Using Knowledge Base and DSS to Fill the Gaps in Agile methodologies for Managing Project Life Cycles

Lamia Atma Djoudi
Synchrone Technologies
17 Square Edouard VII
Paris, France

Alexandre Monza
Synchrone Technologies
17 Square Edouard VII
Paris, France

Indu M. Anand
Sushila Publications
Chelmsford
Massachusetts, USA

Miguel Rome
Synchrone Technologies
17 Square Edouard VII
Paris, France

Contact: atma.djoudi@ synchrone-technologies.fr

***Abstract*— Current software development methodologies for projects management are based on last methodologies as Agile. In this paper we propose an innovative methodology for project management. This is done by emphasizing quality management, software development, skills, team values, etc. Our vision began by analyzing these keys concepts. Then studying the used methodologies. Proposing precise process and skills are our priority. The uncertain situation are also taking into account in our approach.**

Our vision is to refine the methodology used for the management, find an answer for uncertain situations, to make the right decision for each stage of life cycle of the project. All these all points are the keys of our methodology. This brings us back to the spirit of Decision Support System, which is one of the pillars of this methodology.

With the principles of DSS, we refine the quality of Agile methods. The project may have a sequel or prospects (especially the side research and innovation). The best solutions are offered. Similarity and reusability are important in our methodology. Choose the best skills will be the case study of our approach in this paper.

***Keywords : Decision Support System, Agile,* Intelligent system, Knowledge Base, Evaluation.**

I. Introduction

Today, companies face strong pressures to increase agility and competitiveness. However, the vast majority of strategic initiatives in companies fail, which means that companies are unable to achieve success of their strategy. The main reason for these failures is the lack of consistency between the different components of the companies. At the same time, the need to operate as an integrated whole is increasingly more important. Currently, these challenges are addressed by a dominant functional perspective or management, as recommended by management and organizational sciences.

Several factors are responsible for the growing complexity and difficulty of project and program management at large companies : new and ever-changing technologies, big data, globalization, and company size itself. Added to this mix are the demands of business customers to seek the best solutions for least possible cost and best possible technology. The key inquiries in this landscape are (1) what is the organization of essential work factors? And (2) which methods to choose to plan project management tasks, mobilize the resources, address the risks, etc?

Work organization is a central factor of productivity and efficiency. Proper project management can optimize task scheduling and resource mobilization, and limit the risk from unexpected events. Indeed, managing a project requires planning, budgeting and decision making, the soft skills of persuasion, negotiation, etc. Among the mistakes that may be inadvertently made, with serious consequences during the implementation of a project relate to:

- *Cohesive Team building*, so that the legitimacy of each member on the team is respected and recognized;
- *Motivation for the Duration of the project*, an issue particularly sensitive for long-term projects with several collaborators ; valuing and sustaining collaborative efforts is key to project success ;
- *Team Work*, such that the working group operates with consonance and coordination for the success project.

The vision behind our approach is to optimize a combination of these factors and minimize unforeseen consequences in decision making to achieve the objectives of the project. Keeping in mind that execution of the project is further subject to the constraints including:

-Constraints of time : (1) absolute, external constraint, imposed on everyone, project may die if it is not met ; (2) there may also be penalties for delay ;

-Constraint of cost : profitability constraint

-Constraints of quality: (1) total satisfaction, or (2) partial satisfaction ; (3) and consequences of no satisfaction.

In this paper, we present a new view of the two key points for success in business projects mentioned above: project management and organization of work product within the project / company. With our methodology, it is possible to shape the project lifecycle by choosing the right combination of development activities, as well as proper allocation of effort and resources to better meet the project objectives and the quality assurance goals.

A key element of our proposed approach is to conduct a study after firming up the project specifications. That point in the project life cycle is best to find the right balance between human creativity, which is a projection into the future, and factual data analysis, which is a look-back to the past, which many organizations inevitably attempt through trial and error. At that point in the process, in this way, we

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.190

may also promote curiosity, anticipation of prototyping, providing valuable tips for managers, and be at a beautiful place of learning by error before errors become costly.

