INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

Received: 28-06-2020

Accepted: 19-09-2020

Published: 12.10.2020

Editor: Dr. Natarajan Gajendran

Citation: Sarker KU, Deraman A, Hasan R, Abbas A (2020) Explicit specification framework to manage software project effectively. Indian Journal of Science and Technology 13(36): 3785-3800. https://doi.org/10.17485/IJST/v13i36.1244

*__Corresponding author__.

ku_sarker@yahoo.com, ksarker@mec.edu.om

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Explicit specification framework to manage software project effectively

Kamal Uddin Sarker[1,2]*, Aziz Deraman[1], Raza Hasan[3], Ali Abbas[2]

1 Faculty of Ocean Engineering Technology and informatics, Universiti Malaysia Terengganu, Malaysia
2 Department of Computing, Middle East College, Muscat, Oman
3 Institute of Informatics and Communication, Malaysian University of science and technology, Selangor, Malaysia

Abstract

Objectives: This research addresses the causal relationship among processes, tasks, resources, and stakeholders of the software projects to improve the success rate that is 31% only now (CHAOS report 2019). **Methods:** A software project management framework is developed based on the findings of an empirical study on recent commonly used methodologies and root cause analysis on the reasons for project failure. The cause-effect relationship is prioritized by Structural Equation Modeling (SEM) to develop the Explicit Specification Framework (ESF). The framework is tied to the pinpointing with ontology for formal presentation, descriptive logic for unambiguous information specification, and ikigai for decision-making. **Findings**: The proposed framework is evaluated by a group of experts and their average scope is evaluated by a group comparison mechanism called Grey Rational Analysis (GRA). The evaluation process considered 6 parameters and compared it with 4 commonly used project management approaches. The highest GRA rank is found for the average of six distinct parameters though it is comparatively more complex for explicit documentation. The three-layer architecture of ESF remarks on roles, functions, and respective stakeholders in each layer presented by a comprehensive comparison study with literature and found that it enhances formal practice, sustainability, and management capability. ESF is easy to implement for distributed project management and improving automation facilities in the software industries. **Novelty/Application:** Artificial intelligence, the Internet of things, and big data would be common elements of a software project and ESF would meet the demand for project information specification, management, and control.

Keywords: Software Project Management; Software Quality Control; methodology; multi- criteria decision making; structural educational modeling; ontology

1 Introduction

Software Engineering (SE) emerges systematic project management practice in software industries[1] for enhancing productivity and integrating quality factors. In addition, SE focuses on software development tasks, processes, and quality assurance[2]. It also includes formal workflow in the software industry with the documentation for all traditional phases of the system development life cycle[3]. At the early age of software engineering, standardization organizations and professional bodies were introduced. International Organization for standardization (ISO), Institute of Electrical and Electronics Engineers (IEEE), and Capability Maturity Model Integration (CMMI) are well-known standardization organizations in information technology. Professional bodies and certification companies are also upgrading tools, guidelines, and policy-procedure to improve the quality of the software.

The Standish Group report[4] showed that the software project failure rate is decreasing very slowly and till now nearly 50% of the projects are challenging, 20% failed, and rest 30% is only successful. According to their modern classification resolution of FY2011-2015; grand, large, medium, moderate and small are the classes of the software projects. And the report noticed that the larger project faces more challenges than a smaller project and failure rate is linearly related to the size of the project. Moreover, the analysis of public repository[5,6] has shown that time estimation of a project highly influenced by the user requirements. Effective hours and duration of the project are linearly proportioning with the size of a project, but user demand can break the linear relationship. So, human resources, time, and technical resources estimation become a critical task, and external entities like to influence on planning.

According to the Project Management Institute[7], a project could be a success when it is successful in the process handing, project management, project delivery, business goals, and organizational strategies. The projects' performance depends on the satisfaction rate of the product that is given by the client, user, and relevant stakeholders. It also measured by the ratio of investment and return for business purposes. Sometimes, it is confusing to define a project is successful or not but our consideration is very clean with the following three resolutions.

Resolution-1: A project is a failed project, if and only if it stops at any phase without delivering the product after executing it.

Resolution-2: If a project is delivered to the customer with lacking quality measures such as poor user satisfaction and time or/and budget exceeds, it is called a challenging project.

Resolution-3: A successful project that is delivered on time within budget with required functional and non-functional requirements that enhance user satisfaction.

At the beginning state, all projects seem to succeed, at the middle phases, it faces issues and moving towards into challenging state; if the project team could not control properly it ended with failure or challenging status. This research wants to avoid an unexpected situation rather than facing the situation. A software project could not be a success or fail only for a single person, task, or process but it is teamwork, and the team is responsible to make success. It is going to propose a project management framework with the target to cut the number of challenging project rates (resolution-2) so that the success rate would be increased and the failure rate would be decreased. So the Explicit Specification Framework (ESF) integrates the role of process, tools, and person in a closed framework that leads to formal management practices.

This research proposes, elaborates, and evaluates a software project management framework. It has a systematic management structure and formal practice that can improve the quality of a software product. And, it will improve the success rate of the software industries and business goals. The quality guidelines of the framework would contribute to improving user satisfaction. In addition, it would enhance sustainability by using ontology and ikigai.

2 Research Methodology

Section 1 presents the value of the study and research domains. Section 3 consists of a brief explanation of needful tools and techniques to manage software project management. This section has summarized the upcoming actions and techniques that are going to carry out in this work. Moreover, it consists of the hypothesis of the research (Table 3) that focuses on the major factors that need to consider for the proposed framework development. Section 4 has structured the framework with consideration of the project complexity parameters: size, risk, nature, stakeholder, requirements, quality, management, methods, and tools. It has also been addressed to resolve unexpected threats of a project by formal communication, proper documentation, realistic plan, proper method, sufficient feasibility study, change control plan, effective management plan, risk analysis, and mitigation plan. This framework decomposes the fundamental system complexity factors in smaller and simples form that reduces risk and ambiguity.

It is a three-layer project management framework that is given in section 4. Ontology, descriptive logic, and ikigai have been proposed as the tool in the project initiation phase for requirement analysis and feasibility study. Project execution is the second step where ontology could be a tool for manual and automatic design. The third stage carries previous tools to improve the management process, documentation and record-keeping practice, and sustainability practice by re-usability of existing resources. Subsections having each part of the framework with the figure, logic, and mathematical modeling. The framework is evaluated in section 5 by the factors of the hypothesis as well as the complexity of a methodology. GRA is used to evaluate according to the response of the experts based on the variables of GRA (section 3.6) and shows significant improvement by formal practice. It is comparatively better than few similar methodologies (plan-driven and agiles) in certain criteria.

3 State of the Literature

3.1 Background study

The research has been structured, illustrated, and analyzed by ontology, Descriptive Logic (DL), ikigai, Structured Equation Model (SEM), and Grey Rational Analysis (GRA). Each of these has a historical background of development and applications. Ontology has been applying in multidisciplinary applications after introducing from philosophy; while ikigai was introduced for measuring life and job satisfaction but increased its' application in decision-making. DL is a pioneer tool than relational algebra for rule setting in information specification for a domain. SEM is a statistical approach for structural analysis of a model or framework. GRA is being used for a comparative study on different approaches in the same domain to generate rank. Ontology is used to build scientific reasoning for project handling. It suggests explicit specification of domain information with DL so that it could be used to the automatic project management system. Moreover, the DL document is ambiguity free and easy to restructure, and the organization can improve the re-usability of information. The aim of this project is to improve sustainability, maintainability, and quality in the project as well as the product.

