[0.6, 0.8)</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Very High</td>
<td>[0.8, 1.0]</td>
<td>5</td>
<td>1</td>
<td>16</td>
</tr>
</tbody>
</table>

Considering the Risk Exposure calculation presented in this paper, we realize that the risk’s classes has the probability distribution described in Table 3, since the Risk Exposure is defined as Probability versus Impact and both are defined in the interval \([0.1, 0.9]\).

Table 3. Probability of occurrences of each classification.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>0.5050</td>
</tr>
<tr>
<td>Low</td>
<td>0.2424</td>
</tr>
<tr>
<td>Average</td>
<td>0.1515</td>
</tr>
<tr>
<td>High</td>
<td>0.0707</td>
</tr>
<tr>
<td>Very High</td>
<td>0.0303</td>
</tr>
</tbody>
</table>

From this distribution we can note that the most critical risks, classified as "High" or "Very High" represent only 10% of the possible RE values, and the class of risk "Very Low" represents 50% of the possible values of RE.

6. Metrics application and preliminary assessment

The RP, PRP, and ERP metrics were applied in two software development projects from the same environment, of a research group specialized on the development of educational software, located in Brazil. In each project, data collections were made in weekly meetings with the leaders of each project. These projects are related to the development of distance education systems for training of health professionals. Part of the results presented in this section illustrates how the presented metrics enabled the risk assessment.

The first studied project, called "Web Design Project", which focus on development of the front-end of a health professional training system. This project had a team with 6 members, distributed in different buildings, and with low experience in the used technology.

The second assessed project was the "Web Service Project" with the objective of developing the back-end and an API to be used in training systems, with a smaller team, 5 members, allocated in the same room, and with low experience on the used technology.
To collect data, in both projects we used an agile risk management process called GARA [25], consistent with agile development methodologies, such as Scrum, focused to multiple projects and simple enough for the risk management activities, such as the data collecting, do not harm the project development.

To illustrate the way and better understand how the metric is calculated, one of the data collection executed in the project called "Web Design Project" adopted the following set of steps. Initially, in the meeting for data collection, the questionnaire is applied, where the eight questions determine the value of the PCF, based on the answers given, the weights of each question and the formula for generation of PCF value previously presented. In this data collection, the obtained value was PCF = 1.27695.

Still in the same data collection of the project "Web Design Project", a brainstorming session was performed to the project leaders, aiming to identify the maximum possible risks, in order to set the URPW value (and then compose the final metric value). For every risk identified it is required to estimate a value of probability and impact (a value between \{0.1, 0.2, ..., 0.9\}). After this, 13 different risks were identified about the project, then were evaluated the products probability X impact, i.e. the Risk Exposure (RE). According to Figure 1, 7 risks were classified as “Very Low”, 4 as “Low” and 2 as “Average”. Therefore, from the defined summation of the weights, the URPW value (for each metric) was calculated as below:

\[
URPW_{RP} = 21 = 7 \times 1 + 4 \times 2 + 2 \times 3, \\
URPW_{ERP} = 23 = 7 \times 1 + 4 \times 2 + 2 \times 4, \text{ and} \\
URPW_{PRP} = 13 = 7 \times 1 + 4 \times 1 + 2 \times 1.
\]

Making the product of each URPW value by the PCF value, according to the given metric's definition, the following final values are obtained: \(RP = 26.81595\), \(ERP = 29.36985\) and \(PRP = 16.60035\).

The same procedure is performed for each collection of each project, thus defining the respective values for each of the metrics presented. The collections were performed in weekly iterations and Figure 1 shows part of the obtained values.

![Projects Evaluation](image)

Fig. 1. Application of the metrics in real environment.

Some important points about the values above presented can be observed. First, the fact that for higher Risk Exposure values the project presents higher values for RP and ERP metrics. As mentioned, the PRP metric only reflects the amount of risks identified during the data collection. At iterations 1 and 2, RP and ERP because the RE values were all included among the classes "Very Low" and "Low", and these levels have the same weights in the metrics (1 and 2, respectively).

At the iteration 3, it is visible the difference between the values, because these collections contain the risk classification "Average" that have different weights in metric RP and ERP, which are 3 and 4, respectively. The
difference of weights for risks with higher RE value, from the classification of "Average", influences the final metric value, as can be seen in the iteration 3 in both projects.

With the final values of the metrics, it is possible to assess the growth trends or decrease in the general level of risk exposure of each project, make comparisons between different projects based on identified risks and evaluate the efficiency of actions and resources applied at specific moments of the project life cycle in order to reduce risks.

7. Final considerations and ongoing work

This paper proposed metrics for software risk management: Pure Risk Point (PRP) and Exponential Risk Point (ERP), both based on Risk Point. PRP is more efficient to assess the amount of risk present in the project. ERP accentuates the occurrences of risks according to their classification, high and very high risks have higher weights. The idea was to provide alternatives in order to improve metrics sensitivity. Using these metrics it is possible to quantify the risk of a project providing tools for decision support in multiple projects environment that can be used to make comparisons between projects and risk monitoring during project life-cycle.

For now we have some ongoing work: by proposing other metric variations that prioritize certain design characteristics according to the purpose of study. For example, consider the value of risk contributions by the amount (Normalized Risk Point) and consider the risk contributions of values without dividing them into categories (Continuous Risk Point). Therefore, we are planning to conduct a case study in order to assess the metrics and its applicability in multiple software project environments.

Acknowledgements

The authors would like to thank Brazilian Council of Science and Technology – CNPq, Brazilian Ministry of Health and SABER Tecnologias Educacionais e Sociais research group for the whole support to develop this work.

References