Protocol to Design Techniques for Implementing Software Development Best Practices

Gloria Piedad Gasca-Hurtado[1(✉)], Vianca Vega-Zepeda[2], Mirna Muñoz[3], and Jezreel Mejía[3]

[1] Maestría en Ingeniería de Software, Facultad de Ingeniería, Universidad de Medellín, Carrera 87 no. 30-65, Medellín, Colombia
gpgasca@udem.edu.co
[2] Departamento de Ingeniería de Sistemas Y Computación, Universidad Católica del Norte, Avenida Angamos 0610, Antofagasta, Chile
vvega@ucn.cl
[3] Centro de Investigación en Matemáticas, Av. Universidad no 222 98068, Zacatecas, México
{mirna.munoz,jmejia}@cimat.mx

Abstract. Software engineering best practices allow significantly improving the software development. However, the implementation of best practices requires skilled professionals, financial investment and technical support to facilitate implementation and achieve the respective improvement. In this paper we proposes a protocol to design techniques to implement best practices of software engineering. The protocol includes the identification and selection of process to improve, the study of standards and models, identification of best practices associated with the process and the possible implementation techniques. In addition, technical design activities are defined in order to create or adapt the techniques of implementing best practices for software development.

Keywords: Software process improvement · Standards · Models · Protocol · software engineering

1 Introduction

The implementation of best practices for software development projects contributes to its success only if there are techniques suitable for this implementation. Identifying and selecting best practices to implement improvements in the processes of software development, it should include assessment activities and verification in order to check the quality and quantity of the selected practices.

This paper is derived from the results of research projects development between the Universidad de Medellín (Colombia), the Universidad Católica del Norte (Chile) and the scientific contributions of Centro de Investigación en Matemáticas de México (CIMAT). The research projects have focused on the definition of techniques and mechanisms to improve the way you run the software development projects successfully.

R.V. O'Connor et al. (Eds.): EuroSPI 2015, CCIS 543, pp. 115–126, 2015.
DOI: 10.1007/978-3-319-24647-5_10

Each time a technique is proposed to implement a best practice for software development, iterative activities are generated. It allows design a new technique or adapt some existing techniques.

The purpose of this paper is to propose a guide to identify, select and design techniques that contribute to the implementation of best practices for software development.

The main reason why this guide is proposed is to establish a systematic and organized way to: identify, select, validate and propose techniques for implementing best practices in software development. This guide is called protocol because it includes the definition of a systematic and repeatable process. It is formalized and documented for use whenever it is required. This protocol can be used for any research project in software engineering that requires or includes the creation of techniques for implementing best practices.

Through the protocol, a methodical guide is proposed. It is composed of steps documented in detail. This proposal aims to optimize the results of the process, allowing quickly structuring any of the best practices that require one or more techniques needed for their implementation in a software development project.

For the construction of this protocol, different guide documents have been used that give sustenance to the proposal. Among these highlights:

1. "Procedures for Performing Systematic Reviews" [1] de B. Kitchenham
2. "Systematic Review in Software Engineering" [2] de J. Biolchini
3. "Models and Standards Similarity Study method" [3] de J. Calvo-Manzano, et al.
4. "Experimentation in Software Engineering. An Introduction" [4] de C. Wohlin.

The first two sources mentioned were used as a guide to identify the different components that make up the protocol. The third source has been adopted as the standard method for conducting Phase 2 of the protocol, while the Wohlin [4] proposal is the basis for the pilot phase included in the protocol.

Due to the team being geographically distributed in different countries, it became necessary to standardize the method of work to get consistent results in various selected areas. Additionally, the formalization and dissemination of procedures and methods applied, can serve as a guide for any researcher who wants to generate some methodology or strategy to enable the implementation of best practices.

The construction of the protocol was developed based on the experience of the researchers involved in the project, besides using as an example the definition of a protocol for the development of a systematic review [1].

The protocol has been adjusted keeping in view the results gotten in its application in different cases, of which one is described in detail in the present paper, as case study to validate the usefulness of this protocol.

The use of this protocol is recommended to all researchers who wish to propose a methodology, strategy or guide to help small and medium organizations to implement best practices suggested by international standards and models.

This paper is structured as follows: The second section provides a conceptual framework. The third section presents the protocol to design techniques for implementing of best practices, motivations and context of the work done. In the fourth part, the application of the protocol is shown for the creation of a technique to support

the identification of the risks through a taxonomy, as a best practice aimed at achieving the success of software development projects. The paper ends with the conclusions drawn from the work done.