3.2 Ontology

Ontology has been introduced for information sharing in computer science by Gruber(8) to specify the domain knowledge. Moreover, it brings uniformity in the structure of informatics, rules, and the relationship of domain knowledge. Nowadays, most use of ontology in semantic web applications(9,10) and followed by W3C to develop semantic web ontology language(11). It has been applied for genome data modeling(12) that brought a common working platform for gene data specification in research; agricultural ontology(13) has been settled to accept and share common data among users. Patient information collection, investigation data, and treatment record has been managed and analyzed by a health informatics system called health ontology(14). In software engineering, data visualization ontology(15) is implemented in a procedural approach; pattern matching ontology is used for information searching(16) and data cleansing ontology(17) to remove irrelevant data. Our advanced works modeled graphical ontology for requirement specification(18); descriptive logical ontology for integrated quality factors(18); relational ontology to integrate green computational criteria(19) and ontological design architecture for the internet of things based application(20). In this framework, we have integrated automatic ontology to improve the sustainability practice in software farms.

3.3 Ikigai

Ikigai is a Japanese concept and Kamiya developed this model to measure the life satisfaction of a person with overlapped multiple circles of a van-diagram(21). It describes having a sense of purpose of life as well as the inner self of an individual. The psychologist used this for multifaceted record construction for information modeling(22). Kumano used ikigai to show the cause-effect relationship in information specification and the author used structural equation clustering for developing ikigai for clustering human lifestyle data into two groups (positive and negative)(23). We prefer the van-diagram architecture of ikigai to build a model with the criteria of the feasibility study. This model helps to make a quick decision by measuring the organizational capabilities and project requirements. Moreover, the project manager can do resource allocation effectively.

3.4 Standardization

A project has a temporary effort to develop a product or service that has unique results(24) and it is new or a modified one(25). There is a set of interrelated tasks having a logical sequence. A project team accomplishes the activities by using the right tools, technology, and methods. A team could be subdivided into several sub-teams and a sub-team has to complete a specific task of the project in a fixed time. The outcome of a project depends on the performance of each team member where team management is the most important task. This framework shows interest in team management activities too. Project management is a systematic study to enhance teams' capability. There is no literature that focuses on a unique reason for project failure, but the most important reason is the weakness of team management[26]. A good reflection was given by the article[27] for project failure, which highlighted the weakness of method and linkage with tools, documentation, and choice of computer language. It is directly blameworthiness on the group performance instead of an individual act. According to Ardhendu M. and S. C. Pal[28], a successful software project should complete on time and budget with functionalities and good quality. Our work also includes sustainability factors along with quality factors of IEEE, CMMI, and ISO standards. The operation time of the project dependent on the business goal of the company too, but it is in threat due to the rapid progress of technology.

The business goal of software does not only depend on efficiency, portability, flexibility, accuracy, and sustainability but also the lifetime of the product too. Energy consumption of the software product is becoming higher day by day and the federal energy regulatory commission of the USA showed importance on sustainability practice by computing[29]. The report has encouraged to apply green soft computing by integrating less power-consuming hardware, applying effective algorithms, and an effective storage system in a device. This report and our earlier works[18–20] encourages us to work on this domain too. This research accessed on the architecture of Software QUALity Enhancement (SQUALE)[30] that is founded on the ISO-9126 standard to retrieve software quality factors. SQUALE architecture is more detailed than ISO-9126 with detailed explanation and that assessed by code and documentation. The measurement metrics, rule checker, and top-down quality assessment by predefined criteria and factors that help to develop the proposed framework. Our work is flexible in fixing the criteria of a project and criteria are selected by Structural Equation Modeling (SEM).

3.5 Structural equation model

Researchers are implementing Structural Equation Model (SEM) to find latent variables from known variables that help to know the influential factors of a project[31]. According to the outcome of the SEM, a manager gets important of a particular task and he can divide the respective weight for each. For example, project status (success, fail, or challenging) is a latent variable while risk factors and resources of the project are measurable those are helping to measure the status. An SEM model also provides the guidelines for hypothesis generation and research question development[32]. The proposed framework will help the expert to develop their own process model by using SEM for a particular project. SEM is popular in social science and commonly used in science[33], commerce[34], and many other areas but sometimes it might be biased. Figure 1 shows the comparison between two influential factors and SEM weight should be given by experts based on the nature of a project. (e.g. a financial transaction system has higher importance for security and not streaming video quality, but a video conference system gives more importance in video/audio quality than security.)

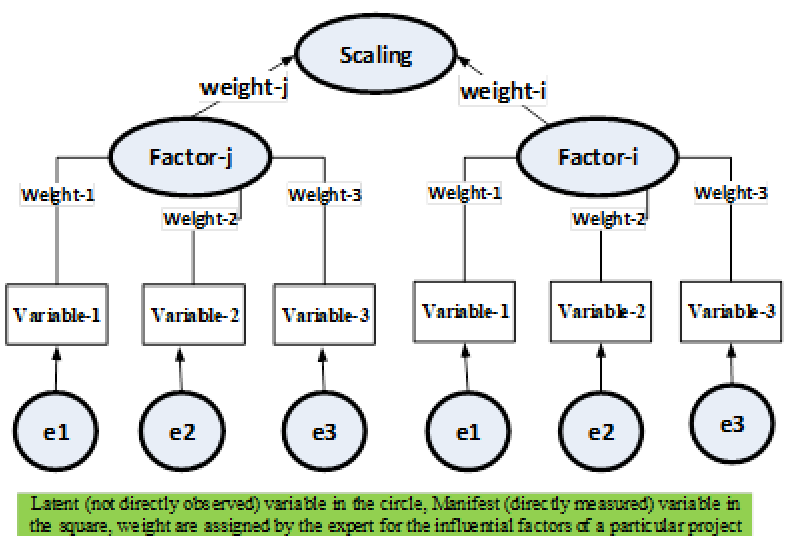


Fig 1. SEM to priorities software quality model

3.6 Grey Rational Analysis (GRA)

GRA is one of the popular approaches in multi-attribute decision-making theory. It is facilitated to combine with fuzzy logic [35]. It is used to modeling the concept and constructed ideas [36]. Filtering features for optimal outcome identification is one of the most important advantages of this method [37]. This paper has applied this technique as a tool for evaluating the framework with few methodologies and ranked based on multiple criteria because it can be implemented for rating, group decision, and risk endurance [38]. This research considered the formula $r = \sum[w(k)\xi(k)$ for grey ranking [40], where w(k) is the weight for comparison factors and this research considers all are equal. $\xi(k)$ is a grey rational coefficient that is calculated by equation (1).

$$\xi_i(k) = \frac{\Delta \min + p\Delta \max}{\Delta x_i(k) + p\Delta \max} \tag{1}$$

Where p=0.5 that is a distinguished coefficient has applied for this calculation. $\Delta x_i(k) = \left|x_{\min(k)-x_i(k)}\right|$ as well as $\triangle min$ *and* $\triangle max$ are the respective minimum and maximum deviation sequences and calculated by equation (2). This is generated from the normalized table, where the normalized table has developed by equations(3) and (4) for benefit and defect respectively.

