Monolithic Ontological Methodology (MOM): An Effective Software Project Management Approach

DOI : 10.36909/jer.10679

Kamal Uddin Sarker[*], Aziz Deraman[*], Raza Hasan[**], Ali Abbas[**], Babar Shah[***], Abrar Ullah[****]

[*]University Malaysia Terengganu; Malaysia. **Middle East College, Oman. [***]Zayed University, UAE. [****]Heriot-Watt University, UAE.

*****Correspondence: ku_sarker@yahoo.com, ksarker@mec.edu.om (Kamal Uddin Sarker)

ABSTRACT

Due to rapid changes in software applications, especially incorporating the demands of self-regulating technologies becomes a major challenge in software projects. This research focuses on technological, managerial, and procedural challenges, which are believed to the most significant factors contributing to projects failure. To address these issues, this study proposes Monolithic Ontological Methodology (MOM) which addresses the weakness in the existing benchmark methodologies including PRINCE2, Extreme Programming, and Scrum in terms of project management, quality control, and stakeholder involvement. The MOM consists of seven phases and each phase has the required number of iterations until it is approved by management. The updated information is recorded and shared with the respective teams. The standard documentation with control language is structured by Descriptive Logic (DL) that reduces ambiguity and technical debate. Furthermore, the illustration of the MOM includes figures, logical expressions, and descriptions. To demonstrate the effectiveness of the proposed methodology, an Analytic Hierarchy Process (AHP) was performed. The findings indicate the validity of MOM concerning considered performance metrics. Although the applicability of the proposed methodology involves relatively more documentation and formalities. The adaptive nature of MOM makes it suitable for the standard organization and brings sustainability to the organization by implementing distributed project management.

Keywords: Software project management; software quality control; methodology; sustainability; ontology.

INTRODUCTION

According to the Standish Group Report (S. Hastie, & S. Wojewoda, 2015) every year a significant number of IT project fails partially or completely for reasons related to people, task, process and/or environment (Timo O et al, 2014); there is no single reason to point out for project failure (N. Cerpa & J. M. Verner, 2009) as there has been a causal relationship among the above-mentioned reasons. The successful delivery of a project depends upon the understanding of tasks and adopting an appropriate process that maps to the skills, experience, and relationship of team members. Incorrect or inefficient resource allocation is one of the key causes of project failure (Al-Ahmad W et al, 2009). Some research studies emphasize the understanding of requirements, realistic plan, appropriate methodology, accurate design, the effectiveness of implementation, adequate testing and correct resource allocation (time, cost, technical and human); may contribute to a successful project. Noor Habibah Arshad et al (2007). suggested the organizational structure and policy; whereas McLeod et al (2011) argued that knowledge of the business domain and motivation of the team contribute to a

successful project. It is essential to perform risk management (Haneen Hijazi et al, 2014), changes management, and resolve technical debates by management. Most of the methodologies describe a workflow, phases, processes, and internal sequence with one or more iterations. The success rate of a project primarily depends on the performance of the management team. Moreover, the management team should enhance sustainability by reusing design and code, effectively employ internal tools, and implement in-house environmental sustainability practices. There are plenty of methodologies and selection of a methodology is difficult due to the criteria: size, risk, stakeholders' demand, and complexity of the project. "Each methodology is unique" and "each project is unique" so appropriate methodology selection is a complex work too. Though the best practice is continuing in the software firms, till now more than 20% can not be completed and around 50% of the project delivered without user satisfaction (S. Hastie, & S. Wojewoda, 2015) and this research has proposed MOM to ensure effective project management approach that can improve quality of the product too.

LITERATURE REVIEW

An ontology relates to the existence, reality, and the categories of being and their relationship (Gruber T, 1995). The philosophical ontology was introduced in computer science (Gruber T, 1995) to specify features of domain knowledge and information sharing (Neches R et al, 1991). Sooner it was adopted by computer scientists in the application area of artificial intelligence (Guarino N, 1998). It became popular in the fields of knowledge management systems (Lai L.F, 2007) and a vast amount of ontological application is found in web science i.e. semantic web and e-commerce (Fensel Dieter, 2004). Standard research platform creation is one of the

important contributions of ontology. The use of ontological approach improved research on genome data format specification (Ashburner M et al, 2000), vocabulary specification for agricultural information (Clément Jonquet et al, 2018), and data standardization in health informatics (Sunitha A. & Suresh B. G, 2014; David Riaño et 1l, 2012). The semantic web (Gómez-Pérez A. & Corcho O, 2002) is an active research area and popular for the use of ontology in computer science which provides a basis for the semantic web. It also applied for information specification, clustering, and object-oriented structuring, or class hierarchy development. There is a wider acceptance of this approach in computer science and therefore, W3C (Heiyanthuduwage S.R et al, 2016) standardized the ontology in the semantic web. Information visualization ontology gathers information from multiple sources to display in a single unit (Fluit C et al, 2006). Ontology is used for requirement specification (Kamal Uddin Sarker et al, 2017) and quality factors specification (K. U. Sarker et al, 2018) by descriptive logics in software project management. It emphasizes sustainability practice (Kamal Uddin Sarker et al, 2018) into an IT project by improving the quality of the process and product, reducing execution time and cost (V.K. Chawlaa et al, 2018). Practicing project management is being momentous with explicit specified information of ontology. Recent studies have shown that ontology modeling could be effective for smart application (K. U. Sarker et al, 2019) and big data management (Sarker Kamal Uddin et al, 2019).

A typical methodology is a collection of sequential and simultaneous actions. The methodology selection depends on user requirements, project size, degree of risk, and nature of the project. While some methodologies are good for high-risk projects but not appropriate for small

projects. The waterfall model is a standard one among plan-driven approaches, which is the foundation of many methodologies today, however, it overlooks risk management. Similarly, XP is an agile approach, which is appropriate for small or medium-sized projects (K. Schwalbe, 2009). Rapid Application Development (RAD) is an incremental model, which is considered appropriate for low-risk projects (R. S. Pressman, 2005), Scrum is a popular agile method for large object-oriented projects while the Spiral model is good for high risk and large projects (K. Schwalbe, 2009; R. S. Pressman, 2005). Agile methodologies are flexible towards requirements changes and focused on customer satisfaction and teamwork (L. Williams, 2010). However, Scrum and XP lack documentation practice, unstructured managerial functions, and no support for distributed projects (Faiza Anwer et al, 2017). Moreover, agile does not consist of the functionalities of a virtual project management approach. PRINCE2 supports a virtual project management approach that is popular in the UK and currently used in more than 50 countries (Radka Vaníčková. 2017). But it takes longer decision-making time and less or no importance is given to human management. It does not fully support the change management process and output prediction before completion. Distributed project management allows working from any corner of the world in a software project that is economical for software industries and scope for experts. But existed methodologies not enough to manage distributed project management.