Managing a project requires not only coordinated actions but also multiple skills and skill levels, as well as multiple resources, all geared towards the project objectives. In particular, the skills required for effective project management vary by fields, disciplines and levels - sometimes alien to each other. This holds true also for resources. In particular, managing a project requires: (1) An overall structure of the project; (2) task scheduling; and (3) a control resource for costs and deadlines.

Along with the project objectives, there are management elements for decision making to meet the objectives in a timely fashion, keeping in mind: (1) "*innovation*" because of today's pace of advances in technology, quick obsolescence and customer expectations ; (2) "*competition*" because market requires innovation, but fast ; (3) "*quality*" consciousness because customers expect to meet their needs precisely.

The discussion of current development of project management methods as Agile illustrates the importance of the concept of user understandability, taking into account the uncertain situation to achieve the goals fixed for a project. Also to satisfy the customers.

However it is not enough to study the technical approaches of these methods, we need Decision Support System (DSS) field to expand the boundaries of solutions. To give an example, we previewed a case study about Agile methods for project management. We describe the process proposed by Agile of an ongoing project. Then we present our methodology to give more granularity for Agile methods.

This paper is structured as follow: Section 2 presents an overview for methodologies of project management. Then, section 3 presents the issues and objectives. The Case study is described in section 4. Related work is presented in section 5. Section 6 describes conclusion and perspectives.

II. METHODOLOGIES OF PROJECT MANAGEMENT

Many different software paradigms and tools for the study and management of projects are used. Notable among these models are:

- **The "cascade" model** [1]**.** This control program is presented in series of steps. Execution of each phase ends on a *specific* date. Even if the date is extended with a possibility of a return, it is ideally limited only to the phase that precedes it and is called into question. The development is linear. This model may present a problem of "tunneling" of the view, and failing to take into account users' evaluation during development.
- **The "V" model [2].** This flexible approach is the most commonly used method. In the left, downward-sloping branch of the V, development of business requirements, application design parameters and design processes are defined, and in the right, upward-sloping branch of the V, testing and debugging is done, with unit testing followed by bottom-up integration testing. The code is written at the base point of the V, and the extreme upper right point of the V represents product release and ongoing support.
- **Agile model [4].** To overcome the limitations of the methods listed above, several changes were made in the methods, leading to what is known as Agile methods.

These methodologies face head on the knowledge management activities and the rate of unpredictable changes in software projects. Agile methods [5] emphasize people, communities of practice, communication and collaboration to facilitate the practice of sharing tacit knowledge in the team. Agile processes use feedback, rather than planning as their primary control mechanism. The feedback is driven by *regular* testing and scalable versions of software, and these phases are simplified in order to shorten the duration. The developers and users may be called upon to express or answer questions related to the project. These methods rely heavily on socialization through communication and collaboration to access and share tacit knowledge within the project team.

Our own experience at a company and in collaboration with various clients / projects, confirms increasing use of Agile method. Its main advantages are: (1) to allow the minimization of the differences between the developed and produced business needs ; (2) to lead to a cost reduction study which is conducted in conjunction with the development ; (3) to limit the possibility of tunneling for users and management involved in the project ; and (4) to permit great flexibility and responsiveness for short, high-value-added projects. Over the past few years and despite the wide use of such methods by us and our customers, we have found some limitations:

- Availability, which is more difficult for small teams, including the drafting of project documentation;
- A lack of forward planning as the project progresses;
- A need for consultation/ interaction with other projects/teams because the fact is that not all employees may assemble teams receptive to change.

These limitations may prevent effective application of any Agile project. For example in projects where the decisions can rapidly change, or a problem is unstructured or the circumstances are uncertain. To handle such problems, other methodologies are used. An important example is: Decision Support system (DSS) [6]. It is specifically designed to facilitate the decision making process in the operations of organizations. DSS is used for management and planning and to help with decision making about issues that may be rapidly changing and not easily specified in advance, e.g., semi-structured and unstructured decision problems. A DSS may be entirely automated, human-powered or a combination of both.

