Knowledge-Based System's Modeling for Software Process Model Selection

Abdur Rashid Khan[1],
[1]Institute of Computing & Information Technology,
Gomal University, Dera Ismail Khan, Pakistan
[1]dr.arashid@gu.edu.pk
drarashid.khan09@gmail.com

Zia Ur Rehman[2]
[2,]Institute of Information Technology
Kohat University of Science & Technology (KUST), Pakistan
[2]ziagulzia@gmail.com

Hafeez Ullah Amin[3]
[3,]Institute of Information Technology
Kohat University of Science & Technology (KUST), Pakistan
[3]hafeezullahamin@gmail.com

Abstract—**This paper depicts the knowledge-based system named as ESPMS (Expert System for Process Model Selection) through various models. A questionnaire was developed to identify important parameters, which were evaluated through domain experts in about all the universities of Pakistan. No exact system was found, which could guide Software Engineers for selection of a proper model during software development. This paper shows that how various technologies like Fuzzy Logic, Certainty Factors, and Analytical Hierarchy Process (AHP) can be adopted to develop the Expert System. Priorities assignments to critical factors have been shown for decision making in the model selection for a problem domain. This research work will be beneficial to both students and researchers for integrating Soft Computing Techniques and Software Engineering.**

Keywords- ESPMS, Expert System, Analytical Hierarchy Process, Certainty Factors, Fuzzy Logic, Decision-Making.

I. INTRODUCTION

This article presents a conceptual framework for selection of an appropriate software process model showing the whole work through various models. The main goal of this research work is to guide the Software Engineer for decision making about selection and evaluation of software process model through implementation of Soft Computing Technology. Expert system named as **ESPMS** (Expert System for Process Model Selection) was developed using ESTA (Expert System Shell for Text Animation) as a development tool. .

A software process defines the approach that is taken as software is engineered. But software engineering also encompasses technologies that populate the process – technical method and automated tools [1]. Professional system developers and the customers they serve share a common goal of building information systems that effectively support business process objectives. In order to ensure that cost-effective, quality systems are developed which address an organization's business needs, developers employ some kind of system development process model to direct the project's lifecycle [2].

Software process is a framework to build high quality software [1]. The most difficult task in software engineering is to select an appropriate software process model, which completely suits for the particular situation. If a particular software process model is not selected then it will become a bottleneck for the software product; takes more time, higher budget than the estimated one. Mostly software projects fail due to inappropriate modeling. That is why care must be taken during a selection of software process model. As, most of the times domain experts (software engineers) are few in numbers, are much busy and/or not available in time, so such types of systems are much important to novice users.

This research aims to devise a theoretical framework for software process model selection, which will help Knowledge Engineer and Software Engineer in developing high quality, cost-effective software, well in time within the available resources.

II. STUDY DOMAIN

A. *Software Engineering and Artificial Intelligence*

The integration of matured AI methods and techniques with conventional software engineering remains difficult and poses both implementation problems and conceptual problems [3]. Artificial Intelligence and Software engineering both disciplines have many commonalities. Both deal with modeling real world objects from the real world like business processes, expert knowledge, or process models. Recently several research directions of both disciplines come closer together and are beginning to build new research areas. Some of these research areas are the following; Software Agents play an important role as research objects in Distributed AI (DAI) as well as in agent-oriented software engineering (AOSE). Knowledge-Based Systems (KBS) are being investigated for learning software organizations (LSO) as well as knowledge engineering. Ambient Intelligence (AmI) is a new research area for distributed, non-intrusive, and intelligent software systems both from the direction of how to build these systems as well as how to design the collaboration between ambient systems. Last but not least, Computational Intelligence (CI) plays an important role in research about software analysis or project management as well as knowledge discovery in databases or machine learning [4].