2 Overview

2.1 Process Improvement

The term *software process improvement* was originally proposed by WS Humphrey in 1988 [5]. The scope of process improvement considers to understand existing processes, evaluate and change them to improve the quality of the product that is being developed and / or reduce costs and development time [5].

It is important to note that in order to improve a process, adopting new methods or tools is not enough, organizational factors, procedures and standards that affect the process always exist.

Generally speaking, the improvement of a process is an iterative activity which considers three main stages: Measuring the current process, a step of analyzing where the process are valued and modeled and finally, the introduction of changes.

The cost of a process improvement project is a major barrier for small enterprise [6]. This has led researchers to propose "light" strategies for process improvement.

2.2 Best Practices for Software Development

Generally speaking, a practice is a method or technique used to conduct a part of a process and it describes how it is done. A *best practice* is some practice that has been incorporated on several projects and has been experimentally checked that contributes to the success of the objectives of these projects [6]. Given the above, software development best practices are recommended activities to replicate with a very high probability that the project will end successfully.

The capability to identify and transfer the best practices in an organization is a critical factor for to get a competitive advantage. This strategy has become one of the management techniques from the second half of the nineties.

These practices are published as success factors, lessons learned and observations [6].

Jacobson et.al [7] defines a practice as a guide for dealing with some dimension of software development. The authors suggest that this term despite of being commonly used in the software development industry has no single definition.

2.3 Techniques

According to the Real Academia de la Lengua Española [8], a technique is a "set of procedures and resources from which a science or an art is served" or "ability to run anything, or to get something."

In SWEBOK guide [9], various techniques are proposed to perform the tasks and processes associated with software development, it guide to practitioners on how to perform certain activities required for successful development projects. This is how you

can find suggested techniques for requirements elicitation, or to measure software process, or techniques for the analysis and assessment quality, among many others [9].

Applying this concept to the work performed and described in this paper, it can be stated that the objective of an technique for implementation of software development best practices, is to establish the way (mechanisms, methods, tools) for organizations, especially in small setting, can adopt and adapt best practices for software development, proposed by the largest standards or international models recognized in business and academic area. This work is necessary, since these models and standards have been developed for large organizations or government entities, which causes that they are not directly applicable in small setting.

3 Protocol for Creation of Techniques to Implementation of Best Practices

The protocol to design techniques to implement best practices for software development has the following characteristics:

1. Repeatable process. It defines, describes and documents a protocol to systematically apply the procedure as many times as necessary to design techniques or adapt those that may be useful in a particular workspace.
2. Agility. The procedure is described and documented to generate results agile.
3. Process Optimization. Ability to optimize processes that rely on best practices, from a better selection of techniques.
4. Quality validation for selected techniques. The evaluation of quality of best practices selected and implemented in an organization, is facilitated when suitable techniques are used for each area process.
5. Ease of creating techniques. With the protocol, it is want to establish steps and examples that give ease of creation of techniques and effectiveness of the implementation of best practices.

The creation of protocol is based on the design science research cycles identified by Hevner [10]. This proposal involves three stages interconnected by research cycles: 1) Environment: aims to identify the application domain, 2) Design Science Research: include building processes of artifacts and its evaluation and, 3) Knowledge Base: includes the new foundations (theories and methods, experience, new products and processes) to the knowledge base of the organization. Stages one and two are connected by Relevance Cycle where research requirements are defined and field tests are conducted. During the second stage, the Design Cycle is develops, it is responsible for the creation and testing of new proposals. Stages two and three are joined by Rigor Cycle that is responsible for increasing the knowledge of an organization

3.1 Protocol Phases

The protocol is divided into 4 phases, each with a set of tasks to be developed. The phases and activities are:

Phase 1. Planning. Establish a plan of study of the process or area of susceptible process to improve for the software development.

Task 1.1. Identification of the process or process area.

Task 1.2. General description of the process or process area.

Task 1.3. General review of models and standards that may include the process or process area of interest.

Phase 2. Selection. Select and prioritize best practices that enable the achievement of the objectives to improve of a defined process.

Task 2.1. Selection of models and standards that incorporate the process area of interest.

Task 2.2. Comparative analysis of selected models and standards.