$$x_i(k) = \frac{|x_i(k) - x_0(k)|}{\max x_i(k) - x_0(k)} \tag{2}$$

$$x_i(k) = \frac{x_i(k) - \min x_i(k)}{\max x_i(k) - \min x_i(k)} \tag{3}$$

$$x_i(k) = \frac{|x_i(k) - x_0(k)|}{\max x_i(k) - x_0(k)} \tag{4}$$

Methodology evaluation variables and GRA: The independent variables $x_1(k), x_2(k), x_3(k), \ldots\ldots\ldots x_6(k)$ is the outcome for each response of the experts in six factors F1, F2, F3,….., F6 (section 5), each factor is scored with the average score of experts that is a number from 0 to 3, or 4 (section 5). The normalized value for dependent variables (F1, F2, F3, F4, F5) for each factor by $x_1(k)$ is calculated by equation (3) because it is calculating the benefits of the methodology where the maximum score indicates better performance. On the other side, F6 represents the complexity of the methodology that is calculated by equation (4) where higher value means fewer advantages. The weight for each average score is considered an equal priority and the distinguished constant is considered 0.5 (standard grey rational co-efficient).

Table 1. Initial hypothesis and evaluation criteria for software project management

Item	Description
Hypothesis-1 Sources Key findings Proposal	**There is a causal relationship between processes, tasks, methodologies, and stakeholders.** Project Management Institute [7] and literature [27]. Around 50% of projects are till now challenging, 20% failed, and 30% is success according to the Standish Group report [4]. (resolution of section-1) A framework that brings processes, tasks, methodologies, and stakeholders together.
Hypothesis-2 Sources Key findings Proposal	**A software project is highly influenced by resources (time, cost, technology, and human)** Literature with analysis of public repository software project data [5] [6] A project may fail if initiated without proper estimation of time, cost, technical resources, and human resources. Explicit specification project and justified allocation resources. Ikigai is proposed to measure the organization's capability
Hypothesis-3 Sources Key findings Proposal	**A Project manager plays a vital role to make a project success** Project Management Institute [7] Individual, technical, and managerial quality is required to deal with projects effectively [7]. An appropriate project management approach can guide to accomplish tasks on time.
Hypothesis-4 Sources Key findings Proposal	**Sustainability becomes one of the most important factors in software quality that enhance business goals.** Standardization Organizations (ISO, IEEE, and CMMI) as well as literature [29] The re-usability of soft resources can improve sustainability practice in software industries and organizations can achieve business goals as well as enjoy competitive advantages. Standard documentation (Standard format, ambiguity free, sharable): descriptive logic and ontological presentation.
Hypothesis-5 Sources Key findings Proposal	**Effective and formal stakeholders' involvement can minimize uncertainty in a project.** It is directly blameworthiness on the group performance instead of an individual act [27] [28] Formal practice improves the responsibility, liability, and performance of the organization. A formal methodology that inspires formal policy, procedure, and documentation.

4 Proposed Framework

Our proposed framework includes a mathematical model, computing algorithms, and a statistical model that is known as, Structural Equation Model (SEM)[39]. Figure 2 represents the generalized framework that is partitioned into three major categories: Pre-project, Inter-project, and post-project. In the pre-project stage, the team will understand the influential factors of the project by the SEM, and they will make the decisions by using Ikigai. Inter-project ontology has been developed for design by Descriptive Logic (DL) and it is proposed for converting to automated ontology if the company needs for distributed project management environment. DL presentation enhances a logical specification facility and that is useful for the developing phase because its' logical presentation can easily convert to computational coding. Post-project activities like the reuse of code and design, upgrading, and maintaining would be easier. Moreover, it provides short but informative documentation that can easily retrievable for a technical person. Algorithm 4.0 decomposed the subsections of a project.

Algorithm 4.0

START

1. *Understand [requirementsOf: user, expert, client] & Set influential factor by SEM*
2. *Generate [ontology, involve stakeholders]*
3. *Revise 1:2 until user satisfaction*
4. *Generate and Apply [ikigi]→ Measures (Goals)*
5. *if (goal not positive) Stop Project else*
6. *Apply ontology and generate [methodology, timeline, policy]*
7. *Sign agreement [stakeholders]*
8. *Follow [timeline, methodology, ontology]*
9. *Continue until end*

END

4.1 Pre-Project activity

SEM (Figure 1) is influenced by risk, cost, time, quality, and stakeholders. The weight of a factor is varied from another one based on the nature of a project and it is assigned by the expert. The research is considering the conventional symbol and X=t+e formula used to verify the model where X is the measured score, t for the true score, and e for error.

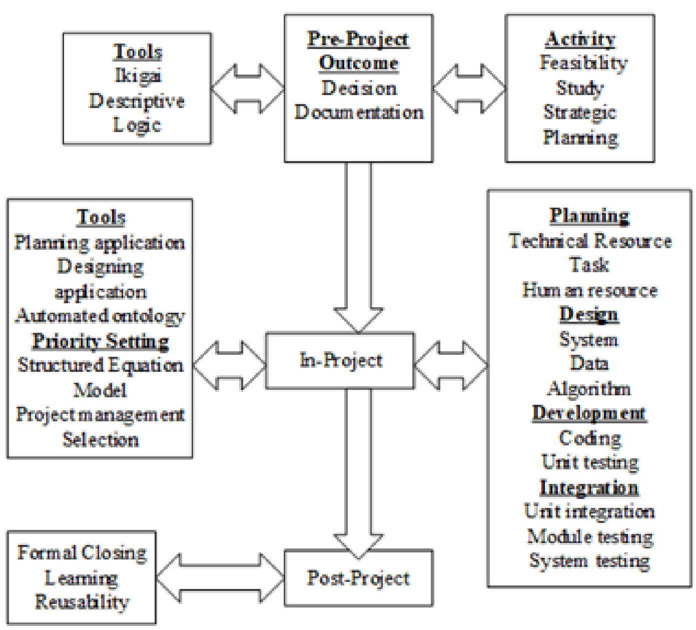


Fig 2. Project management framework

The SEM provides the rank for influential factors of a project that used to generate ikigai. An ikigai model consists of the information of the organization, product features, and external entities (Figure 3). It performs a comparative study for required domains and remarks on four major decisions to a final decision (start or stop the execution). That means SEM and ikigai model performs a feasibility study that is organized and scientific. A project's capability, possibility, and threats are measurable as well as manageable. The combination of both tools has developed a formal approach in the project initiation phase with a systematic, visual, and logic presentation. The four-circle ikigai model (Figure 3) has reflected four different matters of the subject. The intersection of the two circles has delivered a common measure and finally, four measures together generate output known as Ikigai. Figure 3 a generalized ikigai that is derived by the truth table (table 4.1 a) for if any outcome is negative the decision is negative; if all outcomes are neutral then the result is neutral and review the ikigai; otherwise, the result is positive.