RESEARCH METHODOLOGY

The research has proposed a methodology that is elaborated with diagrams, ontology, and DL in the section below. It is comparatively massive in size and decomposed into multiple lattice

forms as well as an explicit presentation with DL. It is called "monolithic" due to the explicit specification of the project information (Kamal Uddin Sarker et al, 2020a) is presented by a rigid structure of the methodology. The concept of ontology is used to present project information in a formal structure that can improve sustainability by reusing, sharing, and re-constructing (Kamal Uddin Sarker et al, 2020b). MOM aims to address current issues relating to the managerial process, documentation, sustainability, and way of control. The monolithic methodology is decomposed into seven lattice ontologies and each lattice ontology reflects the respective phase that is demonstrated by diagrams, logical expressions, and explanations. MOM is proposed to allow the virtual management system to improve business goals. Gray Rational Analysis (GRA) and Analytical Hierarchy Process (AHP) are the two most popular methods in Multi-Criter a Decision Making (MCDM) where GRA is used for group comparison while AHP is used for pair comparison (Sarker KU et al, 2020). The process of analysis includes pair comparison in five factors: involvement of management, standard documentation, sharing environment, engagement of stakeholders, and consideration of software quality factors. The analysis process is synchronized with reciprocal matrix, normalized matrix, and priority values based on the average score of the expert. A better score is found for MOM than XP, scrum, and PRINCE2 that is visualized in a graph. The work is concluded with future work that recommends the importance of a virtual ontology.

MONOLITHIC ONTOLOGICAL METHODOLOGY (MOM)

MOM (Figure 1) consists of seven sequential steps and each one is directly controlled by management. It will allow the required number of management interactions within a task in

any phase. The management team will be formed with a hierarchy and they are directly connected to a phase for providing instant feedback. For additional feedback, management is responsible to communicate with respective stakeholders. Updated information will be recorded and shared with respective phases and stakeholders. This practice will provide standard documentation and formal management. Each inclusive lattice ontology is decomposed from the monolithic (Figure 1) with the role of stakeholders.

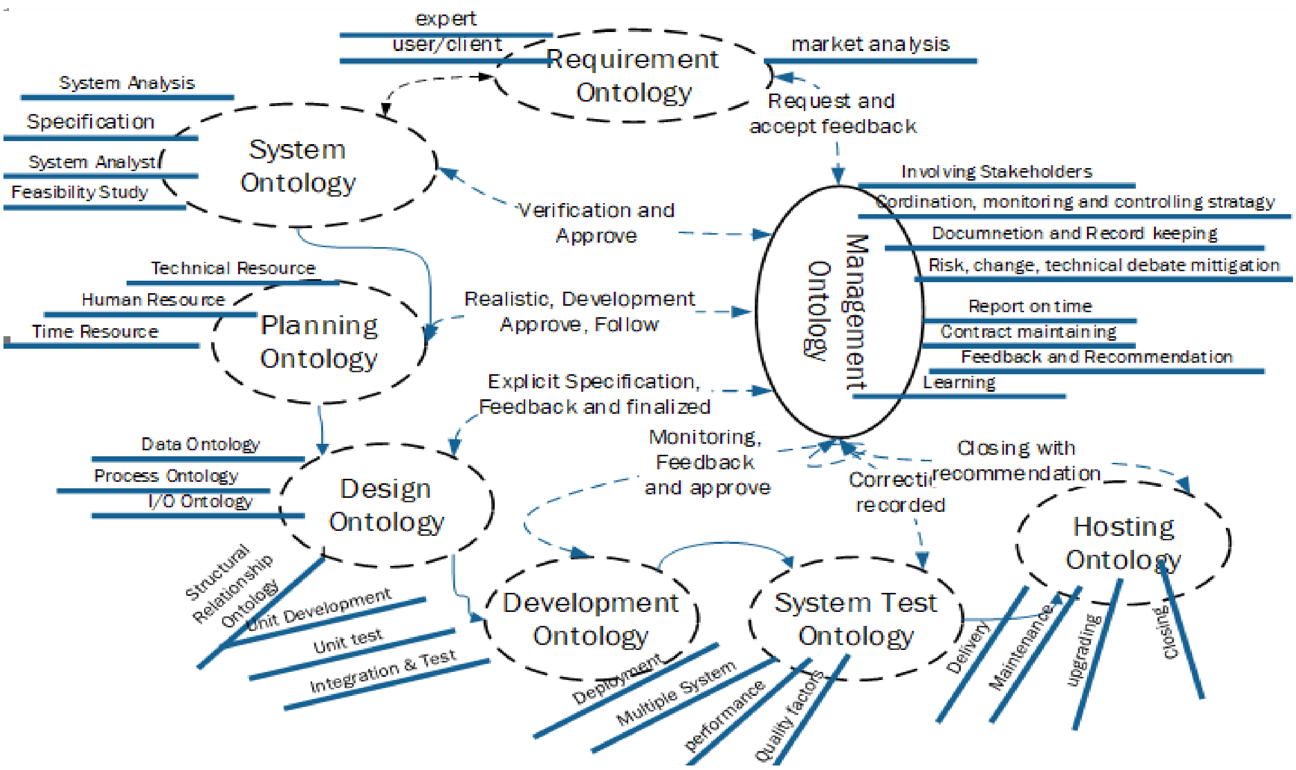

Figure 1. Monolithic Ontological Methodology (MOM); dash lines indicate flexibility in the scope of the phases based on the project's nature while solid lines indicate the fixed functionalities of the stage.