Task 2.3. Selection of best practices to be implemented

Phase 3. Standardization. Study and analysis of the implementation technique for the best practices selected.

Task 3.1. Description of the best practices selected

Task 3.2. Description of the recommended technique

Task 3.3. Creating process assets

Phase 4. Validation. It includes experimentation, evaluation and institutionalization of the process.

Task 4.1. Experimentation

Task 4.2. Quality assessment

Task 4.3. Recommendations for improvement

Task 4.4. Technique institutionalize and lessons learned

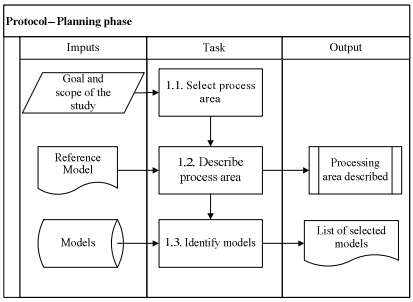
3.2 Protocol Description

The protocol to design techniques for implementing software development best practices consists of four consecutive phases, each one of them with a set of tasks to be performed. When the protocol application finalizes, results a new technique or the adaptation of an existing technique that allows software development organizations implement best practices in a particular process area, which contributes to improving development processes of organizations and software industry.

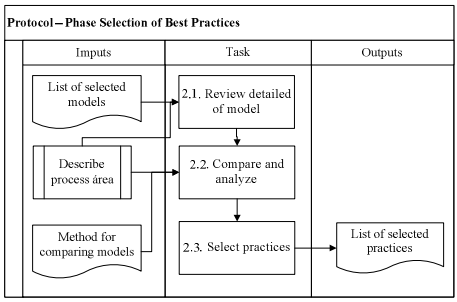
Next, phases and tasks that make up the protocol are detailed. In the explanation of each phase, will be detailed the relationship of each phase with the inputs and outputs, with the specific tasks that make up the corresponding phase. In the Fig. 1 the structure of the process to follow during each one of the phases are shown.

Phase 1. Study Plan of the Process or Process Area. The selection of the key process area is based on the experience and knowledge of the researchers involved in the project, besides the expert opinion and interest expressed by members of the companies that recognize which areas are most relevant to improve their software development processes. This phase is composed by these tasks:

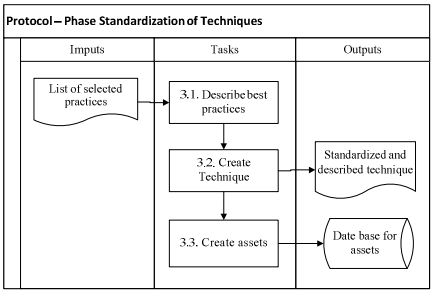
Task 1. Select the process area. This selection is done based on the expert opinion of researchers and project managers.



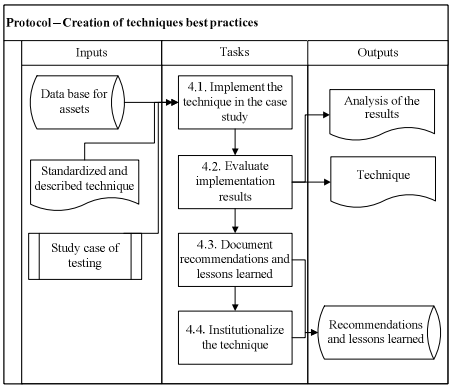
Phase 1. Planning. Establish a plan of study of the process or area of susceptible process to improve for the software development



Phase 2. Selection. Select and prioritize best practices that enable the achievement of the objectives to improve of a defined process



Phase 3. Standardization. Study and analysis of the technical implementation for the selected best practices



Phase 4. Validation. It includes experimentation, evaluation and institutionalization of the process

Fig. 1. Protocol Structure described by phase

Task 2. Describe the process area. It should develop a description of the process area selected, using as basis the reference model. The objective of this task is to fully understand the objectives and scope of the area to identify the presence of it in other models that could eventually be identified by a different name using it by said reference model.

The reference model is selected because of their relevance and prestige among the developer companies and academy.

Task 3. Identify models, policies and standards that incorporate the process area. The final task of this phase is a general review of models, policies and standards that also incorporate the process area of interest. The objective of this task is to select the most suitable models for comparison and selection of best practices.

In Figure 2, the tasks of Phase 1 are shown, with inputs and outputs for each task.