Fig 3. Ikigai for a feasibility study

Ikigai model generates an ikigai (decision) that fully depends on the assigned parameters. This research suggests collecting data for domain experts. Logic is also important to apply weight and experience people can suggest only this. The outcome of the circles is generated by the SEM. Table 4.1 (b) explains the conceptual description of the feasibility study and the way of developing ikigai. It also proposes to develop a variable number of ikigai if the project team wan. It is also possible to develop nested ikigai for extracting deeper decision. This activity could be automated by application where a project team can get feasibility study documentation easily and it will cut work for the next projects because it is re-usable.

This work has developed the framework in 3 steps with a causal relationship among the steps. This framework is a guide to generate an ontology for a particular project. A system has input, process, and output that are active with logical data flow and dependency-relationship. The product's quality is a successful outcome of all units of the system and a perfect combination of technology, process, and effort. Actually, the unlimited expectation and performance have no boundary so an optimal solution is acceptable. The expected performance of a system also depends on the characteristics of the software, nature of the software, and demands.

From a set of alternatives, the optimal choice selection is identified by ikigai too (Figure 4). The group of quality factors has been considered for certain circumstances and it varied based on the requirements. The test case dependency is the matter of the processes, stakeholders'satisfaction, and quality factors and can be model with ikigai too. The ontology and prototype are developed in the next section based on this classification. This specification has brought a measuring facility by functional point. Rule generating, building class hierarchy, and sibling are possibly by HAS-A, and IS-A relationship. Moreover, metadata can be implemented for a knowledge domain. On-demand resource allocation becomes more cost-effective with it and it will reduce technical debate.

Table 2. Decision table

ikigai	Measure yourself	Skill versus requirements	Select better options	Business goals
Positive	Positive	Positive	Positive	Positive
Positive	Positive	Neutral	Positive	Positive
Negative	Positive	Neutral	Negative	Positive

Table 3. Requirement specification and the measures for a feasibility study

Ontology
Section-A: Requirement (Functional, Non-functional); Functional{ Client/User Requirements}; Non-Functional{additional stakeholders requirements ∪ Quality factors, social, legal and ethical issues}; Policy Requirements: Sustainability ={ Reusability of design, code; Green Concern etc.}
Section-B: Measuring Self(technology, time, investment, skill or manpower, other resources)
Section-C:
Algorithms:
Develop Set of all requirements like section-A
Priority setup and conflict resolving of quality factor (High-quality visualization ⊥ Green concern) to simplify the requirements
Measure self like section-B
Apply ikigai (Figure 3) to know the strength of the organization
Apply ikigai (Figure 3) to and develop multiple scopes on project
Apply ikigai (Figure 3) to and get a better scope on the business domain
Take the decision according to the truth table (table 4.1 a)
Finally ikigai=initiate or stop the project

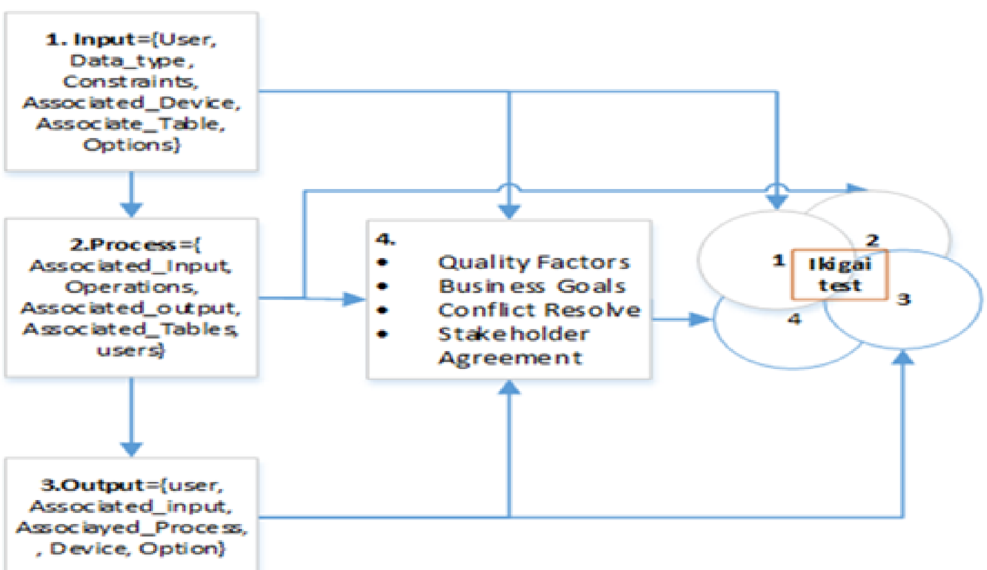


Fig 4. System requirement specification ontology

4.2 Inter-Project activity

The System specification ontology has been demonstrated by HAS-A and IS-A relationship (Figure 5) based on the functional requirements, quality factors, and association relationship. Only input, process, stakeholders, and output are mentioned here. The finding of relations is represented by controlled language: <input field: user-ID has Properties :{string {character, digit, symbol}, not more than 6, associated with {user, keyboard, user-table}}>; <stakeholders: internal {name and responsibility}; external {name and responsibility}>; <process: has sub-process {activity, order and associated with the database, rules}> ;<output: has properties :{quality factors, structure, associated with the process, input, rules and database table}>

The complexity of an information system is increasing with the dimensionality of incoming data sources, relationships, information redundancy, and inconsistency. Controlled language is one of the best methods to represent software information. The uniform structure of information minimizes confusion among stakeholders. Objective-based achievement of a project[40] will be easy to map business functions to software methods specifications by IS-A and HAS-A relation. And ambiguity reduction by structured language would enhance team performance.

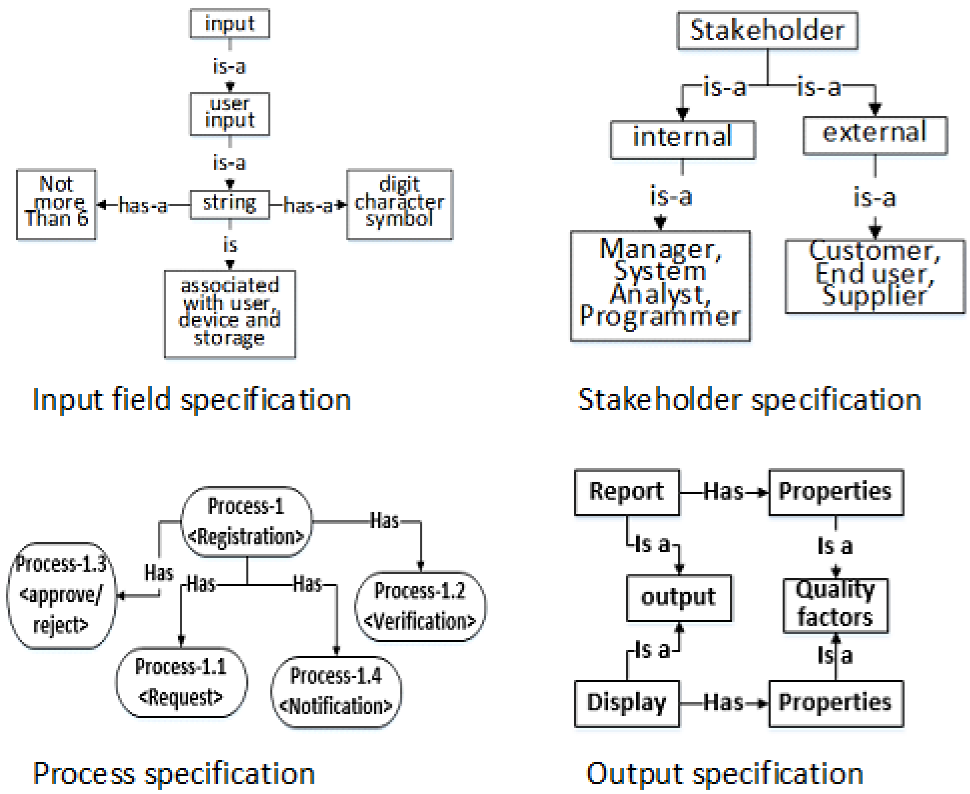


Fig 5. Project information specification

Further relationship of the process to process has presented by tree formation which is modeled with a descriptive logic approach[41]. For the traditional meaning of the general terminologies are used in the descriptive logic presentation. Stakeholders (x, y, z): x, y, and z are an individual person who has a contribution to the project.