An expert system is a computer program which captures the knowledge of a human expert on a given problem, and uses this knowledge to solve problems in a fashion similar to the expert [5]. Computer programs using Artificial Intelligence (AI) techniques to assist people in solving difficult problems involving knowledge, heuristics, and decision-making are called expert systems, intelligent systems, or smart systems [6]. Software development process is a very complex process that, at present, is primarily a human activity. Programming, in software development, requires the use of different types of knowledge: about the problem domain and the programming domain. It also requires many different steps in combining these types of knowledge into one final solution. One of my key observations is that expert programmers rely heavily on a large body of standard implementation methods and program forms [7]. Software Engineering is a highly dynamic field in terms of research and knowledge, and it depends heavily upon the experience of experts for the development and advancement of its methods, tools, and techniques [4]. Learning is based on knowledge and experiences related to the different processes, products, tools, techniques and methods applied to the software development process. The overall objective of a Learning Software Organization (LSO) is to improve software processes and products according to the strategic goals of the organization [8]. Literature study reveals that there is a great intersection between Software Engineering and Artificial Intelligence. The knowledge of experts of related fields can be captured, elicited and copied into computer to work just like commitment to users.

B. Background and Related Work

Literature study shows that research has been taken on different aspects of software process models but there is no standard criteria being developed for evaluation and selection of software process model. To select an appropriate software process model, which completely suits for a particular situation is very difficult task as much of the problems as well as process models cannot be separated among each other due to their mixed characteristics. Process modeling is a rather young and very active research area during the last few years, new languages and methods have been proposed to describe software processes [9]. Lonchamp [10] focused some framework conceptual issues and terminology of process, such as; framework for Process, Process Models, Meta-process, and process centered software engineering environment. Yu & Mylopoulos, [11] presents a model which captures the intentional structure of a software process and its embedding organization. The model is embedded in the conceptual modeling language Tools. The expert systems could advise the domain engineer in programming without the detailed experience in programming languages, Integrate with the help of deductive database and domain knowledge, the previously developed software components to new complex functionalities [12]. Canfora [13] describe the results and lessons learned in the application of the Framework for the Modeling and Measurement of Software Processes in a software company dedicated to the development and maintenance of software for information systems. LSO is an organization that learns within the domain of software development, evolution and application [8]. Modeling concept is well accepted in software engineering discipline. However, there is still a lacking integration of software process modeling and software process measurement by software engineers. This paper aims to portray the idea and result of integrating measurement in software process modeling [14]. Kim & Gil [15] propose a complementary approach, KANAL (Knowledge Analysis) which helps users and check process models. Liao et al, [16] described an ontology-based approach to express software processes at the conceptual level. Software process is viewed as an important factor to deliver high quality products. Although there have been several Software Process Models proposed, the software processes are still short of formal descriptions.

Literature study reveals, the integration of matured AI methods and techniques with conventional software engineering remains difficult and poses both implementation problems and conceptual problems [3]. Lonchamp [10] focused some framework conceptual issues and terminology of process. Raza [7] stated software development problems includes conceptual specifying, designing, testing the conceptual construct and representation problems that comprising representing software and testing the reliability of a representation. A basic problem of software engineering is the long delay between the requirements specification and the delivery of a product. This long development cycle causes requirements to change before product arrival. Canfora [13] described, modeling and measurement are two key factors to promote continuous process improvement. Turban [17] stated expert system may contain components of knowledge acquisition subsystem, knowledge base, inference engine, user interface, explanation subsystem, and knowledge refinement system. Awad [18] described four components of AI systems: a knowledge base, an inference engine, justifier/scheduler, and user interface. Durkin [5] stated that Expert systems solve problems using a process which is very similar to the methods used by the human expert. Knowledge base, working memory, inference engine, and Explanation Facility are the components of Expert System.

Knowledge base is core component of expert system in AI. Durkin [5] described, the knowledge base contains specialized knowledge on a given subject that makes the human a true expert on the subject. This knowledge is obtained from the human expert and encoded in the knowledge base using one of several knowledge representation techniques. One of the most common techniques used today for representing the knowledge in an expert system is rules. Turban [17] described a system which emulates human intelligence in system by capturing knowledge and expertise from knowledge sources is known as artificial intelligent system. Hence it is the need of the day to develop a Knowledge base System to work just like a consultant for Software Engineers for selection of a proper process model for software development.

III. PROBLEM MODELING

In this research work critical factors were identified to select a process model for a specific problem through a questionnaire, which were verified by domain experts and were analyzed using SPSS. These factors were assigned weights using AHP. Decision making process in the proposed research work has been shown through various models and tables. Following paragraphs describes various models to depict the problem domain:

A. Conceptual Modeling of Prototype ESPMS

This model depicts the whole process of knowledge acquisition through decision making using ESPMS. See Fig 1.