Phase 2. Selection and Prioritization of Best Practice. As input to the second phase of the protocol, you have the list of the models, policies and standards that according to the first overview also incorporate the process area of interest. Using the mentioned inputs, the tasks that make up this phase are:

Task 1. Detailed review of models, policies and standards. Standards and selected models are pre-screened for the selection of those that effectively incorporate best practices to the process area of interest. The criteria for selection or exclusion of some models should be discussed jointly by researchers and project participants.

Examples of criteria are:

- Those models developed by Institutions with recognized status are included, such as the Software Engineering Institute (SEI) from Carnegie Mellon University**,** International Organization for Standardization (ISO), Institute of Electrical and Electronics Engineers (IEEE), the Spanish Association for Standardization and Certification (AENOR), the Colombian Institute of Technical Standards (ICONTEC) and the National Institute for Standardization (INN) among others.
- Those policies or standards that are of interest from companies for certification are included.
- Proposals that include only a subset of best practices in some other way are excluded.

As a result of this task a more restricted list of models, policies and standard that incorporate the process area of interest is obtained.

Task 2. Comparative analysis of selected models. The objective of this analysis is to identify what are the most frequent best practices among the models and how practices that differ between them are complementary. It is suggested that this analysis is developed using as a basis the Method of Similarity Models and Standards (MESME) [3].

As result of this task you have the set of best practices that appear in the analyzed models, i.e. the practices that are repeated and the complementary.

Task 3. Selection of best practices to be implemented. The amount of practice and own practices selected depends on aspects such as available resources, the relationship between best practices, project restrictions, among other aspects to consider.

Phase 3. Formalization of Implementation Technique for the Selected Best Practices. The entrance to this third phase corresponds to the list of best practices that have been selected for implementation. The tasks to be developed in this phase are:

Task 1. Description of best practices selected. The aim is to characterize and understand in depth. It is very important to know what the inputs and expected results for each of the practices are, as well as traceability exists between best practices and techniques or tools for implementation.

Task 2. Creating implementation technique. This task refers to the creation of methods, methodologies, tools, frameworks or other mechanisms to implement and integrate, especially in small companies and best practices easily. With the creation of the implementation technique, it seeks to minimize the costs of incorporating best practices and achieving the objectives of improving the software development process that small organizations proposed.

This task must deliver as results a detailed description of the recommended technique, that is, identify and describe the steps and tasks to be performed, deliver an activity diagram or flow that allows to clearly understand what and how should be done.

Task 3. Create process assets. An active process is a collection of entities used in software process that aid in the development of activities thereof. They are useful in definition, maintaining and institutionalizing of the processes [11]. These assets allows to define standard processes of an organization [12].

Given the above definition, the objective of this task is to identify and design the assets and artifacts to facilitate the implementation and use of the proposed techniques. Between the process assets templates are considered, checklists, forms, among others.

This stage is very important, since for small enterprises, to have a process assets library is a key element that reduces training time and helps guide the approach in the organization [12].

Phase 4. Validation or experimentation. The fourth phase of the protocol corresponds to validation or experimentation of the proposal developed**.** This phase is composed by these tasks:

Task 1. Validation or experimentation. It corresponds to the implementation of a case study. It is important to highlight that this experimental method is empirical, so you can target both the qualitative and quantitative analysis [13].

There are three strategies for case studies [13]:

a) Comparison of results between a new proposal and a baseline.
b) Develop two projects in parallel ("brother projects") by choosing one of them as a base.
c) Apply the new proposal on selected components and compare these results versus those components which were not applied.

The selection of one of the above strategies must be analyzed taking into account the nature of the proposed technique, and the conditions and restrictions of the particular study.

Task 2. Quality assessment. After the application of the technique in the particular case, the quality of products obtained with the application of the new proposal must be evaluated, as well as the quality of the modified process. This quality assessment

in post-mortem form will allow identifying lessons learned and adjustments needed to the proposal.

Task 3. Improvement recommendations. These improvements may be as in the applicability of the same, or the identification of activities/tasks necessary to include or adjust.

Task 4. Institutionalization of the technique. It corresponds to train staff of the organization that will adopt a new proposal and perform an accompaniment in the implementation of the new processes

Next, case study in which it has developed the validation protocol is described.

