Defining Indicators for Risk Assessment in Software Development Projects

Júlio Menezes Jr.

Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil, 50740-560 jvmj@cin.ufpe.br

Cristine Gusmão

Departamento de Engenharia Biomédica, Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco, Recife, Brazil, 50740-530 cristine.gusmao@nutes.ufpe.br

and

Hermano Moura

Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil, 50740-560 hermano@cin.ufpe.br

Abstract

The usage of indicators acts in both strategic and tactical levels, it is effective for optimization of processes and also supports managerial decisions. Despite the relevance of risk management in software projects, it is in fact still usually overlooked by the organizations that develop software. One reason for this fact is that the concept of risk is abstract and subjective, and its management does not bring apparent immediate practical result. Differently, for example, in the financial market, where the risk management is consolidated and widely applied, and it is currently a need for the companies that work in this field. Briefly, risk management practices just became effectively consolidated in the financial market field when the uncertainties became measurable. In this context, this paper aims to define and propose indicators that are specific for environments of software projects in order to support risk assessment activities — risk identification and risk analysis. To achieve this objective, we first developed a systematic mapping study in order to collect evidences about metrics, indicators and relevant information for risk assessment. These findings were combined with the identification of measurable risk factors, providing, thus, a set of categorized indicators for software development environments.

Keywords: Indicator, Risk Management, Software Project Management.

1 Introduction

Project management is currently a consolidated area in the context of Software Engineering. Its practices, associated with an inherently changeable nature of the software, have promoted the need of new methodologies of project management which are adherent to software development environments. It can be argued that these methodologies have been proposed with a unique goal: to ensure the projects successes.

One of the ways to highlight the view of these factors is through the usage of risk management practices. Additionally, the importance of the usage of processes, techniques and tools of risk management is even more recognized in software development environments [1]. Some authors consider that managing projects is managing risks [2]. This fact is due, in part, by the understanding that a significant portion of projects failures could be related to a poor risk management. Risk management is one of disciplines related to project management, and its usage becomes increasingly necessary as the size and complexity of software grow [3].

This complexity is better perceived when we consider that one Information Technology organization rarely executes one project at time. Different projects are executed simultaneously, and once they are in the same environment, they are sharing resources. Therefore, there is also a concern not only to each project restriction – scope, cost, time and quality –, but also to the relationships between projects. In this context, project management activities aims to optimize the usage of resources within the environment.

The need of software products by organizations in general is perceptible, and it is in many cases essential for their improvement or even their survival. Likewise, the software development process must be aligned to the business models of these organizations in order to generate products that meet their needs, commonly inside constraints of cost, time, scope and quality. Faced with these challenges, companies that produce software have to manage effectively their projects, under penalty of losing their place in the market because of lack of effective and proactive practices of management. Then, it is necessary to have an organizational structure that allows, for example, the overview of riskier projects, and their control based on information from risks or that indicates risks, providing a better risk-driven management.

Despite the relevance of risk management in software projects, given the diversity of existing approaches, models and processes that deal with risk in one project, multiple projects environments or distributed projects [1, 4, 5, 6, 7, 8, 9], the risk management is still usually overlooked by the organizations that develop software. One reason for this fact is that the concept of risk is abstract and subjective, and its management does not bring apparent immediate practical result [10].

Another problem that inhibits risk management is that it is a relatively recent area in software project management, and its practical application tends to be biased [8], because there is still a high degree of subjectivity especially during the activity of risk assessment. Differently, for example, in the financial market, in which risk management is consolidated and widely applied, it is currently a need for the companies that work in this field. Briefly, risk management practices just became effectively consolidated in the financial market when the uncertainties became measurable. In this light, we can say that there is a lack of indicators and metrics that support risk management in the context and viewpoint of software project management, even when we consider organizational factors.

In this context, this paper aims to present a set of indicators that support the risk assessment in environments of multiple software development projects, and the methodology to define and evaluate these indicators. After this introductory section, this paper is organized as follows: Section 2 presents main concepts about risk management in multiple project environments; Sections 3, 4 and 5 presents the application of the defined metodhology, introduced at Section 1.1 below. Finally, Section 6 draws the main conclusions and future work.

1.1 Methodology

To better understand the process of risk indicators definition used in this work, this sub-section describes the necessary steps used to conduct the following activities: i) Systematic Mapping Study, ii) Indicators Definition and iii) Indicators Evaluation.

The systematic mapping study was performed with basis on two technical reports [15, 16], including the process of search strings definition and used sources. The systematic mapping study had two goals: i) to find related work that deal with indicators and metrics in risk management and ii) based on the work found, to identify the type of information that each indicator or metric uses to measure risks. For the second item, we also tried to extract data and information used to the measurement of the identified indicators and metrics. For example, the indicator A is composed by the sum of B and C, therefore, B and C are information used to measure risks. At the end of this step, for better understanding and definition of the proposal, the found information was categorized.

For risk indicator definition, we used the result of the information categorization found in the mapping study, the ad hoc literature review and the SEI Risk Taxonomy [5] as the basis to organize the indicators. Finally, the risk indicators were qualitatively assessed through questionnaires submitted to specialists in the domain of software project management and risk management, with the goal of identifying improvement points, and also to evaluate

the research methodology adopted to define the risk indicators. The detailed application of this methodology can be found at Sections 3, 4 and 5 of this paper.

2 Risk Management in Multiple Projects Environments

In the context of project management, it can be argued that all projects involve risks. The decision to pre-plan a project is in itself something risky, and therefore it is necessary an analysis of several internal and external factors to the environment in order to define the project feasibility. The controlling and monitoring of risks during the project life cycle is a way of perform a better visualization the main critical success factors, enabling more efficient decision-making. The area of risk management in projects has been increasingly studied in the last years by academic and industry practitioners [11] because of its inherent presence in projects environments.

Therefore, the risk management permeates all the phases of project management, analyzing especially issues related to scope, cost, time and quality. In other words, there are several factors that, if not property managed, can contribute to project failure.

In an organizational environment, it is important to notice that there are multiple projects running simultaneously. In this light, it is necessary to analyze not only the critical success factors in an isolated way. Projects are sharing resources and people, and the reason for existing projects is linked to strategic goals of organizations. Inside this context, there is a need to manage multiple projects simultaneously, considering restrictions in each project and between projects, so that they could be optimized and successful.

There is a consensus about the risk management activities in general. These are:

- i) **Risk Management Planning:** The goal of this activity is to define the risk management strategy, the necessary resources and schedule to perform the process of risk management. According to PMBOK Guide [12], this activity generates an artifact called Risk Management Plan, which describe policies, cost, responsibilities, schedule and all the necessary activities to execute the Risk Management.
- ii) **Risk Assessment:** This activity comprises two processes: Risk Identification and Risk Analysis. The first one aims to raise all the existing possibilities and events in a project or organization, whereas the second one characterizes and prioritizes the identified risks in order to help in making decisions. To identify risks the following techniques can be used: brainstorming, work breakdown analysis, risk breakdown structure, checklists, among others. Risk analysis characterizes the most important aspects of risks, aiming to explore the best mitigation strategies. It deepens the view of risk identification. In this process the risks are usually categorized and prioritized according to some specific criteria in order to focus the risk management on the main risk factors. It also determines risk drivers and performs estimations, commonly through analysis of probability likelihood and impact consequences of an event.
- iii) **Risk Planning:** This activity aims to handle the identified risks according to their analysis. We assume that an organization or a single project cannot deal with all identified risks at same time, therefore Risk Planning activity selects the risks to be managed. In addition, for these risks, contingency plans and action plans (to minimize or avoid) are elaborated. These plans are assessed through a cost-benefit analysis, in order to verify the cost of action versus cost of risk occurrence. So, if the cost of action is bigger than the cost of risk occurrence, it does not worth to apply the action, and therefore another alternative must be taken into consideration.
- iv) **Risk Controlling:** Risk Controlling is responsible to monitor and track risks and to check if risk management activities have been effective. With this activity, it is possible to update mitigation strategies as well as to reorganize risk priority, once during the project life cycle new risks can arise and another risks may have their priorities reduced.
- v) **Risk Communication:** Risk communication can be considered as a point of integration between the other activities presented before. This activity is a critical component for successful risk management [13], because risk communication allows information exchange between project stakeholders, contributing to collective effort and integration of activities in the process of risk management.

2.1 Multiple Project Risk Management

Companies usually manage risks on each project separately, and they do not observe the influence of risk in organizational level between projects. Most of the works in the literature does not consider that risks could be shared between different projects in the same environment.