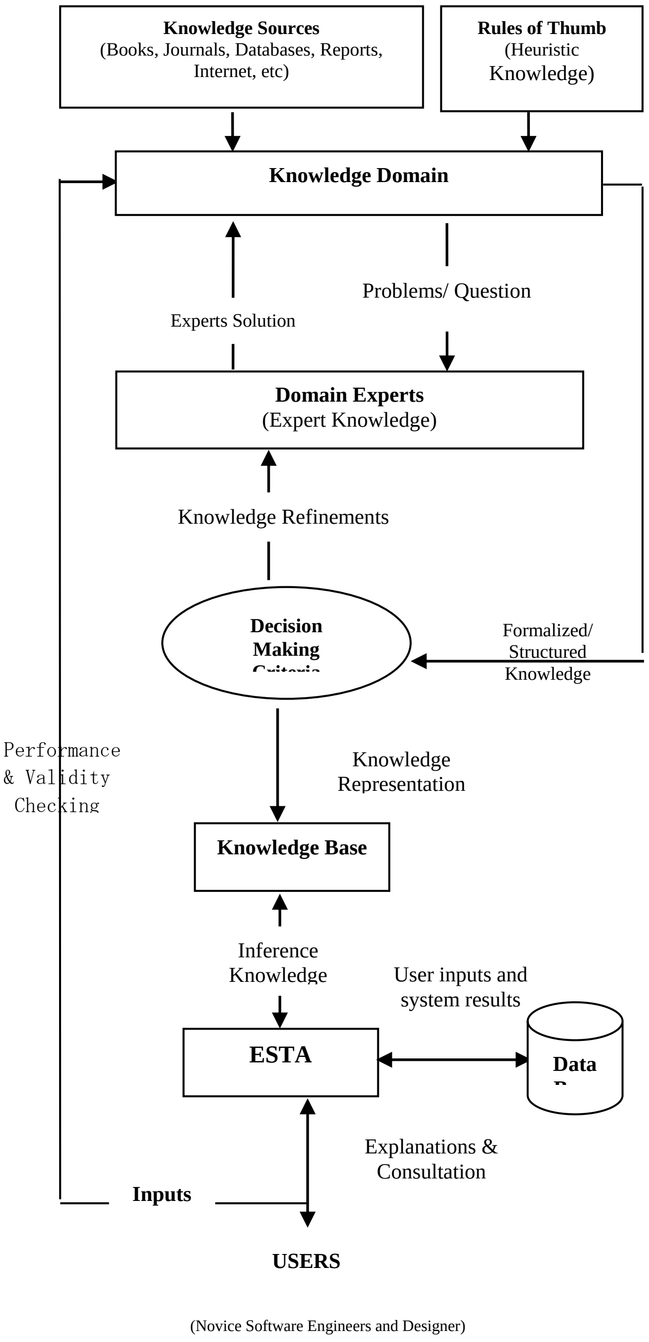

Figure 1. *Conceptual* Model of the ESPMS

B. Decision Making Model

The questionnaire (i.e. knowledge acquisition tool) was sent to about 100 domain experts of 124 universities (both public and private) in the country. Expert opinions were analyzed through using SPSS and weights of the critical factors were known. The analysis resulted that project team, project type & risk management, and Validation & verification were on the top among the parameters [19]. See Figure 2 and Appendix-I for detail.