4 Case Study

The case study aims to identify mechanisms, techniques and methods for the implementation and adaptation of best practices for software development in a multi-model environment. The multi-model environment is a new approach to the process improvement that is characterized because it seeks the integration of multiple models for improving software development processes [14]. This approach seeks to harmonize and standardize best practices proposed by different models [14]. It has been found in this type of environment multi-model a good solution to reduce the complexity of process improvement in small and medium enterprises in Latin America, since it allows selecting and focusing on the best practices for small organizations, considering various models that allows a complement and strengthen of the identified practices.

This case study focuses on the application of each one of the phases of the protocol to identify a set of best practices that enables to improve the process of risk management in a medium sized enterprise.

Each one of the protocol phases, it was used and following results were obtained:

- Selection of a team of experts formed by: a) researchers and teachers involved in the study case, in order to have the academic vision, b) software professionals, to have the current view of the industry needs on process improvement and best practices, and c) a group of graduate students, which provide an updated vision through the constant search for information related to the area and the project context.
- Weighted average of a preliminary list of proposed areas of process such as: Traceability Requirements, Risk Management and Quality Management. This process results in an interest by creating a technique for Risk Management Process, particularly for identification and analysis risk
- Identification of best practices such as: risk classification as the basis of better identification [15], [16] through a taxonomy of risks [17].
- Incorporating fuzzy logic concepts to achieve a risk prioritization and subsequent analysis.
- The design of a method is defined to identify and analyze the risks comprising the following steps:

1. Examine each element of the structure proposed in the taxonomy.
2. Select the set of questions related for each element selected in the previous step, according to the proposal of the questionnaire made by the SEI [13].
3. Perform a plenary session involved in the software development project to identify risks in response to questions related to the selected items [18].
4. Review previous efforts risk management in similar projects, if any, and complement the questions with such review.
5. Examine documents lessons learned and complement the questions with such review.
6. Document identified risks, including the context, conditions and consequences of the risk occurring.
7. Select the items according to the structure proposed in the taxonomy.
8. Planning the next cycle of risk identification from the selected elements in the previous step.

The designed method is based on the following elements:

a) risk taxonomy based on questions proposed by the Software Engineering Institute [17] and,

b) risk analysis matrix based on fuzzy logic. These techniques help to decrease the sense of vagueness in defining the measures in the risk analysis when qualitative values are used.

The mentioned aspects are developed in a computer tool as a way to start the validation process and in order to expedite the identification of risks and the incorporation of fuzzy logic concepts for risk analysis.

Risk assessment is an "assessment" of something hypothetical defined as *risk*, which must then be interpreted as "high", or "low", or "tolerable" [19]. For our proposal we used the linguistic values of the fuzzy logic to risk analysis in a scale of values multiple, as show in the Table 1.

Table 1. Rating scale for risk analysis used fuzzy logic

Linguistic value	Very high	High	Medium	Low	Very Low
Numerical value	100	75	50	25	0

From the implementation of the method in a software tool a software prototype is achieved. It has been designed using the concepts of analysis, design and implementation of software engineering as part of the product development.

Using this functional prototype a pilot validation of the technique starts. The proposed pilot validation aims to evaluate the software tool in its first phase. From this assessment it is pretended to raise experimental project engineering software for validation of identifying and risks analyzing technique.

The results of the evaluation of software tool indicate that 67% of participants in validation previously identify it risks in their projects. The same proportion (67%) indicated that it was easy to identify risks to the applied technique making use of the tool, while 33% considers it was not easy to identify the risks.

In this case study the validation is considered such a limitation for our proposal, therefore, the next step in this research project in progress, is perform formal an experimentation in software engineering with the support of technological tools like the one validated so far.

5 Conclusions

The adoption and adaptation of best practices proposed on international models and standards, is not an easy task to perform, especially for small enterprises that do not have the financial resources or the time required to make a large investment in improving its processes. However, this improvement is a necessity to have organizations to improve product quality, increase productivity, and respond in the best way to the market [12].

With this reality in mind, the joint project arises between the Universidad de Medellin (Colombia), the Universidad Católica del Norte (Chile) and support from the Centro de Investigación en Matemáticas de México (CIMAT-Zacatecas). This project aims to facilitate small organizations, the instantiation of software development best practices in their own processes, thereby enabling increasing maturity and contributing in obtaining higher quality products.