Each phase has an individual set of definitions and actions that are specified with descriptive logic. Well-defined definitions (specific and ambiguity-free) and actions are used to generate rules i.e.

ontology presented by DL. An overview of each phase is represented by a diagram, called lattice ontology. The output of one activity is used as an input of the subsequent phase(s) so standard documentation can reduce the complexity of information sharing. MOM ensures standard records in all phases to improve reusability and maintainability. This methodology may be effective in an online distributed continent development project. The working principle of the proposed methodology is synopsized by the following algorithm

Algorithm:

- *Start*
- *Develop organizational standard document, format, policy, and convention*
- *Include the required phases from the MOM according to the size of the project*
- *Allocate resources: human, time, hardware and software*
- *Apply MOM as follows:*

For phase 1 to n

Repeat until: Submission_of_team$_i$ is not approved by management and keep a record of each update

If Submission_of_team$_i$ is approved by management

Then, Submit to team$_i$+1 and recorded by team$_i$, team$_i$+1 and management

End if

End for

- *End*

Lattice Requirement Ontology: Primary Software Requirements Specification (SRC) is an action that collects functional requirements. The requirements engineers bring completeness by functional and non-functional requirements specification. The quality requirements are emerged by expertise while market analysts show up on additional features for the system. Management approves the requirements according to the business goal of the organization. The integrated requirements $R \equiv UR_i \cup MD_i \cup ER_i$, where R is the set of requirements, $UR_i \subseteq$ user requirements, $MD_i \subseteq$ market demand requirements and $ER_i \subseteq$ expert-recommended requirements. The optimization technique is applicable to remove redundant requirements during the integration process. Also, conflicts appear on requirements represented by a high-quality graphical display that opposes sustainability demand. And it can be formularized by $UR_i \equiv (MD_i)^{\sim} \vee UR_i \equiv (ER_i)^{\sim} \vee MD_i \equiv (ER_i)^{\sim}$. The understanding of quality factors: security, integrity, validity, elicitation, and changes are new challenges in this phase (Tejas Shah & S V Patel, 2014); MOM facilitates the required numbers of iterations with management to assure the quality by reviewing and overseeing. DL is used for specification to reduce misunderstanding among the stakeholders by standard documentation. It will assist in a distance mode working environment. Figure 2 simply digests the requirement finalization process.

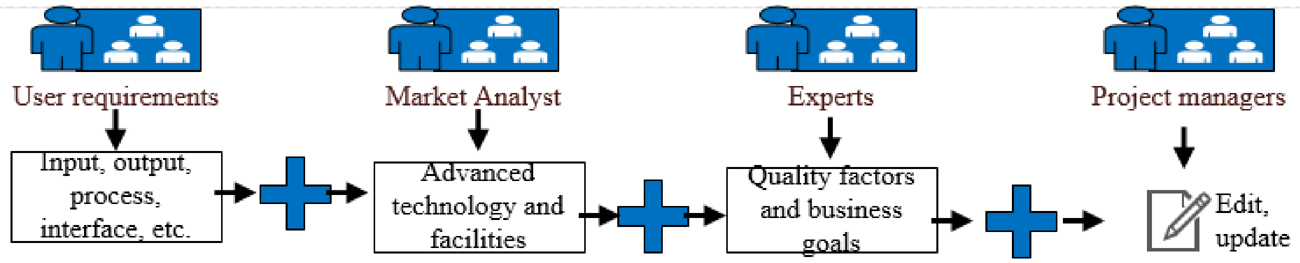

Figure 2. Lattice Requirement Ontology

Lattice System Specification Ontology: When developing a system, the Business Functions (BF) of an organization are converted to Software Functions (SF)s. The systems analysts design a solution that includes all BFs. The BFs and SFs relationship is, *Decompose*(BF)={BF_1, BF_2, BF_3 BF_4……… BF_n} →Mapping($BF_{1,\ldots\ldots,n}$, $SF_{1,\ldots\ldots,n}$) where $SF_i=\sum U_p + BF_{j=0,1,2,\ldots,n}$; and a software process may be unique one (U_p) or with one or more BF so functional dependency can be represented by $SF_i \rightarrow BF_j$. The simplification technique is applied to remove redundant or unnecessary processes by multi-criteria decision-making approach for deducting complexity.The process selection depends on technology, requirements, budget, and business goals of the organization. These are internally dependent on each other and directly align with the objectives of the company. An effective feasibility study (Figure 3) proposes a package with a better combination of minimized risk, optimal throughput, and maximize business profit. A feasibility study includes analysis on: technical, economic, social, time, human and so on. Individual measures are summarized to assess the capacity of the organization. A cumulative collection of individual measures of a software project generates the final output of a feasibility study based on the formula:

$$(OutCome)^{Feasibility} = \sum_{k=0}^{n} \text{Feasibility(Resource)}$$

Where resource={technical, human, cost, time, ethical, etc.}. The study helps to reduce the risks of the project iteration. Actions, processes or quality factors are modernized by management with strategic level thought in the feasibility, aimed to maximize profit and minimize risk. This feedback is important because 64% of organizations faced misunderstanding of the business process, 44% of projects underpinned by missing acceptance while 28% claimed a lack of support from management (Pouya A

K et al, 2018). Hence, MOM introduced formal communication and standard documentation to minimize technical debate.

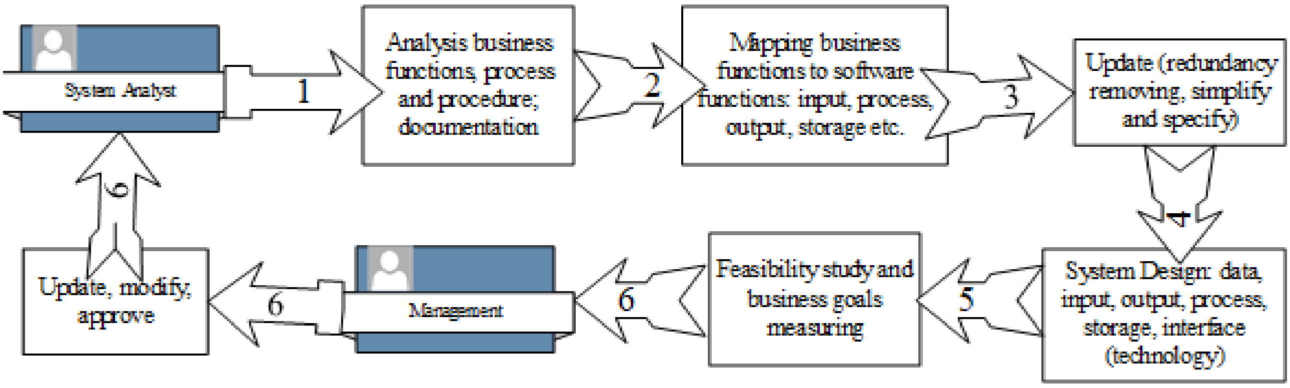

Figure 3. Lattice System Specification Ontology

Lattice Planning Ontology: Software Quality Control (SQC) depends on Software Planning (SP) and Software Quality Assurance (SQA). According to SQA guidelines, and methods a realistic SP development and practice can improve the project success rate. The modeling function of SQC can be, F(SQC)=F(SP, SQA); where F(SP)=F(task, time, resources) and F(SQA)=F(Method, guidelines, standardization, etc.). The managerial plan consists of human resource allocation at an appropriate position based on skills and availability; skill development training hosting; proper utilization of technical resources; effective time management; developing and practicing acceptance policy and monitoring strategy. When the plan is realistic and complete the execution phase of a project can run smoothly. Critical observation on the project includes tracking, monitoring, controlling, and keeping documentation of all events (Figure 4). The ontological monitoring plan records clear-cut and real information which may help in supplementary action like critical performance evaluation