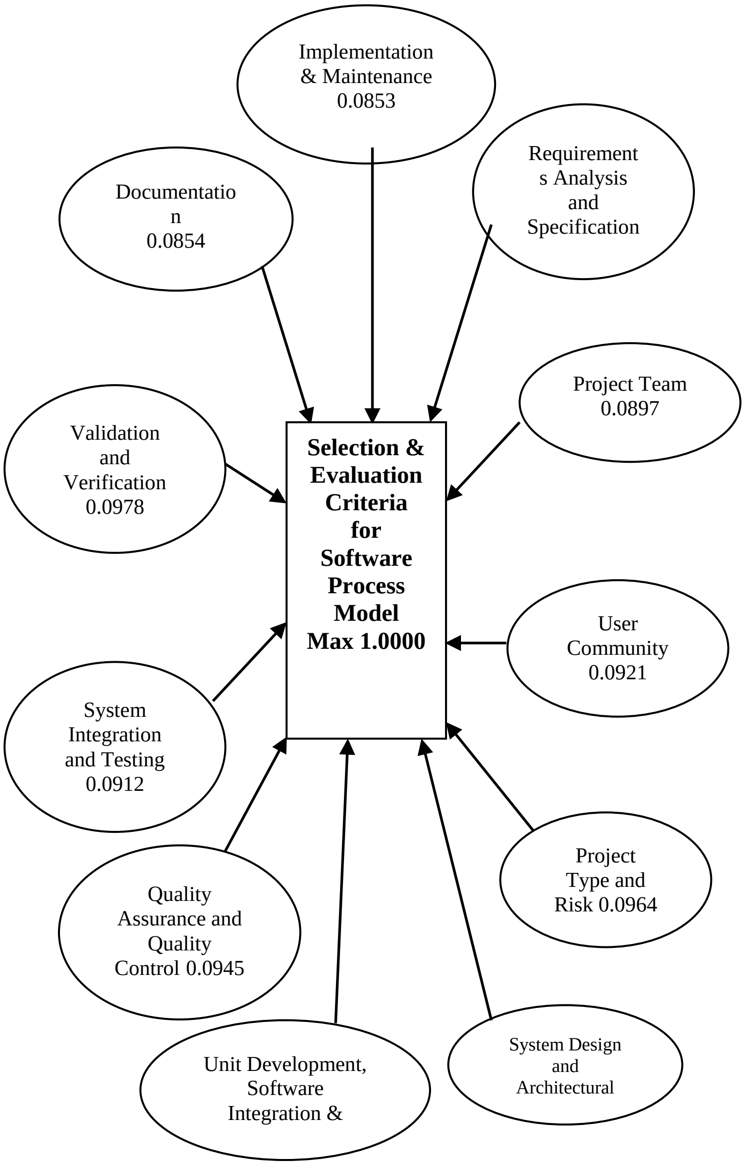
Implementation & Maintenance 0.0853
Documentatio n 0.0854
Requirement s Analysis and Specification
Project Team 0.0897
Validation and Verification 0.0978
Selection & Evaluation Criteria for Software Process Model Max 1.0000
User Community 0.0921
System Integration and Testing 0.0912
Project Type and Risk 0.0964
Quality Assurance and Quality Control 0.0945
Unit Development, Software Integration &
System Design and Architectural

Figure 2. Decision Making by Priorities Assignments to Parameters

C. Expert System Model

Expert System model represents how the Expert System will be developed. Expert Systems are developed either through using expert system languages (i.e. PROLOG, LISP etc) or through using expert system shells (i.e. ESTA, EXSYS etc). We adopted ESTA (Expert System for Text Animation) as development tool for expert system development [20].

ESTA was combined with the knowledge-base to develop the ESPMS, shown as below:

ESPMS = ESTA + Knowledge Base

D. Dialogue Mechanism of ESPMS

ESTA has a special DDE (Dynamic Data Exchange) component, which can share knowledge with the external environment (i.e. other software and databases). Figure 4 represents how Expert System exchanges information with its environment.

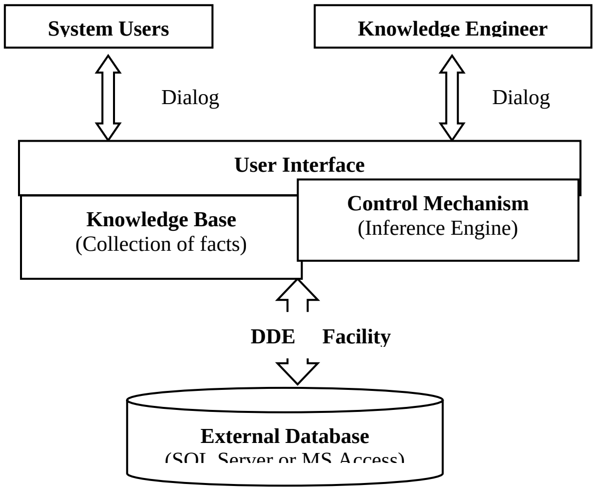

Figure 3: ESTA Dialogue Mechanism of Expert System

E. Production Rules

Rules represent the major elements of a modular knowledge representation scheme, especially when using shells. Rules are conditional statements that are easy to understand and write; they specify an action to be taken if a certain condition is true. They also express relationship between parameters or variables. In expert systems vocabulary, production rules are also called premise-action, cause-effect, hypothesis-action, condition-action, IF…THEN, or IF ….THEN ….ELSE, [18].