The protocol presented in this paper corresponds to one of the outstanding results of this project. It arose from the need to standardize the labor from geographically distributed teams, and as a tool that is characterized by its ease for the orderly and consistent repetition in different scope or key process areas.

The strengths of the proposed protocol lie precisely in this feature, since the proposal achieves collect the knowledge gained during the execution of the research project on a method to define implementation techniques of best practices. Such techniques support small enterprises to incorporate a set of best practices related to a process area, in an agile way, reducing the effort and time to implement process improvement strategies.

In this paper the application of the protocol in the generation of a technique is presented to support risk management through a software tool that integrates: taxonomy based on questions of SEI [17], b) a method of implementation c) risk analysis based on fuzzy logic.

As future work, it is pretended to continue with the validation and testing of the case study through: a) the implementation of the technique in a computer tool and b) experimentation in software development projects of medium and small organizations. These two aspects will allow us to validate the technique and establish areas for improvement, in order to consolidate a useful asset for the implementation of best practices associated with risk management in software development projects process.

In addition, the consolidation of implementation of this protocol in other process areas will be searched for, such as: traceability requirements and practices for quality management in software development. These process areas have been identified as key to the development of software and are referred to in the research project of which this protocol is derived areas.

References

1. Kitchenham, B.: Procedures for Performing Systematic Reviews. Software Engineering Group Department of Computer Science Keele University (2004)
2. Biolchini, J., et al.: Systematic Review in Software Engineering (2005)
3. Calvo-Manzano, J., et al.: Estudio entre modelos y estándares de buenas prácticas enfocado a las prácticas de planificación de proyectos y utilizando CMMI-DEV v1.2 como referencia. In: 3ª Conferencia Ibérica de Sistemas y Tecnologías de la Información (CISTI 2008), Ourense, España (2008)
4. Wohlin, C., et al.: Experimentation in Software Engineering. An Introduction. Kluwer Academic Publishers (2000)
5. Sommerville, I.: Software Engineering, 9th edn. (2011)
6. Zarour, M., et al.: An investigation into the best practices for the successful design and implementation of lightweight software process assessment methods: A systematic literature review. Journal of Systems and Software **101**, 180–192 (2015)
7. Ng, P.-W., Jacobson, I., McMahon, P.E., Spence, I., Lidman, S.: The Essence of Software Engineering: Applying the SEMAT Kernel, 1st edn. (2013)
8. Real Academia Española de la Lengua (2015, Mayo). http://www.rae.es/consultas-linguisticas
9. Bourque, e.P., Fairley, R.E.: Guide to the Software Engineering Body of Knowledge, Version 3.0. IEEE Computer Society (2014). www.swebok.org
10. Hevner, A., Chatterjee, S.: Design science research in information systems. In: Design Research in Information Systems, vol. 22, pp. 9–22. Springer, US (2010)
11. Calvo-Manzano, J., et al.: A process asset library to support software process improvement in small setting. In: EuroSPI, Dublin, Ireland, pp. 25–35 (2008)
12. Albuquerque, A., Rocha, A.: An approach to evaluate and improve the organizational processes assets: the first experience of use. In: Sobh, T. (ed.) Innovations and Advances in Computer Sciences and Engineering, pp. 279–284. Springer, Netherlands (2010)
13. Wohlin, C., et al.: Empirical research methods in software engineering. In: ESERNET 2001–2003 (2001)
14. Mirna, M., Jezreel, M.: Establishing multi-model environments to improve organizational software processes. In: Rocha, Á., Correia, A.M., Wilson, T., Stroetmann, K.A. (eds.) Advances in Information Systems and Technologies. AISC, vol. 206, pp. 445–454. Springer, Heidelberg (2013)
15. The Standish Group International, "CHAOS Manifesto 2011". The Standish Group International, Inc. (2011)
16. Software Engineering Institute, "CMMI for Development, Version 1.3 CMMI-DEV 1.3" (2010)
17. Carr, M.J., et al.: Taxonomy-Based Risk Identification. Software Engineering Institute, June 1993
18. Kendall, R.P., et al.: A Proposed Taxonomy for Software Development Risks for High-Performance Computing (HPC) Scientific/Engineering Applications. Software Engineering Institute, Carnegie Mellon (2007)
19. Ingle, M.M., Atique, M., Dahad, S.O.: Risk Analysis Usign Fuzzy Logic. International Journal of Advanced Engineering Technology **2 (**2011)