of a process. For bottom-up planning, this information is reused in a similar type of upcoming project. A realistic plan with a standard record of sufficient information minimizes the risk of a project. MOM's automatic alert system shares updated information with respective stakeholders. Moreover, resource allocation becomes more critical in a multi-project environment (Amol Singh. 2014), and MOM mitigates the challenges by attaining maximum utilization of resources.

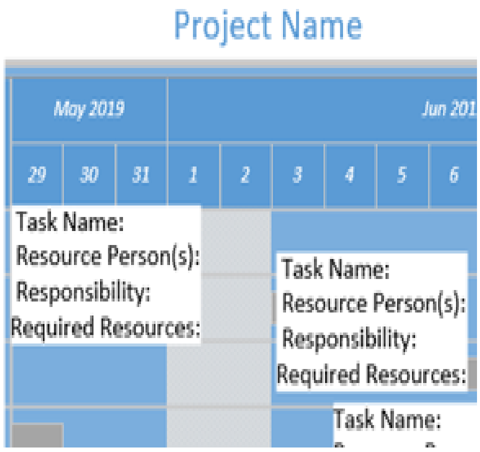

Figure 4. Lattice Planning Ontology

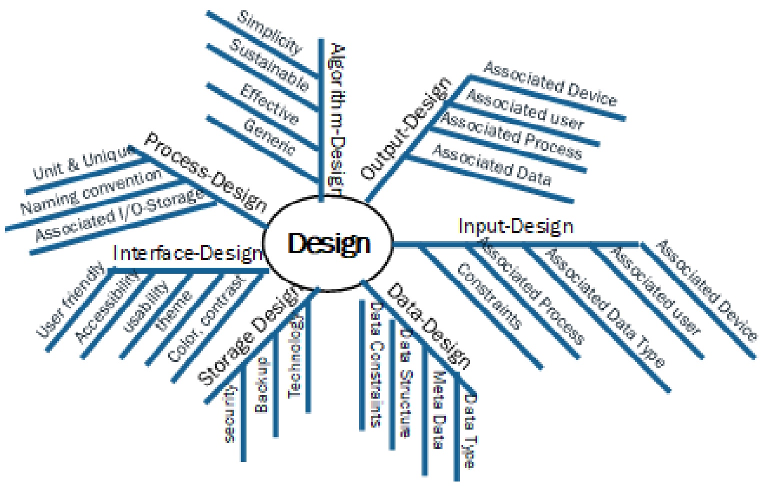

Figure 5. Lattice Design Ontology

Lattice Design Ontology: The design phase comprises software architecture with requirements (Figure 5) that founded on technology. It can be represented as F(infrastructure)=F((hardware, software, database)→ technology). It suggests the setup of hardware, software, and database is one of the most important activities in this phase. An in-depth design includes requirements and their internal and external relationships. For example, an input form influences quality factors of an interface, constraints of the database, and

characteristics of the technology. The context diagram is used to visualize the system, the data flow diagram displays relationships with actors, processes and data files, the entity-relationship diagram shows internal relationships of a database, the use case diagram presents users' rules, system flow and class diagram shows objects' relationship and object-oriented concepts. The aforementioned tools are the members of the tool-set of ontology and are used to specify the information of a system. But MOM gives more importance to DL because it can be an alternative of all by control language, structural context, and predicate logic. Semantic presentation is more flexible and logical to modify and share information in a virtual management system than diagrams. The design phase is decomposed according to the findings of the system specification phase that overcomes the transition challenges (Giuliano Casale et al, 2016). It brings a transition from the application development approach to application composition. It also minimizes ontological challenges (Giuliano Casale et al, 2016) because of its critical reviewing and documentation practice in all phases by management who are experienced.

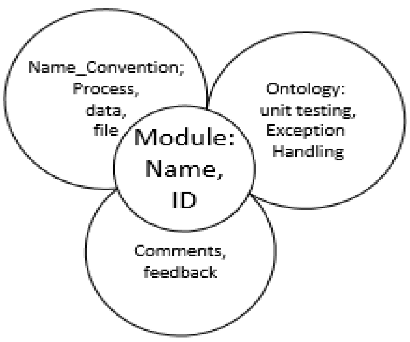

Figure 6. Lattice Code Ontology

Lattice Development Ontology: any shortcomings of the previous phases create more challenges in the coding phase (Giuliano Casale et al, 2016). The MOM provides standard documentation with logical relationships to reduces the risk of this phase. It also ensures coding documentation with a sufficient scope of modification (figure 6). Besides, it reduces the challenges including ambiguity, technical debate and repeating activities (Giuliano Casale et al, 2016) that are common in the conventional coding phase. The MOM module is a unit of an application that is defined by naming convention for file, process, data, and comments. This normalizing meta-data assures the reusability of information, change control and it makes the project maintainable (figure 6). A module consists of a testing template and exception handling ontology in the coding phase. The implementation of logbooks to keep performance records in the unit testing process (figure 7) improves the documentation system and quality of the product. A cluster is developed with a set of interrelated modules and it is evaluated by the cluster testing template. The integrated clusters are the system that is analyzed by system testing. System testing is performed by real users and domain experts before deployment for operation. Module and cluster testing are introduced by MOM so that system testing becomes more efficient and decreases backtracking activities. Less number of claims in the maintenance stage will improve user satisfaction too. Moreover, an exclusive user manual will also improve user performance.