III. APPROACH

In this paper, we present a cooperative way the two methodologies can be used: DSS, to solve problems that are ready to be scientifically modeled; and Agile, to plan and specify the details of a project. Generally DSS includes a good knowledge base (KB), which makes its use reliable. Our Knowledge Base System (KBS) is useful for management of projects, including follow-up, as well as for creating and maintaining profiles of the team members and statistics for projects, customers, and so forth.

Where the goal is to apply the principles of similarity and reusability, and to have precise answer in a short time and to provide for the continuation of sub-projects carried out, e.g., side research, the DSS answers our requirements perfectly. Our approach introduces important parameters that allow making the right decision for a project.

We start by launching our processes to: (1) Understand needs; (2) Satisfy customer; (3) Respond efficiently and at lower cost; and (4)project into future customer needs.

Also, since the workers' competence is important in any project, we address the *skills' need*s, in addition.

We begin by observing that there is a wealth of information that has to be managed in any project, and that it can be generated by different information systems, different teams, etc., and may be shared by several people / teams. In these situations making a decision about internal / external task is not always obvious. Further, the decisions may be structured and unstructured, wherein:

- Structured/programmed Decisions: (1)Decisions made repeatedly; (2)Decisions following some guidelines/rules.
- Unstructured/Unscheduled Decisions (e.g. investing in new technologies) are: (1) New decisions; (2) Have no rules prescribed to follow; (3) Made on the basis of available information; (4) Based on discretion, instinct.

In practice, since many actors are directly involved in project management, a decision regardless of its size may be difficult or have significant consequences. On the other hand, there may be risk of "over-specification" spawning endless sub-projects and therefore expensive to implement.

Agile methods are based on iterative and adaptive development cycles according to the customer's changing needs. They allow and tend to involve all employees and customers in the development of the project. In Agile development, information is regularly exchanged to improve the product. But what if a process / task suffers from an inherent lack of precision? For example, the skill sets for task management.

In order to make the right decision during each step of the project relevant information, documents, personal knowledge, and / or models to identify and solve problems are really necessary. We consolidate this information and refine the Agile Vision to help make right decisions.

Some of the criteria applied for this vision are: (1) Adaptability and flexibility; (2) high level of interactivity; (3) ease of use; (4) efficiency and effectiveness, and (5) extensibility. In addition, for our platform we also want:

-Support/assistance to managers at different management levels, from senior executive to line managers.

-Support for Individuals/groups. Less structured problems often require the participation of several individuals from different departments and levels of organization.

-Supporting interdependent or sequential decisions.

- Support for the intelligence, design, implementation,etc.
- Support for the variety of styles and decision process.

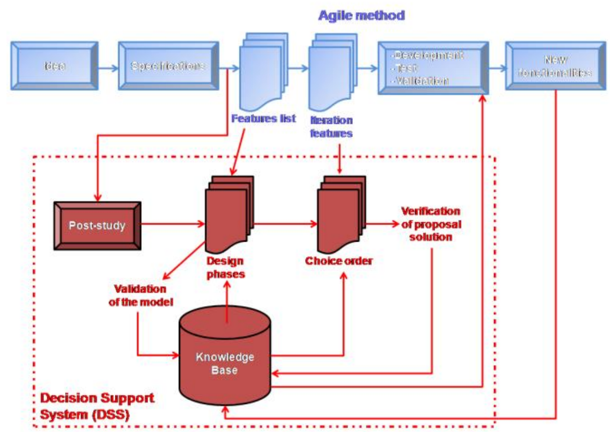

Figure 1: Agile-DSS Approach

Our goals are: (1) Improving the efficiency and speed of decision making activities. (2)Increasing control, competitiveness and predictive decision-making ability of the organization. (3)Facilitating interpersonal communication. (4) - Encouraging learning or training.

In short, with our philosophy, we want to imbue Agile methods with the spirit of a Decision Support System.