The basic idea of using rules is to represent the expert's knowledge in the form of premise-action pairs as follows:

Syntax: IF (premise) THEN (action)

e.g. IF X < 0.20 THEN "Prototype Process Model"

The above example shows that if the value of "X" is less than "0.20" then "Prototype Process Model" will be selected.

F. Symbolic Modeling

For proposed intelligent framework, analytical hierarchy process (AHP) has been used for decision making process in selection and evaluation factors in the software process model. The AHP is a structured technique for handling with complex decision problem, developed by Thomas L. Saaty in 1970s, which is based on Mathematics & Psychology. It provides a framework for solving decision problem and quantifying its elements, for overall goals; also evaluating possible alternative solutions [21].

AHP was used to prioritize the decision making parameters in different levels. The weights of individual factors (i.e. levels) were summed up to level 2 and the weights of level 2 were summed up to get the value of level 1. See Appendix-I for detail.

Maximum weight is 1 and therefore the weight of main goal is 1.000, which is the sum of all the factors weights. To achieve the main goal, first of all we sum up the weights of sub items which become the weight of their groups' parameter, and at the end sum of the weight of groups' parameters become the weight of the goal. The level wise weight assignments to main groups & sub-criteria elements are shown in Appendix-I. To calculate the score of sub-hierarchy the following formula is used.

The following mathematical model evaluates the final value of the objective function. The weight of concerned parameter is multiplied with its assigned weight and summed up together. See equation (1) through equation (4).

$$Kq = \sum_{q=1}^{n} C_q \times W_q \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

Where
Kq = qth main criteria.
n = number of sub-criteria in the qth Criteria.
C_q = Fuzzy value of q^{th} parameter
W_q = Weight of the relative parameter.

To calculate the overall score of the decision hierarchy, the equation is re-defined as:

$$\textit{Total Score} = K_1 + K_2 + K_3 + \ldots + K_{11} \ldots \text{(2)}$$

It implies that:

$$\textit{Total Score} = \sum_{i=1}^{11} Ki \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3)}$$

Where,
Wi= Expert weight of the ith main criteria.
Ki= Calculation obtained from equation (1) or value of main criteria obtained through equation (1).

From equation (1) and (3), we derived equation (4) as below:

$$\textit{Total Score} = \sum_{i=1}^{11} \left(\sum_{q=1}^{n} Cq \times Wq \right) \ldots\ldots\ldots\ldots \text{(4)}$$

Process's evaluation score could not be achieved efficiently through using yes/no, i.e. 1 or 0; as the values of parameters are qualitative in nature. For these qualitative parameters we use fuzzy logic. Zadeh [22] used fuzzy logic to measure the continuous values.

Since, through out in the decision making process parameters' weight assigned from experts' opinion is constant, final decision ranking score computed from the overall decision hierarchy is varied due to the input parameters values entered by users. The value of parameters, might be 0 or 1 [22], if a special parameter is present then the parameter weight is multiplied by 1, else by 0 if not present.

According to Zadeh [22, 24] variables words or sentences as their values is called linguistics variables and the variables that represents the gradual transition from high to low, true to false is called fuzzy variables and a set containing these variables is the fuzzy set. Their degree of membership is [1, 0], where '1' represents highest membership and '0'represents no membership.

We defined a fuzzy variable set for conceptual framework model as:

Fuzzy Set = {Extremely Strong, Strong, Moderate, Weak, Extremely Weak}
Their fuzzy membership values are as: Fuzzy membership value = {1.0, 0.75, 0.5, 0.25, 0}

Table I depicts the fuzzy variables with respective degree of membership value. In the following table, from top to bottom a gradual transition is represented from extremely strong to extremely weak in the fuzzy variables and the respective degree of membership values.