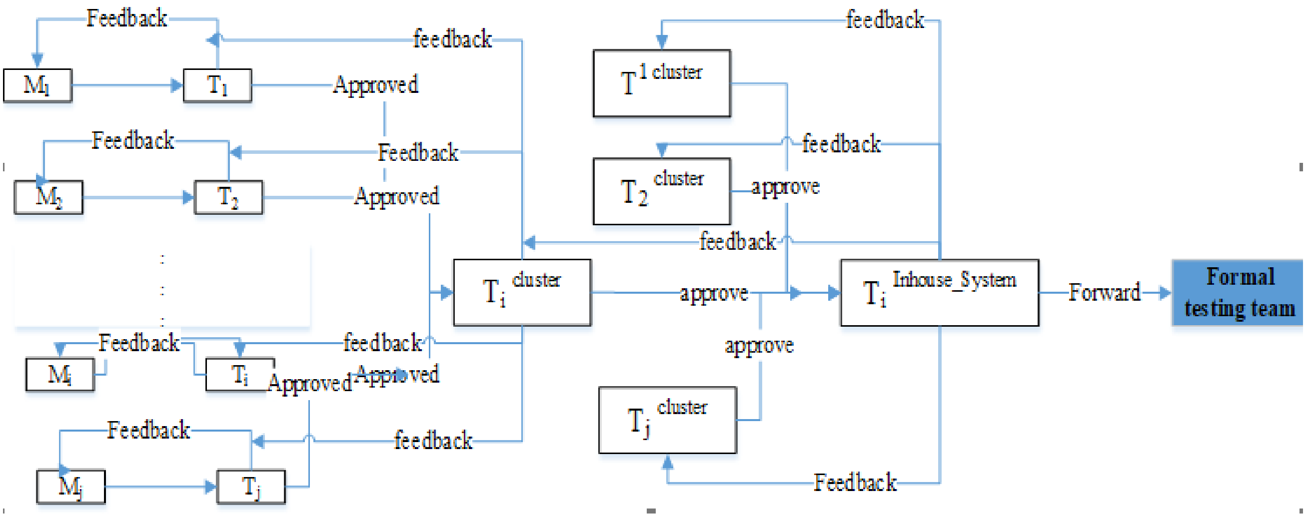

Figure 7. Lattice Testing Ontology

Testing evaluates the completeness of functionalities and incorporates quality factors. Accuracy is acceptable when the required output(Yi) is found for exact input(Xi) according to requirements and it is denoted by, (Expected output(Y(ti)) ≡ exact output(Y(ti)))← ((Expected input(X(ti)) ≡ exact input(X(ti))). Efficiency is the effectiveness of an action, response, and notification that is a function of time. For example, if the required time(Ti) is the minimum time to get output(Yi) for input(Xi) by the process(Pi) then it is an efficient process. In a multimedia application, quality is an optimal output that is accepted based on multiple parameters. For example, *video_conferencing_quality ={synchronization, response_time, audio_signal, video_signal}*. Maximum throughput is only expected when an efficient algorithm is implemented with effective technology. A secure transaction system, information sharing, and communication enhance reliability. In addition, information privacy, threat monitoring and protection, user authentication and privilege are also the parameters of reliability. It consists of policies and procedures for in-house information access, control, and monitoring. Portability defines the scope of a system to adapt to future upgrade versions and variations in platforms. Generalization in design and development enhances the portability in MOM. DL does not depend on technology or computer programming. It is flexible to be implemented for clients and closing of the project with the user manual and managerial document.

- **Maintenance:** a set of services to the client under certain conditions for a duration of time. The mode of communication and service price should be clearly defined.
- **Upgrade:** enhancing the system under a mutual agreement to fulfill near future stipulations of users.
- **Closing:** ending of a contract according to acceptance policy and property right of source code.
- **Hosting:** System submitted for operation and starting access by utilization that consists of few activities mentioned before.

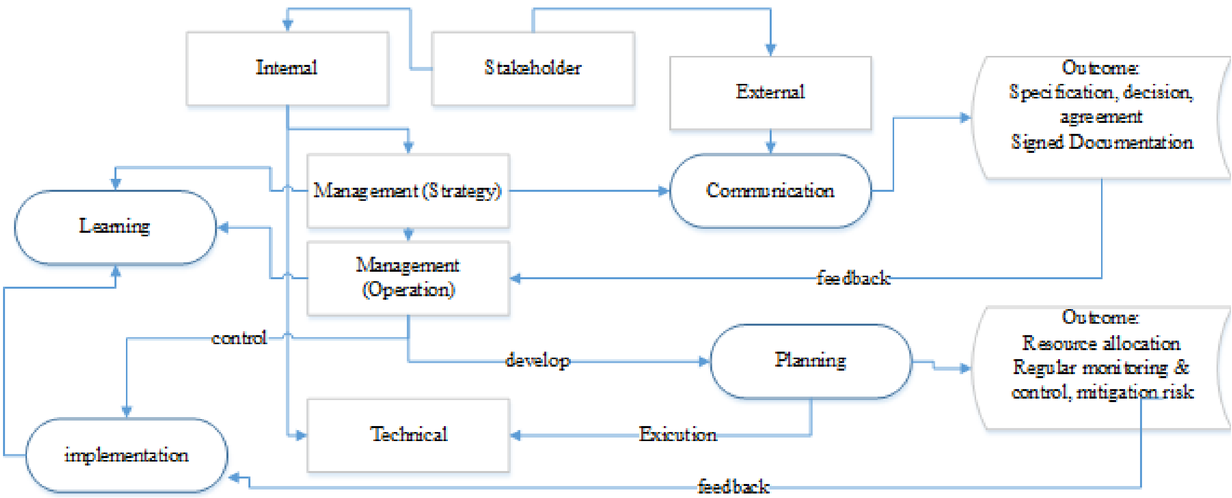

Figure 8. Lattice Management Ontology

Lattice Management Ontology: The aim of this research is to bring all the activities of a project under MOM, which is a unique manageable process. All MOM phases are directly connected with the management team for tracking, monitoring, and controlling effectively. It reduces decision-making time in any phase. A common data-sharing platform is upgrading with the latest decision. So concerned stakeholders are aware of any change and standard documentation reduces ambiguity. The activity mesh of management is reflecting in figure 8. It is a comprehensive diagram that shows processes, methods, strategies and stakeholders' communication. Management is responsible to implement a project and taking necessary actions to make success. Moreover, MOM improves the responsiveness of the stakeholders and emphasis learning from the project. The management team is leading the project and responsible to resolve technical debates, conflicts of interest and risks.

MODEL ANALYSIS

MOM phases are highly influenced by management because this is the key team in a project. A hardworking and well-trained management team can achieve project and therefore business goals. MOM is developed for average-size projects that aimed to ensure the quality of product and process. A group of skilled team members handles different types of software projects in a multi-project environment. There are plenty of influential factors for software project management (Linåker J. & Regnell B, 2017; Luigi Lavazza et al, 2016; Edson Oliveira et al, 2018)and carry different values according to the nature of a software product. A set of common influential factors are selected to perform a comparative study based on the management of the quality of the process, product, and resources in table 1.