- InternalStakeholder (x, y): x and y are stakeholders belong to the organization
- ExternalStakeholders(x, y): x and y are the stakeholders from associate organizations
- SystemProcess(p1,p2,p3): System has process p1,p2 and p3
- ComprisesOperation(P,pi): Process P has process pi
- PropertiesInput(I, pj): Input I has properties pj
- PropertiesOutput(O, pk): Output O has properties pk
- ProcessAccosiatedWith(Pr,Ir,Or): Pr associated with Ir and Or

If t (P) is the specification of process P, C is the concept, and R is the role with a level of process li (i=0,1,2,3,.......) represents highest to the lowest level of Pi (i=0,1,2,3,.......) respectively. Technical modeling for design forwarded to the development team by using descriptive logic presentation for details to apply logical operation and it has mentioned in table 4.2.

Table 4. Descriptive logic ontology

Descriptive Logical Terminology	Description
$\tau(P)$ = ProcessHasSomeValue $R(\tau l)$ when P = li exists $(\mathrm{R}, \mathrm{I_j})\, \mathrm{I_j} : \mathrm{C}$	Process hierarchy dependency: The concept of $_{jth}$ level provides value to the i^{th} level by generating a role in i^{th} level.
$\tau(P)$ = SubClass$Of\,(\tau(C_1)\,\tau(C_2))$ when $P = I_i$: subset $(I_j, I_k)\, I_j : C_1 I_k : C_2$	Class inheritance properties: when one class is generated for another two classes they use their concepts in subclass too.
t(P)=IntersectionProperties($\tau(C_1)$ $\tau(C_2)$) when P= li : and(l_j,lk) $l_j : C_1$ $l_k : C_2$ =($\tau(C_1) \cap (C_2)$)	Common properties of an object: use AND operation to get intersection properties of objects.
t(P)=combineProperties($\tau(C_1)$ $\tau(C_2)$) when P= li : and(l_j,lk) $l_j : C_1$ $l_k : C_2$ =($\tau(C_1) \cup (C_2)$)	Combine properties of objects: use OR operation to get combine properties of objects.
t(P)=not(τI)when if $\varphi = l_i$: not(l_j) l_j : C =~(τI	Different Properties: which has different properties at the same level so they are not the same.
t(P)=inv(τI) when if $\varphi = l_i$: inv(l_j) l_j : C =(τI$^-$	Opposite properties: when one operation is the inverse operation to others. E.g t(redo)=inv(τ(undo))
t(User_Type) ≡τ(End_User)	Equivalence properties: Only end-user can provide the input for such filed
InputData (String: <a/A-z/Z ∧ 0-9>) ∧ String Length<=7	Field Specification: input string in the combination of digit and char but size, not more than 7.
t(Input_Device)≡ t(Keyboard) where Quality_Factors→Usability(Visualization) ∧Accessibility (Shortcut key)	Input role: Input is accepted from a keyboard with usability and accessibility quality
BelongT→user_table/classPurposesOf→Registration ; ∧Login∧edit∧search	This filed only in the user table/class and purposes registration, login, edit, and search process has access to this filed.

The explicit specification of concept, roles, relationship, axioms, and transformation could be modeling with logical expressions (Table 4.2). The logical expression would be converted to programming language for automated ontology development. An automated ontology could digitally record-keeping and share with respective stakeholders. Improving sustainability practice by reusing design and code would be profitable for an organization. The features of an automated ontology for system specification is snaped with Figure 6. And one organization can easily adapt one automated ontology to multiple projects that would be more effective and convenient. This is a tool to control user requirements and change management too. That can include data, design, requirements, and developing strategies like acceptance policy. Figure 7 has featured the system ontology that is used to manage, control, re-use, and update the functional requirements for design and development. Project management policy and regulation that would be extracted could reduce technical debt. The sample extracted rules and acceptance policies form Figure 7 are: <user_type_A: has access on Pi for Ii and Oi with privilege: read/write/execute >; <Pi can possible to update based on client demand, only prior to starting of Pj>; <in house testing of task P has completed within a short period before starting dependent task> AND <client do testing and approve task P with Pi and Pj belongs to I_j and O_j >; <Technology change is the mutual agreement before starting the task P in prior a time > and so on.