One of the highlights of DSS is the ability to gather information from internal sources of an organization in which there is a certain amount of control, and external sources over which control is more limited. Internal sources consist of all the different databases within an organization, such as those related to personnel, finance, assets, customers, etc. External sources are composed of various factors that influence how an organization operates, such as market trends, government regulations, competitors, etc.

The DSS development process must assure cooperation among key stakeholders. The three main types of actors that must be involved in the development process are: end users, experts and knowledge engineers. [7], [8]. To this mix we add the developer as a fourth actor. The goal is to help design a system that the user will see as a true partner, and be actively involved in the exchange of information through interaction and collaboration. The knowledge engineer interacts directly with the expert and the end user to gain domain knowledge throughout the development process.

In addition, we also want the system to be adaptive and scalable. The adaptation must be applied in an intelligent manner based on the principles of artificial intelligence:

Case Based reasoning (CBR), Expert system (ES) and Machine Learning (ML). And go beyond client-side and system analysis. All this data, along with the history of changes is saved in the knowledge base for use later in the project or in other projects. Thus we maintain the learning in DSS and granularity in Agile. The overall design is presented in Figure 1.

Project management using the Agile methods starts with setting the requirements, needs analysis and breaking down of tasks into project features, followed by classification of the features by prioritization. Testing and validation are required for each feature. A new list of features is made respecting the extent and the validation and deployment of complete project (see the top line of Figure 1).

We propose a post-study of customer needs following the specification step but before embarking on processing using the feature lists etc. (Though in fashion, it can be exploited or misused, and fail or leave a customer dis-satisfied).
With this post-study, we propose a new design features. The goal is to have fewer features in parallel with realization.

This recommendation is based on the experience of our experts, the exchange with MOA (project owner), as well as the application of the principle of similarity and reusability to save time and also enrich the features of the project.

A project manager's validation is needed to take the next step in our approach. This step is proposed/ prioritized by platform, skills, principles of similarity and reusability and also the independence of the spots.

A check is also needed to validate our proposal. During this phase, the outlook can be identified. The prospects can be offered to the client for a possible continuation of the project. Also, there will be a return to our knowledge base and to enrich for possible uses.

An implementation report is also saved in our KB. This allows us to provide effective and timely solutions for similar projects. Statistics of market studies, launching of research projects and innovations occur in this phase.

A cooperative DSS allows the decision maker (or its advisor) to modify, supplement or refine the proposals for decisions under the system, before sending them to the validation system. The system improves them over further, and refines the suggestions of the decision maker and refers them for validation. The entire process then begins again, until a consolidated, proposed solution is generated.

The main focus of our vision is to group all the information in a framework to facilitate communication and make the right decision. The key idea is to centralize and automate the organization, collection and processing of monitoring data in a comprehensive way. Our system will be capable of storing data collected by the continuous and periodic monitoring and the results of model calculations. Thus, it improves the analysis process carried out by the staff of a single project or external users.

IV. CASE STUDY

For the successful implementation of any project, the skills of the team members are one of the most important factors. Hence, we examine as a case study the "selection the best skills for a project". We consider as example a project for about 25 months where the client/user identified the need for skills in C, Java and Python. In the normal process, the client can have these skills within its teams. Otherwise it must launch a tender at nearby recruitment companies. Generally, it requires years of experience to find the right candidates. We propose the following scenario:

Scenario 1: The customer needs 3 persons: C with 4 years of experience, Java with 2 years' experience and python with 7 years' experience. Tasks are separated and do not overlap. In our process, we proceed by the following steps:

1- A post-study is launched in order to have information on the skills required. We assume that for the three languages, we shall have: C language with evaluation 13, Java with an evaluation of 6, and Python with an assessment of 16.

Note: this evaluation process [19] is an internal application in our business. It is under development based methods of processing decision trees and branches. Figure2 presents the customer needs. Once we add our assessment, we obtain the 3D presentation (diagram 2 in Figure 2).