However, if we assume that risks are present in all levels of management, and there is a strong relationship between risks in each level, that would be relevant also to see risks in a holistic view. For better understanding, consider the following assumptions about project risk management:

- One single project risk management: we only see risks that affect only one project, most of them related to cost, scope, time and quality. Therefore, we have one project and the respective list of risks to be managed. The goal is just to assure the project success;
- Multiple projects risk management: in this case, we see the same aspects approached on one single
 project risk management plus the shared risks in the environment. Hence, we have a matrix of project
 versus risks. The goal is to assure projects successes considering their optimization, allocating
 resources and analyze the influence of these resources on a set of projects.

In software engineering, we can highlight two approaches of risk management considering multiple project risk management. One of them is the mPRIME Process, a process model for Risk Management in environments of multiple software projects [1]. mPRIME Process is presented through a life-cycle model for environments of multiple projects, comprising a set of roles, artifacts and workflows to support risk management. The process model was built using concepts from literature review and main technical reports and standards from areas of project management and software quality.

Another approach is called GARA – Gestão Ágil de Riscos de Ambiente (in English, Agile Management of Environmental Risks), which is an agile process for risk treatment in environments of multiple software projects [14]. As mPRIME Process, GARA is also presented through a life-cycle model for environments of multiple projects, comprising a set of roles, artifacts and workflows to support risk management. In order to be adherent to agile environments, GARA names risk as impediment.

Activities of Risk Assessment are clearly presented on both processes. mPRIME Process supports the process of risk identification, identifying the risk factors from the environment. The main artifact generated during this activity is a list of risks per project, followed by the relationship between these risks. For risk identification, the process suggests the usage of brainstorming, checklists and comparison by analogy through information from past projects. The risk analysis process approached by mPRIME is qualitative, using scales of probability and impact in order to rank risks. Risk analysis also organizes risks using a predefined categorization.

In GARA, the risk identification is performed using brainstorming technique, generating a list of identified risks or impediments. Risk analysis, in turn, produces a ranking of lists through qualitative analysis, defining probability and impact of each impediment.

However, it is important to notice that none of both approaches mention the usage of indicators to support risk assessment. Only mPRIME Process suggests a list of indicators or metrics for risks, but they focus only on monitoring and tracking. Therefore, the existence of this gap one more time highlights the need of the definition of indicators for risk assessment considering the context of one single project and multiple projects.

3 Systematic Mapping Study

According to Kitchenham and Charters [15], the usage of systematic mapping studies is appropriate when we want to have an overview of a particular subject or theme. Its objective is to summarize the data that answers the research question, and it commonly does not include statistical analysis or meta-analysis. Systematic mapping study is recommended when very little evidence is likely to exist or that the topic is very broad, and it usually is the earlier step for future systematic reviews because the formulated research questions tend to be broader.

It is important to notice that the procedure adopted for systematic mapping study in the phases of planning and execution is the same for systematic reviews. The only difference is the fact that a systematic review presents a deeper quantitative data analysis after the selection of publications. Thus, for conducting this mapping study, the technical report of Biolchini et al. [16] was adopted as the main guide for this systematic mapping study, supported by the technical report of Kitchenham and Charters. [15].

One of the initial activities to execute the systematic mapping studies is the protocol definition, which specifies the research questions and the necessary steps to the execution of the study. The next subsections presents the research protocol adopted for this work.

3.1 Protocol

To define this protocol, the research question focus to be formulated must be clear. The research question is crucial to define the appropriate protocol to the mapping study goals. For this work, the focus of the research question is

related to the identification of indicators and metrics that support risk management in the context of software projects, and to find the set of information and data necessary to assist the risks measurement in software project.

It is noteworthy that, despite the focus on software projects, this study also includes organizational and tactical management levels. Thus, the view of management is broader, since we must take into consideration the financial and management aspects and analysis of its application in software development environments. Therefore, we also consider works of the following areas of knowledge: i) Management, because it deals with management at various levels, and ii) Economics, because it is an area in which risk assessment process is performed quantitatively.

3.1.1 Research Question Formulation

This subsection presents i) the objects to be investigated (Research Questions), ii) the subject and area where the object will be studied (Population), iii) at which point of the population the object will be studied (Intervention), and iv) what are the results to be considered (Outcomes). These are the research questions:

Primary question – What are the evidences of works that approach metrics and indicators for risk assessment?

Secondary question – Based on the evidences found, which information is used to measure risks?

In this research, evidence means any papers published in conferences and journals that bring proposals of metrics, indicators and approaches of risk measurement in general. Our main focus was to find not only metrics and indicators, but mainly the type of information used to define these measures. Therefore, for each evidence we extracted the type of information that composes each proposal of metric and indicator. We decided to provide a more qualitative analysis of the results because we aimed to use them to propose indicators for risk assessment in environments of software projects with focus on the identification of the type of information and associated risk factors. Based on the research question, we defined the following structure, important information to specify the search string:

Population – Works related to management and assessment of projects.

Intervention – Approaches, techniques and methods for risk assessment.

Outcomes – Approaches of risk measurement, indicators and metrics that support risk man-agement and risk assessment. To answer the secondary question, we added the type of data and information used to measure risks.

3.1.2 Inclusion Criteria

To select papers that address the research questions, criteria must be defined. These are:

- Discuss about metrics and indicators for risk assessment;
- Only full papers. Abstracts only will not be accepted;
- Search strings must appear in the title, abstract or keywords;
- Free or accessible through CAPES portal¹;
- Published from 2004 to 2010;
- Written in English;
- It must be part of at least one of the following major areas of knowledge: i) Computer Science, ii) Management and iii) Economics.

The search process and peper selection was performed from August 2010 to January 2011. Therefore we just considered works from 2004 to 2010, a smaller period, to show how the project management community dealt with metrics and indicators for risk management. And as project management is multidisciplinary, we considered works from other areas such as management and economics too.

3.1.3 Approach to Select Sources

To source selection, we used as a criterion to choose of the most relevant and used sources in the area of software engineering and project management that allow a reliable search using web search engines. Three sources were selected: i) Scopus (http://www.scopus.com), ii) IEEExplore (http://ieeexplore.ieee.org) and iii) Engineering Village (http://www.engineeringvillage2.org).

¹CAPES portal (http://www.periodicos.capes.gov.br/) allows the access of main indexed journals and conference papers in main Brazilian universities.

3.1.4 Search Strategy

Table 1 summarizes the adopted search strategy. As shown, to perform searches in the selected sources, we defined a set of keywords to the population, intervention and outcomes. The Boolean operators and the combination between strings are accepted by these sources.

Table 1. Search Strategy

	Strings		
Population	"Software project" OR "Project management" OR "Software development project" OR "Project		
	assessment" OR "Project appraisal" OR "Project appraisement" OR "Project estimate" OR		
	"Project estimation" OR "Project evaluation" OR "Project valuation"		
Intervention	"Risk management" OR "Organizational risk" OR "Operational risk" OR "Risk assessment" OR		
	"Software risk management" OR "Project risk" OR "Risk appraisement" OR "Risk estimation"		
	OR "Risk evaluation" OR "Risk valuation" OR "Risk analysis"		
Outcomes	"Risk measurement" OR "Metric" OR "Metrics" OR "Indicator" OR "Measure" OR		
	"Mensuration"		
	Search Strategy: Population AND Intervention AND Outcomes		

The definition of the research protocol was defined by all the authors, whose have theoretical and practical experience in software project management and risk management.

3.2 Mapping Results

This section shows the results found, including the analysis of the selected papers. We found 506 articles using the search strategy presented at Table 1. The Table 2 shows the number of references retrieved from each source using the defined search strategy on papers published in conferences and/or journals from 2004 to 2010.

Table 2. Number of work found by source

Source	Number of work found
Scopus	339
IEEExplore	49
Engineering Village	118

Figure 1 summarizes the work done so far through the flow used to selection of papers.

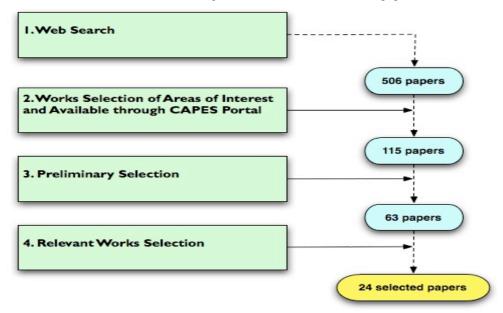


Figure 1. Search summarization

Four steps were performed to the study selection, where initially we found 506 papers and at the end, with the application of the inclusion criteria, just 24 were selected. All the selected papers fit to all inclusion criteria presented at Section 3.1.2. To better organize the activities, we followed four steps. In step 1, we just applied the search strategy using the strings and Boolean operators shown in Table 1. The selected sources enable the search of strings on the abstract, title, indexed keywords and the year of publication. In addition, in Scopus is possible to choose works just from the major areas of Physical and Social Science and Humanities. Considering also duplicated works, 506 papers were returned using the web search engine.