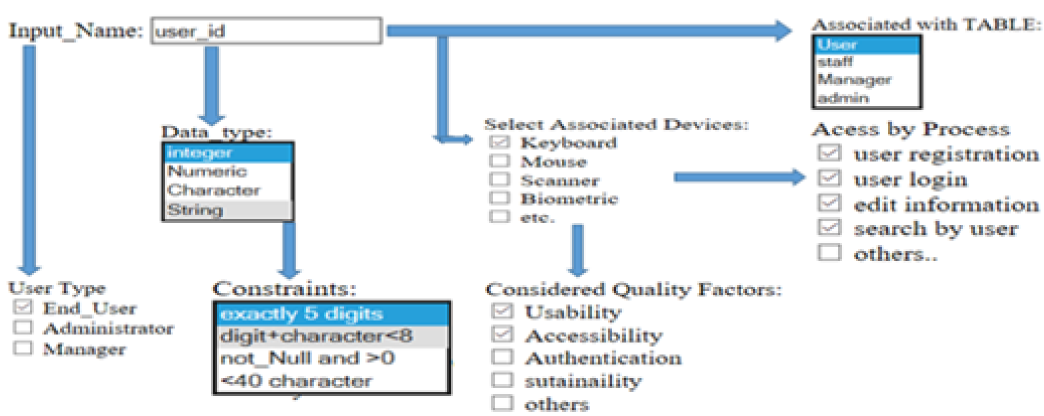


Fig 6. Specification ontology

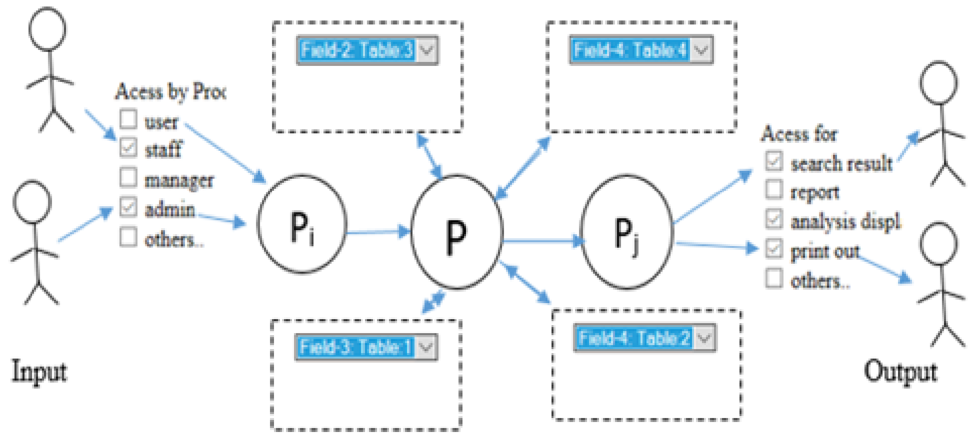


Fig 7. Relationship ontology

5 Discussion and evaluation

The work has integrated the required tasks of a project management team briefly in an ontology within a three-layer framework. The framework is decomposed into multiple subsections with the right tools. The explanation of the methodology is presented with a series of diagrams, logical expressions, and tabular format. It also prototyped for generating automatic ontology that will reduce the information processing complexity of software industries. Moreover, it is developed in such a way that it can make sure stakeholders' involvement, formal communication, standard documentation, and regular tracking. A comparison discussion is given based on the current challenges of project management and the characteristics of near future projects in table 5.1 (a). Figure 8 has details of GRA which is developed among 5 methodologies in 6 different factors. Selected four methodologies are famous in different criteria. The waterfall is a linear traditional methodology, scrum is most important in the Agile family, PRINCE2 is popular for control the project, and spiral are popular for comparatively large and risk-oriented projects. The factors could be scaled like:

F1. Involvement of stakeholders ={3,2,1,0}←{required time with the formal meeting, predefined time with the formal meeting, once formal interaction, no or informal meeting}.

F2. Documentation ={4,3,2,1,0}←{sufficient and standard, sufficient but not standard, insufficient and standard, insufficient and not standard, no record }.

F3. Focusing on the management role={3,2,1}←{always monitoring and in control, regular monitoring and control, certain monitoring, and control}.

F4. Estimation resources and risk={3,2,1}←{scientific measuring approach and experience, measured from experience, nor realistic approach}.

F5. Focusing on the requirement and quality={3,2,1}<{ define, review and update, reviewed and updated, predefined}.

F6. The complexity of the methodology={3,2,1,0}={complex with extra work, complex but no extra work, acceptable complex, simple}

Table 5.1 (a) has shown the way of mitigating current existing reasons by this model and how easily interpret this model for near-future software projects.

A survey was conducted to a few numbers of experts in software industries who are working in different organizations from July to August of 2019. Before the data collection details explanation of the ESF with functionalities and working procedures is given to the respondent. They have managing experience in multiple software projects. Two of them have working exercise with oversize companies by the distributed management system, one has a long time (15 years) freelancing experience for online application development, two are working for banking software development and it is a local company, and rest are working on the local companies who have to work in different types of software projects. All methodologies have pros and cons because each has at least one significant view of development. The influential factors are weighted according to the role in project management based on the aim of the framework and applied SEM approach, so the scale is not the same for all (see the questioner in appendix). The average weight for each factor of a particular method is considered for the input of GRA (Figure 8) followed by reciprocal matrix and grey rank for the p=0.5, which is a distinguished coefficient has applied for this calculation.

Table 5. Framework evaluation

Reasons for Project Failure & Upcoming Challenges	Role of the Proposed Framework
The weakness of process, method, environment, and people [42].	A relationship is developed among process, method, environment, and stakeholders in the framework to eliminate the gaps.
Immature planning, fail to understand complexity, unrealistic budget, weakness on structure and policy [43]	Explicit specification reduces misunderstanding and guides to make realistic estimation and ontology will develop unfair policy.
Wrong methodology and user characteristics. [44]	It ensures the users' interactions, feedback, and formal acceptance as well as itself an explicit methodology.
Lack of information standardization and sharing: CMMI, ISO, IEEE and SWEBOK [45]	Consistent and standard documentation is managed and shared by automated ontology among stakeholders.
Harmonious development depends on the common backbone structure [46]	The ontology of the framework is linked to all components of the project phases.
ISO-9001 Management, ISO-9126 quality factors and CMMI levels ([1–5]	This includes quality factors as a requirement and highlighting sustainability too.
Defect prediction [47]	A structural scientific method proposed for a feasibility study with ikigai and SEM.
Upcoming software will be highly dependent on AI. [48]	Explicitly specification is done by descriptive logic meets the demand for AI.
Introducing smart application and automation in industries [49]	AI is the core of the smart applications that required rules and information specifications is provided by an ontological presentation.
Within 2030 business, client interactive business concept [50]	This approach can easily be implemented for decision-making applications because of the supporting AI and smart applications.
According to Maher ZA et al [51], the performance of the software industry is influenced by the adoption of the stages of software development, role of authorities, quality of motivation, management and attitude, security, culture, change management. In addition, it also proposed actions to overcome such as hiring domain experts (manager, technical, tanning), showing importance in awareness to ensure security.	The framework shows the importance of tools and techniques together to adopt phases with tasks, processes, methods, and tools. Descriptive documentation will reduce ambiguously documentation and logical presentation would improve sustainability that could enhance business goals. Formal culture responsiveness among the stakeholders and fairness in the company. Ontological-representation of explicit specification information easy to convert into an automatic management system. Role, policy, and procedure could be extracted, updated, and implemented from the automatic project management system if implemented according to this framework.

	F1	F2	F3	F4	F5	F6
Waterfall	1	3	2	2	2	1
Scrum	2	2	3	2	2	2
PRINCE2	3	3	3	2	3	2
Spiral	3	2	2	2	3	2
ESF	3	4	3	3	3	3

(a) Grey evaluation matrix

	F1	F2	F3	F4	F5	F6
Waterfall	0	1	0	0	0	1
Scrum	1	0	1	0	0	1
PRINCE2	1	1	1	0	1	1
Spiral	1	0	0	0	1	1
ESF	1	1	1	1	1	0

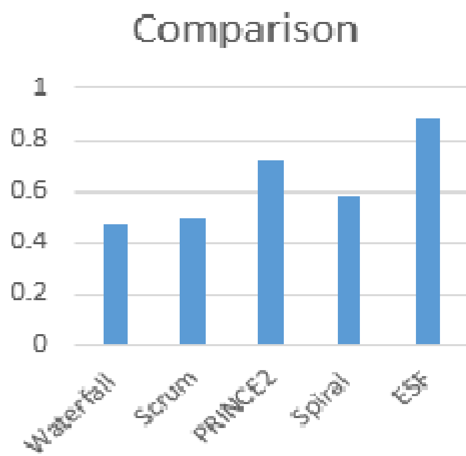
(b) Grey Normalized matrix

	F1	F2	F3	F4	F5	F6
Waterfall	0.33	0.5	0.33	0.33	0.33	1
Scrum	0.5	0.33	1	0.33	0.33	0.5
PRINCE2	1	0.5	1	0.33	1	0.5
Spiral	1	0.33	0.33	0.33	1	0.5
ESF	1	1	1	1	1	0.33