Table 1. Selected Influential Factors of a Project

Features	**Extreme Programming (xp)**	**Scrum**	**PRINCE2**	**MOM**
Development Approach	Iterative and incremental	Iterative and incremental	Incremental / shared	Plan-driven with single-phase iterative
Stakeholders' participation	Throughout the process	Not defined	Formal	Throughout the process but more formal
Project/process management	Not defined	Practice available	No focusing on people management	Highly involvement
Documentation	Less	Less	Documentation oriented	More and formal throughout the process
Development order defined by	User/client or customer	Scrum team	Management	Project management team
Acceptance criteria	Defined	Defined	Defined	Step by step generated
Testing	Acceptance/unit/ integration	Not mentioned	Defined	Unit testing by unit developer, integration & system by the required persons
Adaptability	Focusing on object-oriented	Focusing on object-oriented	Can be adapted to any project	Can be adapted to any project
Product quality	Not mentioned	No clear project definition	Focusing	Focusing
Change control	Not mentioned	Frequently change	Partial	Well-structured management
Working environment	Face to face collaborative	Not clear	Shared	Distance environment

Though the generalization comparison includes so many influential factors in table 1; but AHP includes only the most important five factors (table 2). So it is evaluated by an expert who has more than 15 years of experience in the software industries. The methodologies are comparing with *Stakeholders' Involvements, Management's Activeness, Shared Environment, Formal Documentation*, and *Focusing on Quality of Product*. The comparison comprehends XP, PRINCE2, and Scrum with MOM because those have a better influence on the selected comparative factors. with the MOM. For AHP the paired comparison is mapped in table 2 by the expert during January 2020. Four methodologies create a 4×4 reciprocal matrix for each factor from table 5.1 and according to the following algorithm.

Algorithm: Analytic Hierarchy Process (AHP)

START

1. ***Generating a reciprocal matrix:***

- *SET 1 for the common field of the same methodology and it will cover one diagonal.*
- *Insert actual judgment to the left/right of 1s' (left side here: 9,7,5,3,)*
- *Insert reciprocal value to another side (Aij=1/Aji) (right side here:1/9,1/7,1/5,1/3)*

2. ***Generating Normalized matrix***

- *Normalized matrix: Divide each element by the sum of the elements of that column*

3. ***Priority Value:***

- *the sum of all row values*
- *A high value means better*

END

A reciprocal matrix is generated for each analogy factor of table 2 and chronologically mentioned in the left-most column of table 3. It has the respectively normalized form of matrix beside and followed by priority values. The priority values are the actual cumulative and this is visualized by figure 9. MOM is better than any other in the area of management-involvement, documentation, and integration of quality factors. It achieves a higher rank in the consideration of "stakeholder engagement" and "sharing the working environment" with PRICE2. Hence, it is better for standard organizations.

Table 2. Pair Comparison

Pair Comparison Scale (odd numbers from 1 to 9): Extreme Favors(EF)=9, Very Strong Favors(VSF)=7, Strongly Favors (SF)=5, Slightly Favors(SlF)=3, Equal (E)=1											
Factors	**Methodology**	**EF**	**VSF**	**SF**	**SLF**	**E**	**SLF**	**SF**	**VSF**	**EF**	**Methodology**
Focusing on Effectiveness of Management	MOM			√							PRINCE2
	MOM				√						Scrum
	MOM	√									XP
	PRINCE2				√						Scrum
	PRINCE2		√								XP
	Scrum	√									XP
Focusing on Effectiveness of Documentation	MOM				√						PRINCE2
	MOM		√								Scrum
	MOM		√								XP
	PRINCE2			√							Scrum
	PRINCE2			√							XP
	Scrum					√					XP
Focusing on Effectiveness of Sharing environment	MOM					√					PRINCE2
	MOM		√								Scrum
	MOM	√									XP
	PRINCE2	√									Scrum
	PRINCE2		√								XP
	Scrum					√					XP

Focusing on Effectiveness of Stakeholders	MOM					√					PRINCE2
	MOM				√						Scrum
	MOM			√							XP
	PRINCE2				√						Scrum
	PRINCE2			√							XP
	Scrum					√					XP
Focusing on Quality of the Product	MOM				√						PRINCE2
	MOM			√							Scrum
	MOM		√								XP
	PRINCE2				√						Scrum
	PRINCE2		√								XP
	Scrum						√				XP

Table 3. Comparison (Reciprocal Matrix, Normalized Matrix, Priority Value) with AHP

Reciprocal Matrix (1)

	MOM	PRINCE2	SCRUM	XP
MOM	1	5	3	9
PRINCE2	0.2	1	3	7
SCRUM	0.3333	0.3333333	1	9
XP	0.1111	0.1428571	0.1111	1

Normalized Form (1)

	MOM	PRINCE2	SCRUM	XP
MOM	0.6081081	0.7720588	0.421875	0.3461538
PRINCE2	0.1216216	0.1544118	0.421875	0.2692308
SCRUM	0.2027027	0.0514706	0.140625	0.3461538
XP	0.0675676	0.0220588	0.015625	0.0384615

Priority (1)

MOM	0.5370489
PRINCE2	0.2417848
SCRUM	0.185238
XP	0.0359282

Reciprocal Matrix (2)

	MOM	PRINCE2	SCRUM	XP
MOM	1	3	7	7
PRINCE2	0.3333	1	5	5
SCRUM	0.1429	0.2	1	1
XP	0.1429	0.2	1	1

Normalized Form (2)

	MOM	PRINCE2	SCRUM	XP
MOM	0.6176471	0.6818182	0.5	0.5
PRINCE2	0.2058824	0.2272727	0.3571429	0.3571429
SCRUM	0.0882353	0.0454545	0.0714286	0.0714286
XP	0.0882353	0.0454545	0.0714286	0.0714286

Priority (2)

MOM	0.5748663
PRINCE2	0.2868602
SCRUM	0.0691367
XP	0.0691367

Reciprocal Matrix (3)

	MOM	PRINCE2	SCRUM	XP
MOM	1	3	7	7
PRINCE2	0.3333	1	5	5
SCRUM	0.1429	0.2	1	1
XP	0.1429	0.2	1	1

Normalized Form (3)