The selection of works from the areas of interest in step 2 involved the elimination of works that do not approach risk assessment in projects in the context of economics, management and computer science and the elimination of duplicates. In this step we just considered works that were free or available through CAPES portal. From 506 papers, 115 were selected.

The step 3 consisted of careful reading of the title, abstract and keywords in order to identify potential works that answer the research questions. We analyzed 115 papers, and 63 were pre-selected for full reading. And finally, in step 4 the full text of the remaining papers was read. After the analysis of 63 articles, only 24 ones brought proposals of measures, metrics or indicators for risk assessment in projects.

The search process in Scopus started on August 17th 2010 and finished on November 2nd 2010, while in IEEExplore and Engineering Village started on November 19th 2010 and finished on 5th January 2011.

3.2.1 Result Analysis

This section presents the result analysis, showing the analysis of selected evidences. To get quantitative data, we performed a data extraction, whereas the following information about the selected papers was collected:

- Type of environment (see Table 3);
- Whether the paper approaches project risk or operational risk (see Table 5);
- Whether the paper is a conference or a journal paper;
- Information used by the proposed metrics, measures and indicators (see Table 6).

In the classification shown at Table 3, we tried to consider what kind of environment each work explicitly consider. Five types were identified.

Table 3. Types of environments found

Type of Environment	Meaning
Information Technology (IT)	The paper describes metrics or indicators for software development
	environments and/or any kind of organization that involves information
	technology services
Construction	The paper is related to construction environments
Industrial	The paper approaches risk measurement inside industries or manufacturing
Research and Development (R & D)	The paper describes environments for research and development projects
Generic	We consider as generic environment if the paper does not mention for
	where the proposed metrics or indicators are recommended

Table 4 shows the number of selected papers per environment. We realized that most of the works found is actually related to organizations/environments that explicitly approach information technology.

Table 4. Type of environment versus number of papers

Environment	Number
IT	15
Construction	1
Industrial	1
R & D	1
Generic	6

It is important to notice that we defined the search strings for any kind of project environment, and we realize that risk management still is very well accepted at information technology environments in general.

Table 5 presents the quantity of works per type of risk. By Project risk we say that it threats negatively the success of a project. And by Operational risk we just looked at ones that consider operational aspects of an enterprise, as a whole, especially continuous activities.

Table 5. Type of risk versus number of papers

Type of risk	Number
Project	22
Operational	2

The indicated result was expected due the fact that the research is focused on project management. Only the references [17, 18] consider operational risks, approaching aspects of organizational changes.

Regarding the type of publication, the number of journals papers was equal to the number of conference papers. This classification is important because works published in journals commonly tend to be more cited because they are commonly more complex, showing not only proposals but also empirical studies.

To get the information and data used by the proposed metrics and indicators to measure risks, we initially listed all information explicitly mentioned in all the selected papers. Hence, we identified 81 different data and from them we created 11 categories, presented at Table 6.

Table 6. Categories of information

Category of	Meaning
Information	
Cost	The metrics or indicator use information about cost to measure the risks. Commonly, they provide cost monitoring using a risk-driven approach or adapt measurement models from financial risks.
Time	The metrics or indicators use information about time or schedule to measure the risks. The most of cases aim to make a better control of schedule or time estimation.
Quality	The metrics or indicators uses information about product quality to measure the risks, commonly included at testing activities.
Team	The information used is about the teams characteristics, levels of knowledge, experience, expertise, motivation, turnover, effort rate and levels of communication.
Code	It approaches information about lines of code or another metrics that use the source code as a source of information.
Risk	It uses information directly from each identified risk such as risk exposure (probability of occurrence and impact), number of risks identified and risk dependence.
Size	It approaches information about size of project or size of product, including number of requirements.
Changes	This category just encompasses changes in scope, requirements, implementation or design phases.
Complexity	It is about measures of complexity level of product.
Design	This category approaches metrics or indicators related to product designing process, with focus on the architectural design (for example, reuse rate).
Organizational	It approaches measures directly related to the process or organizational characteristics, such as maturity level (inversely proportional to levels of risk) or any kind of measure of organizational features and security issues.

It is worth mentioning that this categorization allows a better visualization of available information, since metrics and, especially, indicators may have the most varied forms and levels of detail. Thus, we realized that to categorize the information collected is more useful because there were cases where the objective of the measurement was the same, but the entities and attributes used were different.

Figure 2 presents an overview of categories of information collected from the proposed measures, metrics and indicators. It is perceived that there is a slight preponderance of information related to teams characteristics. This fact reinforces that information about projects teams can provide good indicators of risks in multiple project environments, and that views about the skills and capacity of teams should have more weight in measuring the risk level of a project.

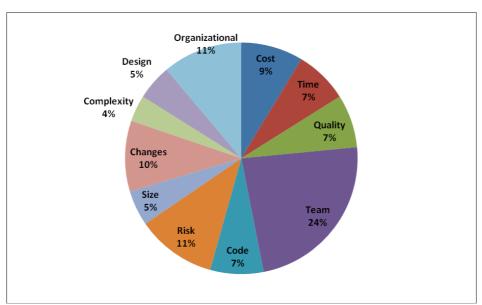


Figure 2. Information category overview

Regarding the Risk category, we found that most of the results use the metric Risk Exposure as a way of risk measurement. Another highlight is the presence of information related to Organizational factors, which demonstrates a strong influence of risks in the context of processes. Maturity level usually appeared also at Organizational category, so, it can be an indicator of risk level of an organization, allowing, for example, the decision-making about any kind of investment on an IT organization.

In what follows, we describe and analyze each category of metrics or indicators found.

Changes: it comprises changes in scope, requirements, development, and any kind of change could be seen as risk indicators. So, some works affirm that changes could strongly impact in the environment. Kulk and other authors [19] present a set of quantifiable risk drivers for IT environments. One of them considers that aspects of changes in projects, such as in user requirements or changes in scope, proposed by [20] and [21] as well. Regarding changes in requirement, Gupta [22] proposes a metrics called "Mission Critical Requirements Stability Risk Metrics (MCRSRM)", which assess risk of software product in terms of changes in requirements.

Code: some works present source code-based metrics for software risk assessment. Izquierdo-cortazar and other authors [23] and Hosseingholizadeh [24] show proposals that could be helpful in reducing the unused code due turnover and that provide an analysis of the software product to reduce bugs just using information from source code. This kind of information could be helpful when the set of projects use the same programming language.

Complexity: only Singaravel and other authors [25] propose a measurement for complexity. Its complexity is related to the product, and this paper only establishes that the complexity of a product is a risk indicator but do not detail the calculation method. On the other hand, it is important to define different indicators considering also the complexity level of a project. If a project is complex, it is somewhat harder to implement because of several types of information, such as project or product size, dependence level, function points, and so on.

Cost: the proposed metrics and indicators are specific to financial risks, so that they encompass ideas from Economics. In fact, they are methods proposals of risk assessment given the depth of the functions for calculation aiming to better view chances of return of investment. For example, Pajares and Paredes [26] propose a metric called Cost Control Index, which is used to monitor the cost variance during a project life cycle using Earned Value Management Analysis (EVM). Other methods are: Mean Semideviation Behavior [27] – a method for risk measurement to project selection –, Value at Risk [28] – an existing metric which is combined to help in the balancing of portfolio of engineering and contracting projects –, and Risk Pricing [29] – a specific proposal for software development whose use ideas and insights from financial market to evaluate risks.

Design: risk of product design process is highlighted in this category. Customization degree is proposed by [30] and [17]. The former is related to software projects whereas the latter concerns to supply chain. The idea is similar for both works: when the customization degree is high, it is easy the reuse components for production, hence the efficiency to produce software with good quality is improved. Additionally [17] proposes a risk metric that combines the customization degree and dependence level between product components.

Organizational: according to four references [19, 29, 30, 31] the level of maturity of an organization or process is useful to indicate the organizational risk. An empirical study was performed by Wu [31], where shows that some risk factors are reduced when the organizational maturity is high. Some aspects about safety are described in [20].