2- Once the skills are evaluated, we launch the request in KB or in job board to find the best candidates. The skills of each candidate are presented as cuboid in our KB, that has as axes: skills, experience and evaluation. The global cuboid must encompass all the skills of the candidate.
Table 1 summarizes the skills, experiences and evaluations for the candidate 3 and 4.

3- The choice of the best candidates depends on other criteria (uncertainties). In our case we have the candidate and the candidate number 3 and number 4. The decision needs the best candidates by the principles of DSS and we discuss the cost, years of experience with the customer.

4- All data, results of the project and the profile of the candidates should be back in our knowledge base. This allows us to make a good and accurate decision for a possible project. To enrich our KBS is also one of our priorities. The process of learning is initiated when new data are acquired. A call for experts is launched if necessary

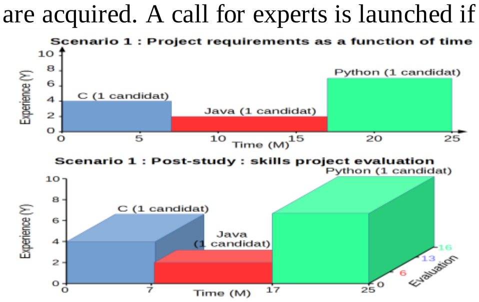

Figure 2: Post-study and skills evaluation

	Skills	Experiences	Evaluation
Candidate 3	C #	2	4
	C	4	15
	Java	3	7
	Python	6	15
	SQL	12	7
Candidate 4	C#	0	0
	C	5	16
	Java	4	11
	Python	8	14
	SQL	4	9

Table 1: Example of evaluation skills

Figure 3 presents the skills of two candidates (3 and 4). By a projection, we can identify the best candidates. In our example and for C skills, for example we have 5 candidates.

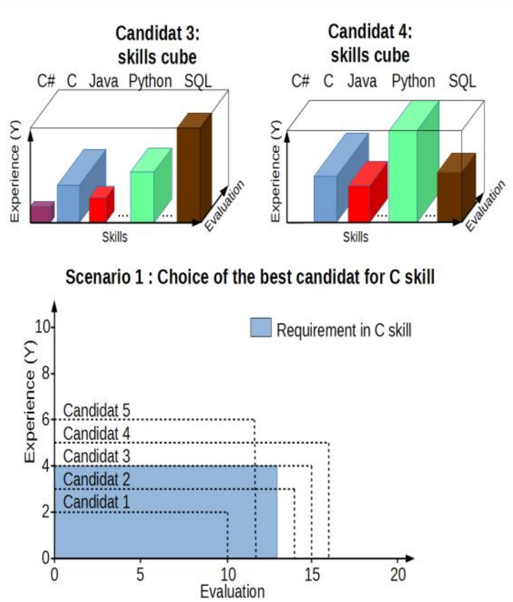

Figure 3: Cubes of skills and C skill projection

V. FURTHER ISSUES AND RELATED WORK

The Agile methods evolved in the mid-1990s. The various Agile Methodologies differ in the approaches to software development and management they propose [9]. They focus on people more than on technologies [10]. The project owner is the main actor in Agile methods[11]. Also, their focus, more on planning than technology, and some other details makes them less suitable Agile methods [12].

With many of these Agile methods, people focus on communication, flexibility and on improvement during and after product development [13].

Some agile methods focus more extensively on project management and collaboration practices such as Scrum, Adaptive Software Development (ASD), and Lean Development. However, some agile methods concentrate heavily on software implementation practices such as Extreme Programming (XP), Agile Modeling (AM), and Feature-driven Development (FDD) [14] focuses only on several specific techniques, i.e., Scrum, Extreme Programming, and Adaptive Software Development. Applying Agile methods is not easy for team members [14].

There is a negative correlation between the size of the organization and the successful implementation of the Agile model, thus the larger the organization the harder it is to employ agile methods [15].Agile model works best for small teams as in large teams. That can have to be maintained can reduce the effectiveness of practices such as informal face-to-face communications and review meetings [12].