TABLE I. PARAMETERS OF FUZZY VALUES

Fuzzy variable	**Degree of Membership**
Extremely Strong	1.0
Strong	0.75
Moderate	0.50
Weak	0.25
Extremely Weak	0

These fuzzy values are input parameters' values provided by the users during consultation and the final ranking score of a particular process's evaluation is calculated by the system at run time, but here an example of computation in the decision making score is depicted. If qualitative value of a parameter is extremely strong, then related numeric value 1.0 will be multiplied with parameter's weight.

Let, suppose a parameter risk analysis is assigned a weight 0.046 by experts then the fuzzy decision score can be calculated as shown in the Table II.

The overall weights of all the parameters are calculated by Equation (4), and the resultant score will be the final score for decision making.

Decision score calculated by linguistic mapping, which is the output of the intelligent system and also description for a selection of software process model ranking is depicted in the Table III.

IV. CONCLUSION

This research is promising to solve the problems associated with the existing approach to intelligent framework modeling. These models will become a base for selection of an appropriate process model for Expert systems development (i.e. **ESPMS**). Neither there exist any strict rules to be followed to select a software process model nor any consultative system to guide novice user. This an attempt to integrate various technologies, like Expert Systems, AHP, Fuzzy Logic and Decision Making to solve real world problems.

TABLE II. FUZZY SCORE CALCULATION

Parameter's fuzzy value	**Membership value**	*****	**Parameter's weight**	**Fuzzy score**
Extremely Strong	1.0		0.046	0.046
Strong	0.75		0.046	0.034
Moderate	0.5		0.046	0.023
Weak	0.25		0.046	0.011
Extremely Weak	0		0.046	0

TABLE III. LINGUISTIC DESCRIPTION OF PROPOSED SYSTEM WITH OUTPUT (X)

System Output	**Linguistic Description**
X < 0.20	Prototyping Process Model
0.20≤X≤0.40	RAD Process model
0.40≤X≤0.60	Evolutionary Process Models (Incremental, Spiral, Win-Win Spiral & Concurrent Development Model)
0.60≤X≤0.80	Waterfall Process Model
X≥0.80	Component-Based Development

V. FUTURE SCOPE

Following the decision issues and the accompany models presented in this paper, a prototype ESPMS can easily be developed. This prototype ESPMS can be linked with external database and other software to develop a full-fledge Expert System for final decision making in selection of a process model for a particular software project. This work may become a base for solving other similar problems.

REFERENCES

[1] [Pressman R. S., *Software Engineering a Practitioner's Approach,* Fifth Edition, McGraw Hill, 2001.

[2] Reddy A. R. M, Govindarajulu P., Naidu M. *A Process Model for Software Architecture,* IJCSNS, VOL.7 No.4, April 2007..

[3] Jorge L. Díaz-Herrera, *Artificial Intelligence (AI) and Ada: Integrating AI with Mainstream Software Engineering,* 1994.

[4] Rech J., Althoff K. D., *Artificial Intelligence and Software Engineering: Status and Future Trends,* 2004.

[5] Durkin, J., "Application of Expert Systems in the Sciences," OHIO J. SCI. Vol. 90 (5), pp. 171-179,1990

[6] Kazaz, A., Application of an Expert System on the Fracture Mechanics of Concrete. *Artificial Intelligence Review.* 19, 177–190, 2003.

[7] Raza F. N., *Artificial Intelligence Techniques in Software Engineering (AITSE),* IMECS 2009, Vol-1, Hong Kong, 2009.

[8] Ruhe G., *"Learning Software Organizations*", Fraunhofer Institute for Experimental Software Engineering (IESE), Volume 2 , Issue 3-4 (October-November 2000), Pages: 349 – 367, ISSN:1387-3326. 2000

[9] Armenise P., Bandinelli S, Ghezzi C. & Morzenti A, *A Survey and Assessment of Software Process Representation Formalisms*, CEFRIEL Politecnico di Milano, 1993.

[10] Lonchamp J. (1993), *A Structured Conceptual and Terminological Framework for Software Process Engineering,* (CRIN), France, 1993.