Quality: most of the papers are related to software testing activities. Souza and other authors [32] propose metrics to support the prioritization of test cases based on risk exposure of requirements. They define the number of test cases and change requests as a source of information to measure technical risks, once the test coverage works better when consider risk analysis before testing process. Number of errors is mentioned by Sultan and other authors [33]. This kind of information seems to be relevant specially to address product risks according a database of errors in past similar products.

Risk: this category just encompasses found methods and techniques of risk assessment. The Risk Exposure calculation (Probability x Impact) is cited by 5 papers [22, 32, 34, 35, 36]. Wang and other authors [37] suggest the combination between probability of risk factor and concepts of Utility Theory, proposing a function called Utility of Risk Factor, which basically estimate the performance of a project. Number of identified risks is suggested in [32], using this information to visualize risks per class. Another proposal is defined in [38] suggesting the calculation of risk dependence between different projects.

Size: two kinds of size were found: project and product. The idea is: when bigger a project/product is, there are more associated risk factors. Regarding project size, Sauer and other authors [18] say that project size is an excellent risk indicator for IT projects, and it made a survey which verify that there is a strong relationship between project size and its performance. About product risk, Kulk and other authors [19] suggest measuring the software, whereas Sultan and other authors [33] suggest number of requirements as a information to product size.

Team: even these numbers do not tell us so much about how to define indicators, we can highlight a interesting result: almost a quarter of the raised information considers people as an important aspect to take into account during the process of risk assessment. And it is clearer especially considering software development environment. Perhaps risks factors related to staff must be seen more carefully during the process of risk identification and analysis in projects. Fiedler [39] discusses risk factors related to resistance due to changes in organizations. In this paper some indicators refer to job satisfaction, rate of sick leave and turnover. Turnover is also mentioned in [35] and [23]. There is information about team size [19, 34, 38]. Team skills are mentioned considering experience [19, 40] and knowledge levels [20, 22] on certain technology or platform. There are also proposals of effort estimation based on the team and time to perform, as described in [32] and [18]. Motivations levels is considered as an important measure for software development environments [30, 40], once the software processes specifically depends a lot from the staff. And finally, some papers discuss about the importance of communication, so that the communication level must be measured to assure projects or organizational successes [20, 30].

Time: the information found about this category is essentially based on project schedule. Pajares and Paredes [26] define a measure called Schedule Control Index, which allow the schedule estimation and monitoring integrated to EVM analysis. A set of indicators is proposed by Turnbaugh [20], and it suggests, but not details, that it is important to define indicators using the schedule as main source of information.

3.3 Threats to Validity

Despite the correct application of research methods, it is also important to highlight the limitations of this paper, so that we can identify points of improvements for future work in this area. In general, we can list the following threats to validity of this systematic mapping study:

• It is based only on three indexing system (Scopus, Engineering Village and IEEE). Although they are useful because they index publications from a large number of publishers, the application of empirical

research in software engineering is still limited [41], which impacts on the application of systematic reviews and mapping studies;

- The restricted number of years (2004 2010) can negatively impact on the results. We just consider these years to consider the state of art of recent works, reflecting the currently overview of the community;
- One single person made the selection of works. It implies that some papers might have been included or excluded incorrectly, even with a clear research question and a protocol built with the authors together;
- The classification of data extracted may have errors because we created it without a tool or statistical method. We just trusted on our experience on the subject to create reasonable categories. Another point is that there were cases where the data in some papers was not so clear to get, so it could skew the results.

4 Proposed Indicators

In this section we present defined indicators that support risk assessment for multiple software projects environments. The proposals are based on: i) Refinement of the found results from systematic mapping study and literature review and ii) Measurable risk factors raised from the risk-based taxonomy proposed by the Software Engineering Institute (SEI) [5].

The choice of this taxonomy is due to the fact that it provides an organized way to categorize risk factors, and it is a consolidated report in software risk management. Thus, the defined indicators are adherent to software development environments and they are based not only on theoretical aspects of bibliographical review, but also on practical application of a consolidated risk classification.

The SEI risk taxonomy provides a framework to structure and organize risks in software development projects, so that it can be used as a basis to elaboration of methods and activities of risk management. The taxonomic Classes are divided in Elements, that are characterized by Attributes. Figure 3 presents how SEI risk taxonomy is structured.

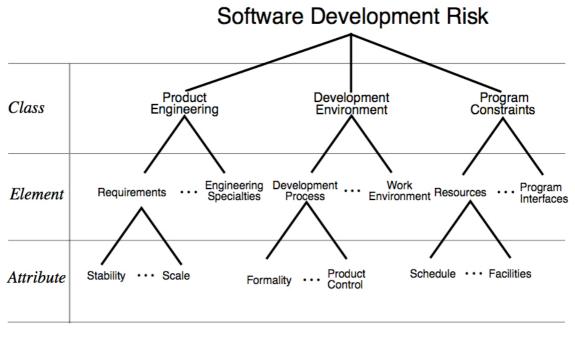


Figure 3. Software development risk Taxonomy [5]

The taxonomy classifies the risks in three big classes:

• Product Engineering: it consists of technical aspects to be performed to the development of product to be delivered to client. It refers to activities of software and system engineering, such as analysis and requirements specification, design and implementation, integrations of software and hardware components, and tests. Engineering product risks commonly derives from requirements that are hard to implement, inappropriate assessment of quality requirements or design specification and poor quality of

code implementation. This class has the following elements: Requirements, Design, Code and Unit Test, Integration and Test, and Engineering Specialties;

- <u>Development Environment:</u> this class involves risks related to methods, procedures and tools used in product development. It also includes management issues. This class has the following elements: Development Environment, Development System, Management Process, Management Methods, and Work Environment;
- <u>Program Constraints:</u> it is composed by external risk factors, i.e., they are outside of control of management, such as contractual, organizational and operational factors. The following elements are described in this class: Resources, Contract, and Program Interfaces.

4.1 Main Principles

Before presenting the proposal of indicators, this section delimits the scope of this research in order to better understand the proposal and the presented results in the next sections.

- 1. As presented before, the definition of indicators is based on the systematic mapping results, literature review and SEI risk taxonomy [5]. Therefore, the proposed indicators are organized by Class and Element according to this taxonomy;
- 2. We consider the context of multiple project risk management. Even most of the proposed indicators use information from an individual project, we also defined indicators considering aspects of the environment;
- 3. Figures 4, 5 and 6 present a quick overview of the indicators per class and element from the SEI risk taxonomy. The highlighted indicators in these figures were not explicitly mentioned by the SEI taxonomy. However they were included because of the findings in the literature, so that they are important to measure risks.
- 4. A group of specialists and professionals in project management and risk management gave a feedback to the proposal presented here. Therefore, the indicators already have some improvements. The description of the assessment of indicators and discussion about improvements can be found at Section 5.

In the next subsections, we present the proposed indicators organized by Class and Element. We did not defined indicators by attribute because several ones are strongly interrelated and the level of detail is high for the concept of indicator.

4.2 Indicators of Product Engineering

Figure 4 presents the overview of indicators related to Product Engineering class.

Requirement stability allows the monitoring of changes in requirements, checking if the change degree is acceptable. If requirements have been changing too much, there would have misunderstandings about the project scope for the team. This indicator can be correlated to **Requirement clarity**. **Requirement completeness** allows that requirements considered more important will be better specified, allowing their better understanding.

Software size can also be estimated according to the number of requirements combined to another information about product, such as presented Function Points. Thus, a bigger system could require an increased attention. The same happens **requirement complexity level** and **requirement dependence**, so that they may be helpful in the better perception of factors that could difficult the implementation of a requirements set, especially if they are considered as priority.

An important aspect in the software development process is the possibility to reuse components, allowing the rapid application development. In this sense, in an environment of multiple projects, it is valid to consider that when there are more reusable components, the software evolution will be better, as so as the sharing of software resources in the environment. Another point to consider to measure is the number of interfaces, so that it can help to achieve the satisfactory integration between different tools and frameworks or avoid a excessive number of interfaces. Therefore, this kind of information may allow the review of related factors, such as quality of specification of external and internal interfaces, difficulty level to integrate components and maintaining.

The indicator **Number of programming languages** is more related to the number of interfaces of the product. Therefore, the development of products that involves different technologies, environments or frameworks tends to use different programming languages. This indicator allows more perception about interfacing points within the product, collaborating for the optimization of software development activities. Unit tests enables the early identification of bugs, so it prevents bigger problems, due to the bug in this case is more easily identified. Inside

this context, the number of unit tests can be considered as a risk indicator for organizations that encourage this kind of test. On the other hand, it is important to keep in mind that when we count unit tests, we must analyze the quality of specification of these unit tests.