Our Knowledge Based System (KBS) can help us to apply the reusability and similarity process in order to benefit from the old success project and to improve Agile methods. With our approach and based on the spirit of our KBS and DSS, we find that the bigger the project, the more efficient are the results than using purely agile methods.

DSS is designed to support users in solving various management issues, with the objective of improving the decision process. [6]. The process of designing and implementing DSS follows several stages. The first stage corresponds to the initiation and the conception of the DSS based on the structure of the decision process and on the needs of the decision makers [16]. Actually, many industrial application of DSS are from vendors' web sites and a new generation of systems have emerged, namely the business intelligent applications.DSS is one of pillars of project management. We agree with by Nemati et al. [17] in their assertion that a good research area of DSS becomes the development of a set of theoretical foundations upon which to build future development and applications.

Enabling good management in a project realization depends on identifying best skills, best tools, etc. Optimization and improving the used methodology is also more interesting.

Current software development practices need to bring up to date methodologies such as Agile. A software development methodology should include the totality of skills, teams, tools, processes, activities, standards, team values, etc. [18]. In Agile methods, we find that the exchange with project manager, and emphasis on skills enjoys the success of similar projects, and helps to take the best choice.

Using DSS in Agile methods is also beneficial for project management. In a DSS project, we can ask for lighter weight, faster and more agile methodologies that can accommodate the inevitable ongoing changes to requirements. We propose a software methodology that can cope with knowledge management activities, along with a rate of unpredictable change in software projects. Nadeer[8] develops a new framework for software methodology based on DSS and Agile methods. He is focused on XP methodology to develop DSS. Our methodology is more general and takes into account more Parameters.

We use the skills as a case study. The evaluation of skills is an important axis in research. In this section we present an overview of the important works and present our opinion. In our research, we enrich this topic by adding the evaluation skills algorithm [19]. In [21], authors propose an expert system based on fuzzy logic and rough set theory and semantic technology. This system is beneficial for project manager working on Scrum Agile method. The main limitation, it cannot help us to choose the best skills.

Using the other variations of Agile method are similarly unhelpful, since they cannot be applied for a project taking into account the previous success or the knowledge base.
For example, the proposers in [22] are interested in minimizing costs by optimizing the allocation of tasks to the players with the required skills. They use the Algorithm of Ants Colony (AAC), and the mathematical modeling of a multi-period multi-system projects in order to propose a heuristic for (AAC) with the simplex (two methods for improvement). Azam et al. [23] proposes a recommender system to help for the best choice.

Consider cost optimization as an example of the flexibility of our approach, since it is very important to complete our evaluation skills. We simply add another new axis in our cube of evaluation for skills presentation. So we are not limited to specific parameters. It is a flexible idea, such that as we add more parameters, we add more precision to the introduction of skill sets.

VI. CONCLUSION

With the increasing complexity of project and tight coupling several technologies, decision making in a project is becoming increasingly more difficult, with the related need to achieve the best solution with lost cost and risk.
Our approach optimizes the main activities of Agile methodologies and supports the decision making process at management level. In addition the DSS enables the user to perform sensitivity analysis based on ~~varying~~ various input parameters. Based on similarity, we can present also a precise and quick decision. In this paper, we presented a new approach to refine the Agile method to do well managing a project. An agile decision support system for project management is proposed integrating and applying, reusability, learning, similarity and KBS.

Our approach provides a rich set of information for decision makers, which can offer an important objective reference to effectively improve the decision accuracy. Also, we can have a vision of the perspective of project analyzed by our approach. We can propose a good management process, good platform, precise skills, and why not a good vision on the future of the project.
Currently we are studying the skills of our business process database. We have over 1,000 engineers placed with more than 100 customers (Banks, Orange, France Telecom, ...).
One of our perspectives for this approach is also the performance study in project management methods especially Agile, and compare it with the performance of our results, with addition of the parallelisms spots in project management to achieve the best performance. The design of an Intelligent DSS will add more value to our approach, and is one of the priorities in our company and our customers.