[11] Yu E. S. K. & Mylopoulos J., *Understanding "Why" in Software Process Modelling, Analysis, and Design,* Proc. 16th Int. Conf. Software Engineering, 1994.

[12] Grobelny P., *The Expert System Approach in Development of Loosely Coupled Software with Use of Domain Specific Language,* Proceedings of the International Multi-conference on Computer Science and Information Technology, pp. 119 – 123, ISSN 1896-7094, 2008.

[13] Canfora G., Garc´ýa F., Piattini M., Ruiz F. & C. A. Visaggio**,** *Applying a framework for the improvement of software process maturity,* Softw. Pract. Exper. **36**:283–304, 2006.

[14] Atan R., Ghani A. A. A, Selamat M. H., & Mahmod R, *Software Process Modelling using Attribute Grammar*, IJCSNS VOL.7 No.8, 2007.

[15] Kim J., & Gil Y., *Knowledge Analysis on Process Models*, Information Sciences Institute University of Southern California, (IJCAI-2001), Seattle, Washington, USA. 2001

[16] Liao L., Yuzhong Qu Y. & Leung H. K. N., *A Software Process Ontology and Its Application*. 2005.

[17] Turban, E., *Expert Systems and Applied Artificial Intelligence.* New York: Macmillan Publishing Company, 1992.

[18] Awad, E.M. "Building Experts Systems: Principals, Procedures, and Applications," New York: West Publishing Company, 1996.

[19] Abdur Rashid Khan, Zia Ur Rehman. Factors Identification of Software Process Model –Using a Questionnaire, *IJCNS) International Journal of Computer and Network Security, Vol. 2, No. 5, May 2010* 90.

[20] Prolog Development Center, Expert System Shell for Text Animation (ESTA), version 4.5, A/S. H. J. Volst Vej 5A, DK-2605 Broendby Denmark, copy right©1992-1998.

[21] Saaty T. L., *The analytic hierarchy process: planning, priority setting and resource allocation.* New York: McGraw-Hill, 1980.

[22] Zadeh, L.A., Fuzzy sets. *Inform. and control*, 8, 338-353, 1965.

[23] Khan, A.R., Expert System for Investment Analysis of Agro-based Industrial Sector, Bishkek –720001, Kyrgyz Republic, 2005

[24] Zadeh, L.A., The concept of a linguistic variable and its application to appropriate reasoning. *information sciences*, 8,43-80, 1975.

[25] Futrell R. T., Shafer L. I, Shafer D. F, "Quality Software Project Management", Low Price Edition, 2004.

[26] Anderson S., & Felici M., "Requirement Engineering Questionnaire", Version 1.0, Laboratory for Foundation of Computer Science, Edinburgh EH9 3JZ, Scotland, UK 2001.

[27] Skonicki M., "QA/QC Questionnaire for Software Suppliers", January 2006.

[28] Energy U. S. D., "Project Planning Questionnaire", www.cio.energy.gov/Plnquest.pdf (last access, 04 August 2009)

[29] Liu & Perry (2004), "On the Meaning of Software Architecture, Interview Questionnaire, Version 1.2, July, 2004.

AUTHORS PROFILE

Abdur Rashid Khan The author is presently working as an Associate Professor at ICIT, Gomal University D.I.Khan, Pakistan. He received his PhD degree from Kyrgyz Technical University, Kyrgyz Republic in 2004. He has been published more than 23 research papers in national and international journals and conferences. His research interest includes ES, DSS, MIS and Software Engineering.

Zia Ur Rehman The author has received his MCS in Computer Science from Institute of Information Technology, Kohat University of Science & Technology (KUST), Kohat, Pakistan in 2005. He is currently pursuing his MS degree in Computer Science from the same institute. His area of interest includes software engineering, AI, knowledge engineering, expert system, and applications of fuzzy logic.

Hafeez Ullah Amin- is a research student at Institute of Information Technology, Kohat University of Science & Technology, Kohat 26000, KPK, Pakistan. He has completed BS(Hons) in Inforamtion Technology and MS in Computer Science in 2006 & 2009 respectiviely from the above cited institution. His current research interests includes Artificial Intelligence, Information System, and Data Base.