Requirements

- Requirement Stability
- Requirement Completeness
- Requirement Clarity
- Software Size
- Requirements Dependence
- Requirement Complexity

Design

- Reuse level
- Interfacing level

Code and Unit Test

- Number of programming languages
- Units tests specification level
- Number of target devices

Integration and Test

- Number of changes requests (Product stability)
- Engineering Specialties
 - Difficult level to implement security requirements

Figure 4. Indicators of Product Engineering class

Number of target devices is helpful in the definition of an appropriate environment to test systems under development, avoiding that the client or user identify errors caused by the usage of different devices.

In the element Integration and Test only the indicator **Product stability** is defined by the calculation of product stability expressed by the number of software change requests, that is a common tool in software development environments.

According to SEI taxonomy [5], the element Engineering Specialties covers system requirements that require aspects that are commonly approached by specialists that are not full-time in a project. The goal of the indicator defined in this element is to give special treatment to security requirements, measuring the difficult level to implement certain security policies in the software, and to ensure that its specifications are correct.

4.3 Indicators of Development Environment

The overview of indicators of development environment class is presented in Figure 5.

The indicator **Experience on the development process** aims to assess the influence of experience on the process, avoiding misunderstandings or bottlenecks. It is also useful to identify if it is necessary to qualify team members on certain development process. **Geographical distribution level**, in turn, just try to identify and analyze risk factors related to distributed development, considering information such as Number of development sites versus distance versus number of native languages versus time zone difference

Development hardware availability identifies and analyzes risk factors related to the infrastructure availability to software development and Development software availability focuses on the availability of software tools. Therefore, these indicators help in visualizing of risk factors that will be generated due a poor availability of necessary resources to development environment. These indicators are measured by type, since it is common to exist different tools or devices with the same purpose. Examples of types: a software tool for requirements specification and a micro server.

Indicators of the element Management Process bring information about risks factors related to management activities. **Project manager experience level** on managing assesses the experience level of the project manager. It is an important indicator to help in allocating the managers according to the size, complexity or type of a project, for example.

Development Process

- Experience on the development process (team)
- Geographical distribution level
- Development System
 - Development infrastructure availability
 - Development software availability
- Management Process
 - Experience level on managing (project manager)
 - Project dependence level
 - Process changes
 - Maturity level
- Management Methods
 - Motivation levels
 - Role organization effectiveness
- Work Environment
 - Conflict level
 - Team focus
 - Turnover

Figure 5. Indicators of Development Environment

Looking more on the view of multiple project management, **Project dependence level** could be helpful to measure the dependence levels between projects. Some projects share resources such as people, devices, tools, so that according to the degree of dependence, the indicator provides decision-making for optimization of projects. Therefore, these points of integrations must be followed during project life cycle, allowing the monitoring of possible delays caused by these dependencies.

Analyzing aspects of process management in an organizational insight, indicators **Process changes** and **Maturity level** approach aspects of changes and maturity or capability levels in the organization, respectively. Process changes is a specific indicator for organizations that are changing processes, so that it can help in establish the correlation between these changes and the performance of the projects. Maturity level uses the following idea: when maturity level is high, lower will be risks, as presented in some findings of the mapping study. It can be helpful in defining investments in organizations and the combination between maturity level and other sources of organizational information helps in monitor the organization awareness to risks, for example.

Motivation level defines team motivation levels as a risk indicator, allowing viewing points of improvement in organizational processes, impacting then in the performance of projects. Another proposed indicator is the clarity of roles effectiveness. This kind of information assists managers in processes improvement, increasing the chances of goals meeting.

Conflicts between staff members, depending on the type and degree, can ruin a project. In this light, Conflict level supports the assessment of risk factors related to conflict management. The idea is to provide an early identification and action plans in order to reduce and control conflicts. Team focus just checks if the project members really understand the mission and project goals. Thus, the monitoring of this indicator can be helpful in reducing misinterpretation about the project scope.

Turnover also must be controlled in order to maintain it in an acceptable level. It is commonly calculated as the number employees leaving, divided by the total number of employees, multiplied by 100. Most of software processes depend on people, and its replacement may be dangerous for projects. Therefore, decisions may be made, seeking to reduce turnover levels and evaluate the impact of turnover on projects. It is important to notice that the turnover may suffer some adaptions, since we are assessing turnover per project, so that there are cases which employees leave a project, but not the job.

4.4 Indicators of Program Constraints

The SEI taxonomy says that Program Constraints class refers to external aspects of a project or risk factors outside control. But in fact, most of the risks raised on the taxonomy refer to events that can affect a project. There are three elements: Resources, Contract and Program Interfaces. The first one comprises most of risk factors,

describing risks related to schedule, staff, and budget. The others elements really are more linked to external factors, such as business, client and contract aspects. Figure 6 presents the overview of indicators of this class.

Resources

- Team knowledge level
- Team experience level
- Team availability
- Team size
- Project complexity
- Project criticality
- Project size
- Risk points
- Financial feasibility

Contract

- External dependence level (product or service)
- Program Interfaces
 - Business value
 - Client participation level
 - Experience of the client

Figure 6. Indicators of Program Constraints

Defined indicators from information about team skills are presented in **Knowledge** and **Experience** of the team, assessing knowledge and experience levels on certain technology. These indicators help in identifying team profile and assess the possibility of to train or contract new people according its technical capabilities. Identifying this type of risks, project managers can define more realistic schedules.

Availability of the team considers team availability according to the premise that there would have shared human resources in different projects. Therefore, this indicator aims to analyze projects risks in the viewpoint of team availability per project. In this context, Team size helps in viewing team size, verifying if the number of members per project is acceptable.

Technical information about a project is described in **Project Complexity**, **Project Criticality**, **Project Size** and **Risk Points** [42]. To measure how much critical or complex is a project may allow decisions about the feasibility of a project. Critical projects must more careful management regard others, once it is supposed to have a lot of restriction, especially of cost and time. Hence, project management indicators like CPI (Cost Performance Index) or SPI (Schedule Performance Index) or product metrics like "number of transactions" can also help in the definition of risks. Regarding complexity, the same happens: if the project is difficult to develop, it may comprise a project risk so that decisions in order to reduce the levels of complexity are necessary. We can consider as measures of complexity: size of a Work-Breakdown Structure, number of dependence between projects or systems, Function Points or Story Points for Scrum-based projects.

Risk Points [42] is a way of to measure the risk level of a project. According to the author, the objective of Risk Points is to assess the complexity of a software project based on number of identified risks.

Financial Feasibility is helpful to perform an ante project risk assessment, verifying if the estimated cost is really feasible. This indicator also is helpful to reallocate resources to projects considered as priority for the organization.

External dependence level brings information about project external dependences, such as products or services. It helps in minimize or avoid risks specially related to outsourcing. **Business value** tries to identify whether there is similar products in the market. The idea is to help stakeholders to maintain the project value acceptable. Indicators about client are described in **Client experience level** and **Client participation level**. They indicate risks about the client involvement and experience, so that would be helpful in mitigate risks due to client, enabling action plans in order to not prejudice the project.

4.5 Applications of the Proposed Indicators

Indicators are basically tools for decision-support, knowledge and control of measured entities and attributes. The usage of indicators in environment of projects allows the better perception of aspects related to the environment, especially processes, products (software and related artifacts), resources and projects.

The application of indicators for risk assessment enables the simultaneous risk identification and analysis. The risk is identified with the collection of information by extracted measures from the environment, and at the same time the level of risk is already defined, allowing its analysis and the definition of appropriate strategies for risk factors – risk planning. Another point to highlight is that the use of indicators in this sense facilitates the risk controlling, because it allows analyzing the effectiveness of actions through the project lifecycle.

The proposed indicators have a reasonable range, which allows the application in different managerial levels. In addition, the structured view of these indicators allows the allocation of responsibilities on certain risk factors, depending on the priorities of the organization. This categorization, due its adherence to environment of software projects, facilitates the deployment and adaption of indicators as well as the better understanding of risks.

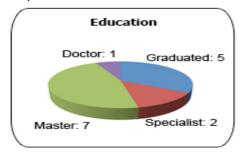
It is important to mention that extracted information of the indicators, if well structured, can provide the historical view of risks per project, allowing the application of benchmarking inside the organization. Therefore the collecting of experiences helps to consolidate a better organizational knowledge, increasing the chances of success of new and ongoing projects.