	MOM	PRINCE2	SCRUM	XP
MOM	0.6176471	0.6818182	0.5	0.5
PRINCE2	0.2058824	0.2272727	0.3571429	0.3571429
SCRUM	0.0882353	0.0454545	0.0714286	0.0714286
XP	0.0882353	0.0454545	0.0714286	0.0714286

Priority (3)

MOM	0.5748663
PRINCE2	0.2868602
SCRUM	0.0691367
XP	0.0691367

Reciprocal Matrix (4)

	MOM	PRINCE2	SCRUM	XP
MOM	1	1	3	5
PRINCE2	1	1	3	5
SCRUM	0.3333	0.3333333	1	1
XP	0.2	0.2	1	1

Normalized Form (4)

	MOM	PRINCE2	SCRUM	XP
MOM	0.3947368	0.3947368	0.375	0.416666
PRINCE2	0.3947368	0.3947368	0.375	0.416666
SCRUM	0.1315789	0.1315789	0.125	0.083333
XP	0.0789474	0.0789474	0.125	0.083333

Priority (4)

MOM	0.3952851
PRINCE2	0.3952851
SCRUM	0.1178728
XP	0.091557

	MOM	PRINCE2	SCRUM	XP
MOM	1	3	5	7
PRINCE2	0.3333	1	3	7
SCRUM	0.2	0.3333333	1	0.333
XP	0.1429	0.1428571	3	1

	MOM	PRINCE2	SCRUM	XP
MOM	0.5965909	0.6702128	0.4166667	0.4565217
PRINCE2	0.1988636	0.2234043	0.25	0.4565217
SCRUM	0.1193182	0.0744681	0.0833333	0.0217391
XP	0.0852273	0.0319149	0.25	0.0652174

MOM	0.534998
PRINCE2	0.2821974
SCRUM	0.0747147
XP	0.1080899

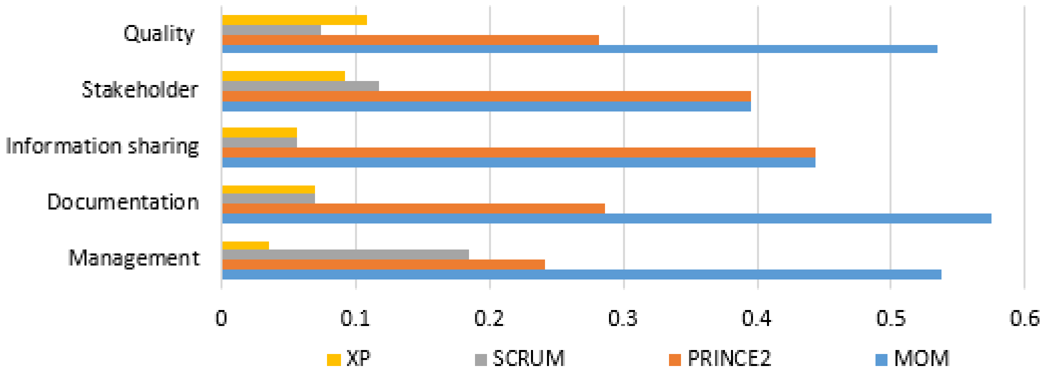

Figure 9. Result Analysis

CONCLUSION

This study proposed an explicit specification methodology that adaptively selects resources of a project from the organization. The MOM is a formal methodology that minimizes the challenges of a project by increasing more managerial interaction in all phases. MOM's explicit documentation that is standardized by DL reduces ambiguity and technical debate. An open-source market of software development and research field can enhance their capability by practicing MOM. This should support an online project management system. Its processes keep the record of all changes, data transfer, and process management actions. It will enhance the reusability and maintainability of the system. Moreover, DL information is applicable to process by an application and knowledge extraction. The system management process of MOM accelerates employees' interaction to improve performance. Stakeholders become responsive because of MOM's clarified working functions. The working load will be measurable and well-defined for each. MOM can easily adjust future enhancements if need. It will ensure the quality of a product because of multiple reviewing systems and reduce project failure rates. It is mainly recommended for a standard software development organization. In a similar type of project, existing information (design, code) can be easily re-used. But, MOM in individual problem solving will face overwhelming because of the in-depth specification and explicit documentation. Once an automated MOM system is drawn up in an organization, their work becomes easier for furthermore projects. It will diminish the feasibility study and execution time for imminent projects and that helps to make an early decision in the project initiation phase. Furthers works might combine more virtual project management features with artificial intelligence approaches to improve the quality of management.

LIMITATION AND FUTURE WORK

This research performs comparison only major factors and future studies may also consider more factors for AHP comparison. Moreover, the research could be evaluated by alternative multi-criteria decision-making tools. It is evaluated by a single expert but could be compared from multiple experts. Also, after implementing into the software industry we can get the real reflection. The findings in this article point to the need to track 7 phases but further study may focus on the control of managerial activities.

ACKNOWLEDGEMENT

Hasan Mahamud Rana (Manager, *Fiftytwo* Digital Ltd. Bangladesh) & Abdullah Al Muqim (Lead Developer and manager, *Coca-cola*, USA) to attend the evaluation process.

References

S. Hastie, & S. Wojewoda. 2015. Standish group 2015 chaos report - q&a with Jennifer Lynch. Retrieved by 28th Feb 2018. Available: https://www.projectsmart.co.uk/white-papers/chaos-report.pdf

Timo O. A. Lehtinen, Mika V. Mäntylä, Jari Vanhanen, Juha Itkonen, & Casper Lassenius. 2014. Perceived Causes of Software Failure-An analysis of their relationships, Journal of Information and Software Technology. 56:623-643.

N. Cerpa & J. M. Verner. 2009. Why Did Your Project Fail. Communication of the ACM. 52(12):130-134.

Al-Ahmad W, Al-Fagih K, Khanfar K, Alsamara K, Abuleil S, & Abu-Salem H. 2009. Taxonomy of an IT Project Failure: Root Causes. International Management Review. 5(1):93-104.

Noor Habibah Arshad, Azlinah Mohamed & Zaiha Mat Nor. 2007. Risk Factors in Software Development Project. Proceedings of the 6th conf. WSEAS (Software Engineering Parallel and Distributed System) 51-56.

L. McLeod & S. G. MacDonell .2011. Factors that Affects Software System Development Project Outcomes: A survey of Research ACM Computer. 43(4): 24-55.

Gruber T,. 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies, 43(5/6): 907-928.

Guarino N. 1998. Formal Ontology and Information System. Proceedings of the First International Conference on Formal Ontologies in Information Systems, Trento, Italy. 3-15.

Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. 1991. Enabling Technology for Knowledge Sharing. AI Magazine, 12(3): 36-56.

Lai L.F. 2007. A Knowledge Engineering Approach to Knowledge Management. Journal of Information Sciences. 177(19): 4072-4094.

Fensel Dieter. 2004. Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer Verlag, Berlin. Second edition.

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, & Sherlock G. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet.25(1):25-9.

Clément Jonquet, Anne Toulet, Elizabeth Arnaud, Sophie Aubin, Esther Dzalé Yeumo, Vincent Emonet, John Graybeal, Marie-Angélique Laporte, Mark A. Musen, Valeria Pesce, & Pierre Larmande. 2018. AgroPortal: A vocabulary and ontology repository for agronomy" Computers and Electronics in Agriculture 144:126-143.

Sunitha A. & Suresh B. G. 2014. Ontology-Driven Knowledge-Based Health-Care System an Emerging Area - Challenges and Opportunities – Indian Scenario. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XL-8:239--246

Gómez-Pérez A. & Corcho O. 2002. Ontology languages for the semantic web. IEEE Intell. Syst. 17(1):.54–60.

Heiyanthuduwage, S.R., Schwitter, R. & Orgun, M.A. 2016. OWL 2 learn profile: an ontology sublanguage for the learning domain. SpringerPlus 5: 291

Fluit C., Sabou M., & van Harmelen F. 2006. Ontology-Based Information Visualization: Toward Semantic Web Applications. In: Geroimenko V., Chen C. (eds) Visualizing the Semantic Web. Springer, London. .45-58

David Riaño, Francis Real, Joan Albert López-Vallverdú, Fabio Campana, Sara Ercolani, Patrizia Mecocci, Roberta Annicchiarico, & Carlo Caltagirone. 2012. An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. Journal of Biomedical Informatics. 45(3):429-46.

Kamal Uddin Sarker & Dr. Aziz Deraman. 2017. Design Aspects of Near Future Soft Computing. International Journal of Electronics Communication and Computer Engineering. 8(5): **318-23.**

K. U. Sarker, A. B. Deraman & R. Hasan. 2018. Descriptive Logic for Software Engineering Ontology: Aspect Software Quality Control. Proceeding *4t^h International Conference on Computer and Information Sciences (ICCOINS)*. 1-5.

Kamal Uddin Sarker, Dr. Aziz Deraman, & Raza Hasan, 2018. Green Soft Computing. Conference: ICEL3S2018. Journal of Fundamental and Applied Science. 65(10):-462-70.

K. U. Sarker, A. Bin Deraman, R. Hasan, S. Mahmood, A. Abbas & M. Sohail. 2019. Kids' Smart Campus Ontology to Retrieve Interest. *4th MEC International Conference on Big Data and Smart City (ICBDSC)*. 1-4.

Sarker Kamal Uddin, Deraman, Aziz Bin Deraman, Hasan Raza, & Abbas Ali. 2019. Ontological Practice for Big Data Management. International Journal of Computing and Digital Systems. 8(3):265-73.

K. Schwalbe. 2009. Information Technology Project Management. 6th edition, Cengage Learning.

R. S. Pressman. 2005. Software Engineering, A Practitioner's Approach. Sixth Edition, Mc Graw. Hill.

L. Williams. 2010. Agile software development methodologies and practices. *Advances in Computers*. 80: 1-44.

Faiza Anwer, Shabib Aftab, Syed Shah Muhammad Shah & Usman Waheed. 2017. Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum. International Journal of Computer Science and Telecommunications. 8(2):1-7

Radka Vaníčková. 2017. Application of PRINCE2 Project Management Methodology. Studia commercialia Bratislavensia. 10 (38): 227-238

Tejas Shah & S V Patel. 2014. A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods. International Journal of Computer Applications. 99(15): 36-45

Pouya A K, Florian M, Martin G, Matheus H, & Stefan V. 2018. Business Capability Maps: Current Practices and Use Cases for Enterprise Architecture Management. Proceedings of the 51st Hawaii International Conference on System Sciences. 10(581): 4603-12.

Amol Singh. 2014. Resource Constrained Multi-Project Scheduling with Priority Rules & Analytic Hierarchy Process. 24th DAAAM International Symposium on Intelligent Manufacturing and Automation Procedia Engineering. 69: 725–34

Giuliano Casale, Cristina Chesta, Peter Deussen, Elisabetta Di Nitto, Panagiotis Gouvas, Sotiris Koussouris, Vlado Stankovski, Andreas Symeonidis, Vlassis Vlassiou, Anastasios

Zafeiropoulos, & Zhiming Zhao. 2016. Current and Future Challenges of Software Engineering for Services and Applications. Procedia Computer Science. 97:34 – 42

Haneen Hijazi, Shihadeh Alqrainy, Hasan Muaidi, & Thair Khdour. 2014. Risk Factors in Software Development Phases. European Scientific Journal. 10 (3):213-31.

Kamal Uddin Sarker, Aziz Bin Deraman & Raza Hasan. 2020a. Ontological practice for software quality control. International Journal of Business Information Systems. Inderscience Enterprises Ltd. 34(3):355-372.

Sarker KU, Deraman A,Hasan R, & Abbas A. 2020. Explicit specification framework to manage software project effectively. Indian Journal of Science and Technology. 13(16):3785-3800.

Kamal Uddin Sarker, Aziz Bin Deraman, Raza Hasan & Ali Abbas. 2020b. SQ-Framework for Improving Sustainability and Quality into Software Product and Process. International Journal of Advanced Computer Science and Applications(IJACSA). 9(11):69-78.

V.K. Chawlaa, A.K. Chanda, S. Angra, & G.R.Chawla. 2018. The sustainable project management: A review and future possibilities. Journal of Project Management. 3(3):157–170.

Linåker J. & Regnell B. 2017. A Contribution Management Framework for Firms Engaged in Open Source Software Ecosystems - A Research Preview. In: Grünbacher P., Perini A. (eds) Requirements Engineering: Foundation for Software Quality.Springer, Cham. 10153: 50-57.

Luigi Lavazza, Sandro Morasca, & Davide Tosi. 2016. An empirical study on the effect of programming languages on productivity. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16). Association for Computing Machinery, New York, NY, USA, 1434–1439.

Edson Oliveira, Tayana Conte, Marco Cristo, & Natasha M. Costa Valentim. 2018. Influence Factors in Software Productivity - A Tertiary Literature Review. International Journal of Software Engineering and Knowledge Engineering. 28(11-12):1795--1810.