(c) Grey Co-efficient

Waterfall	0.4722222
Scrum	0.5
PRINCE2	0.7222222
Spiral	0.5833333
ESF	0.8888889

(d) Grey Rank



(e) Grey Rank

Fig 8. Grey comparison Study

6 Conclusion and Future Work

The framework consists of workflow and corresponding methods to manage that task, suggests using specific effective tools, motivates to consider quality factors, and shows the way of formal practice and record management. It explained issues, importance, and ways of management. It encourages to apply standard applications for distributed project management. Proposed actions, processes, and rules would improve formal practice. Standard documentation deduces ambiguity and technical debate that improves managerial performance of management. Soft-documentation could be easily re-used for further projects and improves sustainability in project management. It assumes that this an overwhelmed framework with sets of tools, techniques, and processes. It increases the workload at the initiation phase of a project but it is only for the first project because later all projects can get access to the same systems with a bit modification. It looks that team members need more knowledge for mathematics and statistical tools but the applied terminologies are very basic for computer science background people. Logic terminologies are comparatively critical than generic language explanations but it is proposed to cut the language barrier for distance mode of management too. Technical documentation, information specification, and representation is a common skill in IT industries too. So descriptive expression could be easily implemented by a standard organization. Recommended automatic ontology tool implementation in an industry could improve its efficiency and sustainability. The business goal achievement would be easier. This article enhances the scope to work in automated ontology development for project management. So Explicit Specification Framework (ESF) is recommended for the standard software industry that has a mission and vision with specific long-term business goals. ESF does not ask for certification from any organization but it encourages adjusting guidelines according to the standardization organizations. ESF is a flexible framework and standard company can adjust according to their ability, business goal, and stakeholders' guidelines.

The comparison study showed better performance for several criteria such as formal management practice, standard documentation, scientific method of information specification and justification, integration quality factors, and keeping control of a project. It has complexity value which is higher than other methodologies for giving more emphasis on the project initiation phase. But it will decrease the workload of subsequent phases because the framework believes on "you cannot control what you cannot measure"[52], "you cannot measure what you cannot define"[53], and "you cannot define what you don't know"[53].

Appendix

Disclaimer

This data only for research purposes and never be disclosed to thirty parties or used for commercial purposes. There are two sections of data collection form and the first part for respondent information that will not attend for research analysis. Only data of section-2 will be used for analysis.

Section-1 Respondent Information	
Name:	Position and organization:
Experience in software development(years):	Experience in software project management: (years):
Email:	Telephone (optional):
Which methodologies using are using?	
Average number of member working in a project: The average duration of a project: Do you work in an international working place? (Yes/no)………………… Years: Do you have working experience in freelancing? (Yes/no)…………………Years:	

Section-2

For each question, there is a range (0,1,2,3,4) of ranking where 4>3>2>1>0. (Example, 3 is a higher rank than 2): You have to mark a number from for the respective methodology;

F1. Involvement of stakeholders ={3,2,1,0}← {required time with the formal meeting, predefined time with the formal meeting, once formal interaction, no or informal meeting}.				
Waterfall	Scrum	PRINCE2	SPIRAL	ESF
0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
F2. Documentation ={4,3,2,1,0}← {sufficient and standard, sufficient but not standard, insufficient and standard, insufficient and not standard, no record }.				
Waterfall	Scrum	PRINCE2	SPIRAL	ESF
0 1 2 3 4	0 1 2 3 4	0 1 2 3 4	0 1 2 3 4	0 1 2 3 4
F3. Focusing on the management role= {3,2,1}← {always monitoring and in control, regular monitoring and control, certain monitoring, and control}.				
Waterfall	Scrum	PRINCE2	SPIRAL	ESF
1 2 3	1 2 3	1 2 3	1 2 3	1 2 3
F4. Estimation resources and risk={3,2,1}← {scientific measuring approach and experience, measured from experience, nor realistic approach}.				
Waterfall	Scrum	PRINCE2	SPIRAL	ESF
1 2 3	1 2 3	1 2 3	1 2 3	1 2 3
F5. Focusing on the requirement and quality={3,2,1}<{ define, review and update, reviewed and updated, predefined}.				
Waterfall	Scrum	PRINCE2	SPIRAL	ESF
1 2 3	1 2 3	1 2 3	1 2 3	1 2 3
F6. The complexity of the methodology={3,2,1,0}={complex with extra work, complex but no extra work, acceptable complex, simple}				
Waterfall	Scrum	PRINCE2	SPIRAL	ESF
0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3