Additionally the indicators can give a complementary view together with the use of other techniques of risk identification and analysis. In this light, the view of indicators is about an additional tool that aims to better manage risks in the environment of multiple projects, since the activity of risk assessment is consensus in the main existing models of risk management in software projects.

5 Proposal Feedback

A relevant strategy to identify points of improvement in this proposal is performing an assessment with specialists of the domain at issue. To assess the proposal, we elaborated a questionnaire that was available on line, whose goals were: i) To assess the proposed indicators and ii) To assess the research methodology adopted to define the indicators.

This questionnaire was made available to a restricted group of professionals and academics with expertise in project management and risk management in the area of software engineering. Fifteen (15) people participated in this feedback. We selected the tool Google Docs² to design the questionnaire and to assist the analysis of results. All the participants have proven knowledge on Software Project and/or Risk Management through publications, graduation, master or doctoral works, and practical experience on the domain. Figure 7 shows the profile of the participants. In the information about Current Position, the participants could select more than one option.







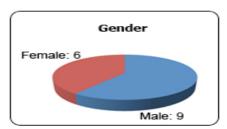


Figure 7. Profile of participants in the feedback

² Google Docs is available at: http://docs.google.com

The questionnaire, available on-line, had two parts. The first one aimed to assess the defined indicators, whereas the second part aimed to evaluate the research methodology to define indicators. The second part used the quality criteria proposed by CASP – Critical Appraisal Skills Programme [43]. Besides the questionnaires, each participant received a readme file containing main concepts and a brief about the research methodology adopted and results. In the next subsections we present the application of questionnaire and the feedback of each part in more detail.

5.1 Indicators Feedback

In order to make the questionnaire not too long for one participant, each one assessed only a class of indicators. The selection was randomized and designed as follows: Product Engineering (5), Development Environment (5) and Program Constraints (5). Each participant was asked to read each indicator and to select its level of agreement (Totally Agree/Partially Agree/Disagree). Each indicator was followed by a brief description and the criteria for agreement or disagreement was: understanding level, relevance and feasibility. There was also a field for comments, whose filling was optional but we asked participants to fill especially in case of disagreement for suggestions, considerations and identification of improvement points.

Tables 7, 8 and 9 present the results for the class Product Engineering, Development Environment and Program Constraints, respectively. Each table presents the indicators organized by element, and the number of records of agreement level: Totally Agree (<u>TA</u>), Partially Agree (<u>PA</u>) and Disagree (<u>D</u>).

Five participants evaluated the indicators of the class Product Engineering, including Masters (3), one Specialist and one Graduated. Three participants consider themselves with advanced knowledge on risk management and two with basic knowledge. About Current Position, we have: Researcher/Professor (2), Software Engineer/Analyst (2), and Researcher/Professor and Software Engineer/Analyst (1).

Table 7. Indicators feedback of Product Engineering class

Indicator	TA	PA	D
Requirement			
Requirement Stability	5	0	0
Requirement Completeness	2	3	0
Requirement Clarity	4	1	0
Software Size	3	2	0
Requirement Dependence	4	1	0
Requirement Complexity	5	0	0
Design			
Reuse Level	5	0	0
Interfacing Level	4	1	0
Code and Unit Test			
Number of Programming Languages	5	0	0
Unit tests specification level	3	2	0
Number of target devices	3	1	1
Integration and Test			
Product Stability	4	1	0
Engineering Specialties			
Difficult level to implement security requirements	4	1	0

In general, according to the comments of the participants, the indicators of the class Product Engineering were well evaluated. Cases of partial agreement were justified through suggestions of improvement points. We highlight:

- Requirement completeness: the analyst who wrote the requirements may not be the same who is going to develop the system. And in the viewpoint of the developer, some requirements could be incomplete. It is important to consider additional information to measure requirement completeness also considering the viewpoint of the requirement developer.
- Software Size: consider not only the number of requirement as a measure, but also information such as number of lines of code and function points.
- Requirement dependence: include the type of dependence, because there are different ways to treat risks due to the type of dependence, for example, showing how much critical the dependence is.

• Unit tests specification level: there could have so much unit tests, but badly-specified. Thus, it is also important to analyze the quality of unit tests specification when to use this indicator.

Six participants evaluated the indicators of the class Development Environment. Four are graduated, one is master and one is doctor. Regarding knowledge about risk management, two consider themselves as experts, two has intermediary knowledge and one has basic knowledge. Current positions of these participants are: Professor/Researcher (1), Professor/Researcher and Manager (2), Manager (1) and Software Engineer/Analyst (1). Table 8 presents the results for each indicator.

Table 8. Indicators feedback of Development Environment class

Indicator	TA	PA	D
Development Process			
Experience on the development process	4	1	0
Geographical distribution level	3	2	0
Development System			
Development infrastructure availability	4	1	0
Development software availability	4	0	1
Management Process			
Project manager experience level on managing	4	1	0
Project dependence level	4	1	0
Process changes	3	1	1
Maturity level	4	0	1
Management Methods			
Motivation level	4	1	0
Role organization effectiveness	4	1	0
Work Environment			
Conflict Level	3	1	1
Team Focus	5	0	0
Turnover	4	1	0

We realized that in most of cases of partial agreement and disagreement by the participants are related to improvements in the description and suggestions of other measures in order to be clearer. These are the main considerations according to the comments of the participants:

- Geographical Distribution level: consider other aspects to measure such as idiom and time zone too.
- Software development availability: instead of mentioning the number of specific software, it is also important to consider the type of software, since there may be several different tools of the same type. Another important information is to relate this indicator with knowledge of the team about certain tool.
- Process changes: this indicator could be seen as an opportunity, once some changes can be positive for the organization. On the other hand, changes are also associated to chaos, so that it is important to control their level in order to not make them excessive.
- Conflict level: this indicator may be very subjective. In this sense, it must be reviewed in order to identify
 ways to measure this kind of risk factor.
- Turnover: take into consideration the fact that a resource can leave the project but it does not necessarily mean that there is a case of resignation.

And finally, Table 9 shows the results of indicators evaluation of the class Program Constraints. We got 5 participants: 3 Masters, 1 Graduated and 1 Specialist. Four of them consider themselves with advanced knowledge in Risk Management and 1 with intermediary knowledge. Regarding current position, each one has a different position: Software Engineering/Analyst (1), Manager (1), Researcher/Professor (1), Developer/Technician (1) and Configuration Manager (1).

The indicators of the class Program Constraints also have been well evaluated in general. Specifically in this class the participants perceived that these indicators could be combined with each other to allow the assessment and monitoring of risk factors.

Table 9. Indicators feedback of Program Constraints class

Indicator	TA	PA	D
Resources			
Team knowledge level	4	1	0
Team experience level	5	0	0
Team availability	3	2	0
Team size	5	0	0
Project complexity	3	1	1
Project criticality	3	1	1
Project size	4	1	0
Risk points [41]	2	3	0
Financial feasibility	4	1	0
Contract			
External dependences level	3	2	0
Program Interfaces			
Product business value	1	3	1
Client participation level	2	2	1
Client experience	3	2	0

These are the main comments provided by the participants regarding Program Constraints indicators:

- Number of external dependences: it is important to compute the type of dependence or contract, considering some criteria for external product or service such as, whether it is paid or free, whether there is technical support, and whether there is planning of evolution or renewing licenses.
- Product business value: depending on the type of product, it may be difficult to measure customer satisfaction. Another point is the fact that business value could be difficult to measure, because it involves different and complex factors related to the market. Therefore, this indicator must be well analyzed in practice and requires validation from specialists in business management too.
- Client participation level: take into consideration the development process of the project. For example, the participation level of a client in a project that adopts RUP is different from a project that adopts Scrum. Another good suggestion given was to combine this indicator with the measure of client satisfaction through periodical validations.
- Complexity, Criticality and Risk Points: although they are considered relevant and powerful, their measurement process is not a trivial task. In fact, these indicators comprise the assessment of a project and the process of data collecting is the combination of different metrics or even indicators. Therefore, these indicators must be analyzed further considering the associated concepts, several variables and appropriate information structure to drive projects.

5.2 Research Methodology Feedback

To specifically evaluate the quality of research methodology to define indicators (second part of the questionnaire), all the participants performed an independent analysis, according to eight defined quality criteria. These criteria are adapted from CASP – Critical Appraisal Skills Programme [43], as a way to assess the quality of qualitative researches according to the principles of good practices in conducting empirical researches in software engineering.