References

[1] D. Marks, “Development Methodologies Compared”, N CYCLES software solutions, December 2002 , www.ncycles.com , 2/2/2005
[2] Macias, F., Holcombe, M., & Gheorghe, M. (2003). A formal experiment comparing extreme programming with traditional software construction. *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03)*
[3] B. Boehm, “Get Ready for Agile Methods, with Care,” IEEE Software Development, January 2002, pp.64-69
[4] Beck, K. et. al. (2001). “The Agile Manifesto”, Retrieved from: www.agilemanifesto.org
[5] M. Fowler, “The New Methodology,” http://www.martinfowler.com/articles/newMethodology.html Accessed on 12/12/2004
[6] Power, D.J. “A Brief History of Decision Support Systems”. DSSResources.COM, World Wide Web, http://DSSResources.COM/history/dsshistory.html, version 4.0, March 10, 2007.
[7] P. Zarate and C. Rosenthal-Sabroux, “a cooperative approach for intelligent decision support systems” in Proceedings of the Thirty-First Hawaii International Conference (HICSS.1998) Volume: 5, On page(s): 72-81 vol.5 , 1998.
[8] Natheer K Garaibeh DSS development and Agile methods: Towards a new framework for software development methodology, International Journal of Machine Learning and Computing (IJMLC) 01/2012; vol. 2(no. 4):438-442
[9] http://martinfowler.com/articles/newMethodology.html
[10] D. Duka, “Adoption of agile methodology in software development”, Proceedings of the 36th International Convention on Information & Communication Technology Electronics & Microelectronics, 20-24 May 2013, Opatija, Croatia, Publisher: IEEE, ISBN: 978-953-233-076-2, pp. 426-430.
[11] T. J. Lehman and A. Sharma, “Software Development as a Service: Agile Experiences”, Proceedings of the 2011 Annual SRII Global Conference, 29 Mar. - 2 Apr. 2011, San Jose, USA, Publisher: IEEE, doi: 10.1109/SRII.2011.82, pp. 749-758
[12] M. R. J. Qureshi, “Agile software development methodology for medium and large projects”, IET Software, vol.6, no.4, pp.358-363, doi: 10.1049/ietsen.2011.0110
[13] Qumer.,&Sellers (2007).*An evaluation of the degree of agility in six agile methods and its implacability for method engineering. Information and Software Technology 50 (2008)*
[14] umsl.edu/~sauterv/analysis/6840_f09_papers/Nat/Agile.html
[15] J. A. Livermore, “Factors that impact implementing an agile software development methodology”, Proceedings of the 2007 IEEE SoutheastCon, 22-25 March 2007, Richmond, USA, Publisher: IEEE, doi: 10.1109/SECON.2007.342860, pp.82-86
[16] Turban, Aronson, and Liang, Decision Support Systems and Intelligent Systems, Seventh Edition, Prentice Hall, 2005.
[17] Nemati, H., Steiger, D. , Iyer ,L., Herschel, R, "Knowledge warehouse: an architectural integration of knowledge management decision support, artificial intelligence and data warehousing", Journal of Decision Support Systems 33, , 2002,43– 161.
[18] A. Cockburn, “Selecting a project's methodology”, IEEE Software, vol.17, no.4, pp. 64-71, doi: 10.1109/52.854070.
[19] Submitted
[20] ReSySTER : A hybrid recommender system for scrum team roles based on fuzzy and rough
[21] set R. Colomo-Palaxios, I. González-Carrasco, J. L. López-Cuadrado, Á. García-Crespo, Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 4, 801-816
[22] M. Fikri, A. E. H. Alaoui, M. E. Khomssi, Assignment Staff with Dynamic Competencies in multi-projects & multi periods : modelling and solving by a hybridization of ant colony algorithm, IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.3, March 2011.
[23] N. Azam, J. Yao Game-theoretic rough sets for recommender systems Knowledge-Based Systems 72 (2014) 96-107