References

1) Laplante P. What Every Engineer Should Know about Software Engineering. Boca Raton. 2007.
2) Software Engineering Body of Knowledge (SWEBOK Version 3), 2014" (pdf). www.swebok.org. *IEEE Computer Society Retrieved*. 2016.
3) Abran A, Moore JW, Bourque P, Dupuis R, Tripp LL. Guide to the Software Engineering Body of Knowledge. IEEE. 2004.
4) The Standish Group International. CHAOS Report. 2015. Available from: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.
5) 2017 HKMH, Dataset. Dataset: Evidence-based Software Portfolio Management (EBSPM) Research Repository. TU Delft. 2017. Available from: https://doi.org/10.4121/uuid:42fd1be1-325f-47a4ba39-31af35ca7f75.
6) Sarker KU, Deraman AB, Hasan R. Descriptive Logic for Software Engineering Ontology: Aspect Software Quality Control. In: 4th International Conference on Computer and Information Sciences (ICCOINS). 2018;p. 1–5.
7) Bannerman PL. Defining project success: a multilevel framework. Paper presented at PMI® Research Conference: Defining the Future of Project Management, Warsaw, Poland. Warsaw, Poland. Newtown Square, PA. Project Management Institute. 2008. Available from: https://www.pmi.org/learning/library/defining-project-success-multilevel-framework-7096.
8) Gruber T. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Int J Hum Comp Stud*. 1995;43:907–928.
9) Uschold M, Gruninger M. Ontologies: principles, methods and applications. *The Knowledge Engineering Review*. 1996;11(2):93–136. doi:10.1017/s0269888900007797.
10) Gomez-Perez A, Corcho O. Ontology languages for the Semantic Web. *IEEE Intelligent Systems*. 2002;17(1):54–60. doi:10.1109/5254.988453.
11) OWL 2 web ontology language. 2017.
12) Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*. 2000;25(1):25–29. doi:10.1038/75556.
13) Jonquet C. AgroPortal: A vocabulary and ontology repository for agronomy. *Computers and Electronics in Agriculture*. 2018;144:126–143.
14) Abburu S, Golla SB. Ontology-Driven Knowledge-Based Health-Care System an Emerging Area- Challenges and Opportunities - Indian Scenario. The International Archives of the Photogrammetry. In: ISPRS Technical Commission VIII Symposium. 2014;p. 9–12.
15) Fluit C, Sabou M, Harmelen FV. Ontology-based information visualization: toward semantic web applications. Springer. 2006.
16) Pickert G. Einfuhrung in Ontologien. 2016. Available from: http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0203/SemWeb/artikel/2/PickertOntologienfinal.
17) Cherix D, Usbeck R, Both A, Lehmann J, Presutti V, Blomqvist E, et al. Lessons learned - the case of CROCUS: cluster-based ontology data cleansing. In: Papadakis, I, Tordai, A, editors. The Semantic Web: ESWC 2014 Satellite Events;vol. 8798. Springer. 2014;p. 14–24.
18) Sarker KU, Deraman A. Design Aspects of Near Future Soft Computing. *International Journal of Electronics Communication and Computer Engineering*. 2017;8(5):2249–071.
19) Sarker DKU, Deraman RA, Hasan. Kids' Smart Campus Ontology to Retrieve Interest. In: Green Soft Computing, Conference: ICEL3S2018. 2018;p. 462–470.
20) Sarker KU, Deraman AB, Hasan R, Mahmood S, Abbas A, Sohail M. Kids' Smart Campus Ontology to Retrieve Interest. In: 4th MEC International Conference on Big Data and Smart City (ICBDSC). 2019;p. 1–4.
21) Kamiya M. Ikigai-ni-tsuite. Tokyo, Japan. MisuzuShyobou. 1966.
22) Kumano M. Ikigai-keisei-no-shinrigaku [A psychology of ikigai development. KazamaShyobou, editor;Tokyo, Japan. 2012.
23) Kumano M. Ikigai-keisei-moderu-no-sokutei-shyakudo-no-sakusei: Ikigai-purosesu shyakudo-to-ikigai-jyoutai-shyakudo [Construction of scales for the ikigai development model: The scales for ikigai processes and ikigai states. *The Bulletin of Education*. 2013;39:1–11.
24) Schwalbe K. Project management approach to IT projects. Tehran, Institute of. Tehran University Press. 2010.
25) Martin P, Tate K. Project management, translated by Mohsen Zokayee. Tehran. Adineh press. 2003.
26) Tilmann G, Weinberger J. Technology never fails, but project can. *Baseline*. 2004;1(26):28. doi:10.1515/gpr.2004.1.4.182.
27) Klein HK, Meyers MD. *MIS Quarterly*. 1999;23:67–94.
28) Mandal A, C S. Identifying the Reasons for Software Project Failure and Some of their Proposed Remedial through BRIDGE Process Models. *International Journal of Computer Sciences and Engineering*. 2015;(1).
29) O'Neill RP, Dautel T, Krall E. Recent ISO Software Enhancements of the Last Decades and Future Software And Modeling Plans. 2011. Available from: https://www.ferc.gov/industries/electric/indus-act/rto/rto-iso-soft-2011.pdf.
30) Bergel A, Denier S, Ducasse S, Laval J, Bellinguard F, Vaillergues P, et al. SQUALE - Software QUALity Enhancement. In: 13th European Conference on Software Maintenance and Reengineering. 2009;p. 285–288.
31) Kline R. Principles and Practice of Structural Equation Modeling. *Guilford*. 2011;p. 978–979.
32) Salkind NJ. Intelligence Tests". Encyclopedia of Measurement and Statistics. 2007. doi:10.4135/9781412952644.n220.
33) Boslaugh S, Mcnutt LA. Structural Equation Modeling". Encyclopedia of Epidemiology. 2008. doi:10.4135/9781412953948.n443.
34) Shelley MC. Structural Equation Modeling". Encyclopedia of Educational Leadership and Administration. 2006. doi:10.4135/9781412939584.n544.
35) Zha XF, Sriram RD, Fernandez MG, Mistree F. Knowledge-intensive collaborative decision support for design processes: A hybrid decision support model and agent. *Computers in Industry*. 2008;59(9):905–922. doi:10.1016/j.compind.2008.07.009.
36) Sahin Z, Yenmez AA, Erbas AK. Relational Understanding of the Derivative Concept through Mathematical Modeling: A Case Study. *EURASIA Journal of Mathematics, Science and Technology Education*. 2015;11(1):177–188. doi:10.12973/eurasia.2015.1149a.
37) Hsu MW. An Analysis of Intention to Use in Innovative Product Development Model through TAM Model. *Eurasia Journal of Mathematics*. 2016;12(3):487–501.
38) Zhong XM, Fan KK. A New Perspective on Design Education: A "Creative Production Manufacturing Model" in "The Maker Movement" Context. *Eurasia Journal of Mathematics*. 2016;12(5):1389–1398.
39) Kaplan D. Structural Equation Modeling: Foundations and Extensions . 2008.
40) Yassien E. Software Projects Success by Objectives. *Journal of Management Research*. 2017;10(1):46–46. doi:10.5296/jmr.v10i1.10149.
41) Brachman RJ, Mcguinness DL, Patel-Schneider PF, Resnick LA, Borgida A. Living with CLASSIC: When and how to use KL-ONE-like language. In: Sowa JF, editor. Principles of Semantic Networks. Morgan Kaufmann. 1991;p. 401–456.
42) Lehtinen TOA, Mäntylä MV, Vanhanen J, Itkonen J, Lassenius C. Perceived causes of software project failures – An analysis of their relationships. *Information and Software Technology*. 2014;56(6):623–643. doi:10.1016/j.infsof.2014.01.015.

43) Verner JM, Abdullah LM. Exploratory case study research: Outsourced project failure. *Information and Software Technology*. 2012;54(8):866–886. doi:10.1016/j.infsof.2011.11.001.

44) G W. Success Rate by Software Development Methodology in Information Technology Project Management: A Quantitive Analysis. vol. 3590342. 2013.

45) Bajnaid NO, Benlamri R, Pakstas A, Salekzamankhani S. An Ontological Approach to Model Software Quality Assurance Knowledge Domain. *Lecture Notes on Software Engineering*. 2016;4(3).

46) Pardo C, Pino FJ, Garcia F, Baldassarre MT, Piattini M. From chaos to the systematic harmonization of multiple reference models: A harmonization framework applied in two case studies. *Journal of Systems and Software*. 2013;86(1):125–143. doi:10.1016/j.jss.2012.07.072.

47) Mende T, Koschke R. Effort-Aware Defect Prediction Models. *Proceedings of the 2010 14th European Conference on Software Maintenance and Reengineering CSMR'10*. 2010;p. 107–116.

48) Peyton T. The Future of Work Jobs and Skills in 2030. 2014. Available from: https://www.gov.uk/government/publications/jobs-and-skills-in-2030.

49) Kearney AT, A. Technology and Innovation for the Future of Production: Accelerating Value Creation. 2017. Available from: http://www3.weforum.org/docs/WEF_White_Paper_Technology_Innovation_Future_of_Production_2017.pdf.

50) Kjaer, Bruel. Beyond tomorrow Scenarios 2030: The future of product development, Copenhagen Institute for Futures Studies. A/S Skodsborgvej 307 DK-2850 Nærum Denmark. Sound & Vibration Measurement. 2017. Available from: http://beyondtomorrow.dk/wp-content/uploads/2017/11/Beyond-Tomorrow-Scenarios-2030-full-report.pdf.

51) ZA M, A S, S C, HM M, NHBA R. Challenges and limitations in secure software development adoption-A qualitative analysis in Malaysian software industry prospect. *Indian Journal of Science and Technology*. 2020;13(26). doi:10.17485/IJST/v13i26.848.

52) T D. Controlling Software Projects: Management, Measurement & Estimation. Yourdon Press. 1982.

53) Buglione L. Some thoughts on quality models: evolution and perspectives. *Acta IMEKO*. 2015;4(3).