CASP is focused on three big issues: Rigour, Credibility and Relevance. A set of questions are designed to approach these issues, which lead to think about:

- Rigour: is there a thorough and appropriate approach been applied to main research methods?
- Credibility: are the findings well presented and meaningful?
- Relevance: how useful are these findings to the researcher and the respective research?

The second part of the questionnaire is subdivided in two sections: the first one presents screening questions that are related to the quality of the research considering objectives and context. The second section draws detailed questions that are related to the three issues presented before – rigour, credibility and relevance. For each question, we let an optional field for comments.

Table 10 shows the number of responses per question. Questions 1 and 2 are screening questions whereas the remainders are detailed questions. To answer this part of the questionnaire, we provided a document containing a brief description about the research methodology and main results.

Table 10. Research methodology feedback per question

Question	Yes	No
1. Was there a clear statement of the aims of the research?	14	1
2. Is the research methodology appropriate?	14	1
3. Was the research design appropriate to address the aims of the research?	13	2
4. Was the recruitment strategy appropriate to the aims of the research?	13	2
5. Were the data collected in a way that addressed the definition of indicators proposal?	13	2
6. Has the relationship between researcher and the proposed indicators been adequately considered?	8	7
7. Is there a clear statement of findings?	9	6
8. Are the indicators valuable to research or practice?	14	1

In general, the research was well evaluated by the participants. Most of disagreements are related to limitations of the readme file, which contained an only very brief overview about the work. We tried to elaborate a succinct document describing only the main aspects of this research. This document was drafted in order to get a reasonable number of participants so that they can analyze the main points of the proposal. On the other hand, we believe that these disagreement points are better described on this work, as mentioned by some participants.

There were relevant suggestions of improvement by the participants. Most of them describe the research as appropriate for now, but they suggest considering more empirical studies in practice. The usage of face-to-face evaluation (brainstorming, focus group or interview) with professionals in project management also was mentioned.

6 Final Considerations

The main objective of this paper is to define a set of indicators that can support the activity of risk assessment – risk identification and risk analysis – in environments of multiple software projects. To achieve this goal, we adopted a methodology that is strongly based on the literature review.

In fact, the purpose of this paper work became from the following question: why do not we measure risks? We realize that risk management in software engineering is still a restricted subject and it is poorly absorbed by industry as it stands today. On the other hand, we also realize that many projects still fail [44, 45] because of factors that would have been better controlled or known. That is, the existence of risks is unquestionable, but they are not perceived because of a set of reasons related the organization constraints and it respective process. This is where risk management would be effective as an instrument to support decision-making in order to minimize or avoid negative events while maximize opportunities. The indicators, in turn, are important managerial tools in this issue of decision-making process.

Within this context, we initially performed an *ad hoc* literature review in order to study the main software risk management models. At the end of this initial study, we noted a gap with respect to the usage of metrics in risk. Therefore, we performed a thorough and rigorous review: the systematic mapping study. Still, an interesting aspect perceived in the end of the mapping was the relatively small quantity of papers in the literature about the subject of this paper. This number probably would be less if we consider the analysis of the selected works according to quality criteria defined in advance. Finally, these findings were combined and organized according to SEI taxonomy [5], allowing the adherence of the proposed indicators to environments of software projects.

The main contribution of this work undoubtedly addresses the use of measurement through indicators in the area of risk management in environment of software projects. Therefore there is a set of indicators whose goal is to assess risks. The idea is to enable the risk identification and risk analysis by capturing information from project and the respective environment.

The categorization of the indicators according to SEI risk taxonomy [5] is important, because it facilitates the use and even the definition of correlations between the involved measures. Thus, project managers have at their disposal a set of information that allows the assessment of risk factors related to the product, the environment, the project itself and also external factors.

Another contribution to be highlighted is to bring the issue to the literature, since many practices of project management are based on experience, practice and feeling of the manager. This fact occurs specially regarding risk management. We know that risks that may affect negatively a project are around everywhere, but their management is usually done implicitly. With this work we expect to arise more interest of software engineering community regarding the use of risk management more effectively.

The fact that professionals and specialists in the area evaluated the indicators is also a determining factor to give more credibility to this work. Certainly the results are preliminary and the proposal requires further adjustments, but the initial achievements evidences that the work has value for both researchers and practitioners. Likewise, the use of the CASP method [43] enabled an analysis from a qualitative viewpoint of the research methodology and the proposal itself. Thus it allowed a critical analysis about the application of appropriate research methods to a multidisciplinary area as project management, more specifically risk management.

Considering that this work used systematic mapping study, it was supposed to find a greater quantity of selected papers in relation to a systematic review. But in this case we can highlight that there is a relatively small number of works in the literature about the process of risk measurement in project environments. In fact, inside IT and software project management there is a gap about this subject. Therefore, the study of risk measurement in software development environments should be seen more carefully, taking into account the aspects of software processes, especially with the increasing of agile methods, which requires a more sophisticated development culture, impacting directly on software processes.

Another challenge is to transform data and information into valuable knowledge, as discussed by Holzmann and Spiegler [46]. Perhaps the usage of information and data to assess risks can make the risk management an activity most often used in software projects environments, by making it less subjective. On the other hand, the selection of information that is necessary for decision-making is not a trivial task, and it requires good skills from project managers to consolidate data into useful and representative indicators.

Definitely this paper raises a different way of thinking about risk management in software projects, since risk factors can be measured, allowing thinking about a model of risk-driven management. In this sense, the contributions brought by this paper can provide initial ways in the designing of this model through the availability and use of information that indicate risks.

6.1 Limitations

This work has limitations and areas of improvement that need to be better analyzed in the future. We can highlight:

- The subject is little mentioned in the literature. Separating risk management and software measurement, there is a consensus that both areas are relevant. However, the limited availability of approaches in the context of this paper makes this work more difficult. This is better perceived due to the small quantity of works found in the systematic mapping study and due to the little variety of works in general about software risk management.
- From the methodological viewpoint, the application of a systematic mapping study on an area that is very practical may bias the results of this work. It is also important to notice that the definition of indicators had a little involvement of the industry. We only used a consolidated risk-based taxonomy and a preliminary assessment of professionals of the area. Likewise, the systematic mapping study has its limitations, as presented at Section 3.3.
- The proposed indicators were not executed in practice. There was no empirical study performed on these indicators. Thus we cannot affirm that the indicators are effective in practice. We just can affirm that there are clues that they are representative for the area due to the usage of literature review together with a preliminary assessment through a restricted group of professionals on the domain and scope of this paper.
- The provided feedback also has limitations. The first one is the reduced number of participants. The second one is the questionnaire itself. The influence of the researcher can eventually bias the answers, especially because of the qualitative focus of this feedback. This limitation could be minimized through the practical involvement of the participant with each indicator, as well as the use of other techniques of direct collection of data beside the questionnaire.
- Since the fill of the place to additional comments in the feedback was optional, some cases of Partial Agreement or Disagreement were not justified by the participants. This fact reinforces the need of a deeper analysis of the proposed indicators.

6.2 Future Work

With the results of this paper, we expect to evolve the contributions, identify new opportunities of research in the area, and settle the main limitations. For future work, we can enumerate:

- Perform a deeper analysis of the indicators and with a larger sample of participants, and consider others techniques beside questionnaires;
- Improve the protocol of mapping study considering more sources and analysis of quality criteria;
- Apply the proposed indicators in a real environment by empirical studies, evaluating the impacts and identification of points of improvement. It is also important to consider different types of environment and different development processes;
- Identify more measurable risk factors through research on different taxonomies and classification of risks;
- Define a appropriate methodology for works in the area of software risk management, specially to define risk factors;
- Define better the indicators associated to the assessment of projects, such as Complexity, Criticality and Risk Points. They involve several sources of information many times unavailable. Therefore it is necessary to structure the information in order to provide a reliable assessment of projects based on risks;
- Identify models of information and knowledge representation related to risks, allowing the usage of information of similar projects to assess risks in new projects;
- Design a risk-driven management model so that it would aggregate information from different management levels. The idea is to allow the use of techniques supported by database and knowledge base in an organization with the goal of to support managerial decision-making based on risks;
- Define software tools and processes that support the proposed indicators;
- Define indicators for positive risks, i.e. assessment of project opportunities.

Acknowledgements

The authors would like to thank FACEPE and CNPq for the financial support given to this work.

References

- [1] C. Gusmão; *Um modelo de processo de gestão de riscos para ambientes de múltiplos projetos de desenvolvimento de software*, Doctoral Thesis in Computer Science, Centro de Informática. Universidade Federal de Pernambuco, Brazil, 2007.
- [2] T. De Marco. *The Deadline: A Novel About Project Management*. Dorset House Pub-lishing, New York, 1997.
- [3] E. Rios. Análise de Riscos em Projetos de Teste de Software. Alta Books, 2005.
- [4] B. W. Boehm, "Tutorial: software risk management", IEEE Computer Society Press, 1989.S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
- [5] M. J. Carr, ,S. L. Konda, I. Monarch, F. C. Ulrich and C. F. Walker, *Taxonomy based risk identification*. Software Engineering Institute, Carnegie Mellon University, USA, 1993.
- [6] R. Charette, *Implementing risk management best practices*. Carole Edrich, 2001.
- [7] D. W. Karolak, "Software Engineering Risk Management", IEEE, Washington, DC, USA, 1996.
- [8] J. Kontio, Software engineering risk management: a method, improvement framework, and empirical evaluation, Doctoral Dissertation, Helsinki University of Technology, Finland, 2001.
- [9] J. S. Persson, L. Mathiassen, J. Boeg, T. S. Madsen, F. Steinson, "Managing risks in distributed software projects: an integrative framework", IEEE Transactions on Engineering Management, v. 56, n. 3, p. 508-532, Aug. 2009.
- [10] T. R. Trigo, C. M. G. Gusmão and A. V. Lins, "CBR Risk Risk Identification Method Using Case Based Reasoning", In 5°CONTECSI International Conference on Information Systems and Technology Management, São Paulo, Brazil, 2008.

- [11] O. Perminova, M. Gustafsson and K. Wikstrom, "Defining uncertainty in projects a new perspective". International Journal of Project Management, v. 26, n. 1, p. 73-79, Jan 2008.
- [12] PMI Project Management Institute, *A Guide to the Project Management Body of Knowledge*, ANSI/PMI 99-01-2004, Project Management Institute, FourCampus Boulevard, Newtown Square, USA, 2004.
- [13] Heldman, K; Project manager's spotlight on risk management. Jossey-Bass, San Francisco, USA, 2005.
- [14] L. Ribeiro, *Processo ágil para gestão de riscos em ambientes de múltiplos projetos*. Master Dissertation in Computer Engineering, Escola Politécnica de Pernambuco, Universidade de Pernambuco, Brazil, 2009.
- [15] B. Kitchenham and S. Charters, *Guidelines for performing systematic literature reviews in software engineering*. Technical report. EBSE 2007.
- [16] J. Biolchini, P. Mian, A. Natali and G. Travassos, *Systematic review in software engineering*. Technical Report, Federal University of Rio de Janeiro, Brazil, TR ES 679/05, 2005.
- [17] M. Esterman, "The Development of Project Risk Metrics for Robust Concurrent Product Development (CPD) across the Supply Chain," *Concurrent Engineering*, vol. 13, no. 2, pp. 85-94, Jun. 2005.
- [18] C. Sauer, A. Gemino, and B. H. Reich, "The impact of size and volatility on IT project performance," *Communications of the ACM*, vol. 50, no. 11, pp. 79-84, Nov. 2007.
- [19] G. P. Kulk, R. J. Peters, and C. Verhoef, "Quantifying IT estimation risks," *Science of Computer Programming*, vol. 74, no. 11-12, pp. 900-933, Nov. 2009.
- [20] L. Turnbaugh, "Risk Management on Large Capital Projects," *Journal of Professional Issues in Engineering Education and Practice*, vol. 131, no. 4, p. 275, 2005.
- [21] C. Verhoef, "Quantifying the value of IT-investments," *Science of Computer Programming*, vol. 56, no. 3, pp. 315-342, May 2005.
- [22] D. Gupta, B. Road, and C. E. Section, "Software Risk Assessment and Estimation Model Department of Computer Engineering," *Risk Management*, pp. 0-4, 2008.
- [23] D. Izquierdo-cortazar et al., "Using Software Archaeology To Measure Knowledge Loss in Software Projects Due To Developer Turnover *," *System*, pp. 1-10, 2009.
- [24] A. Hosseingholizadeh, "A source-based risk analysis approach for software test optimization," 2010 2nd International Conference on Computer Engineering and Technology, pp. V2-601-V2-604, 2010.
- [25] G. Singaravel, V. Palanisamy, A. Krishnan, "Overview analysis of reusability metrics in software development for risk reduction". International Conference on Innovative Computing Technologies (ICICT) (2010) 1-5.
- [26] J. Pajares and A. López-Paredes, "An extension of the EVM analysis for project monitoring: The Cost Control Index and the Schedule Control Index," *International Journal of Project Management*, vol. 29, no. 5, pp. 615-621, Jul. 2011.
- [27] B. Jafarizadeh and R. Ramazanikhorshiddoust, "A method of project selection based on capital asset pricing theories in a framework of mean–semideviation behavior," *International Journal of Project Management*, vol. 26, no. 6, pp. 612-619, Aug. 2008.
- [28] F. Caron, M. Fumagalli, and a Rigamonti, "Engineering and contracting projects: A value at risk based approach to portfolio balancing," *International Journal of Project Management*, vol. 25, no. 6, pp. 569-578, Aug. 2007.
- [29] M. Benaroch and A. Appari, "Financial Pricing of Software Development Risk Factors," *IEEE Software*, vol. 27, no. 5, pp. 65-73, Sep. 2010.
- [30] M. Cuellar and M. Gallivan, "A framework for ex ante project risk assessment based on absorptive capacity," *European Journal of Operational Research*, vol. 173, no. 3, pp. 1123-1138, Sep. 2006.

- [31] Chui Wu, "Exploring impacts of software development process maturity on project risk," 2008 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1033-1037, Dec. 2008.
- [32] E. Souza, C. Gusmão, K. Alves, J. Venâncio, and R. Melo, "Measurement and control for risk-based test cases and activities,". *LATW'09. 10th Latin American Test Workshop*, 2009.
- [33] K. Sultan, A. En-Nouaary, and A. Hamou-Lhadj, "Catalog of Metrics for Assessing Security Risks of Software throughout the Software Development Life Cycle," 2008 International Conference on Information Security and Assurance (isa 2008), pp. 461-465, Apr. 2008.
- [34] J. Rong, L. Hongzhi, Y. Jiankun, S. Yafei, L. Junlin, and C. Lihua, "An Approach to Measuring Software Development Risk Based on Information Entropy," 2009 International Conference on Computational Intelligence and Natural Computing, pp. 296-298, Jun. 2009.
- [35] J. Rong, L. Hongzhi, Y. Jiankun, F. Tao, Z. Chenggui, and L. Junlin, "A model based on information entropy to measure developer turnover risk on software project," 2009 2nd IEEE International Conference on Computer Science and Information Technology, pp. 419-422, 2009.
- [36] R. W. Ferguson, "A Project Risk Metric," Crosstalk (4), April, 2004.
- [37] J. Wang, W. Lin, and Y.-H. Huang, "A performance-oriented risk management framework for innovative R&D projects," *Technovation*, vol. 30, no. 11-12, pp. 601-611, Nov. 2010.
- [38] T. W. Kwan and H. K. N. Leung, "Measuring Risks within a Program Consisting of Multiple Interdependent Projects," 2009 International Conference on Computational Intelligence and Software Engineering, pp. 1-7, Dec. 2009.
- [39] S. Fiedler, "Managing resistance in an organizational transformation: A case study from a mobile operator company," *International Journal of Project Management*, vol. 28, no. 4, pp. 370-383, May 2010.
- [40] N. Mustafa, "Software Risk Assessment Visualization Tool Design Using Probabilistic Inference Models," *Journal of Management*, pp. 1-3, 2010.
- [41] B. Kitchenham, T. Dybå, M. Jørgensen, "Evidence-based Software Engineering", 26th International Conference on Software Engineering, (2004) 273-281
- [42] S. Lopes, Análise e definição de métricas para o processo de gerência de riscos para projetos de software. Graduation Work in Computer Science, Centro de Informática, Universidade Federal de Pernambuco, Brazil, 2005.
- [43] CASP, Critical Appraisal Skills Programme. Available at: http://www.sph.nhs.uk. Access in January 2012.
- [44] J. Dominguez, "The curious case of the chaos report 2009". Project Smart. Available at: http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html, July 2009, Access on January 2012.
- [45] K. Emam and A. Koru, "A replicated survey of it software project failures", IEEE Software, p. 84-90, 2007.
- [46] V. Holzmann, I. Spiegler, "Developing risk breakdown structure for information technology organizations", International Journal of Project Management, v. 29, n. 5, p. 537-546, Jul. 2011.