Design Aspects of Near Future Soft Computing

Kamal Uddin Sarkar* and Dr. Aziz Deraman

*Corresponding author email id: ksarker@mcc.edu.om

Date of publication (dd/mm/yyyy): 26/10/2017

Abstract – Exponential increasing rate of internet users, progress of smart cities and cloud based applications are trending to big data operations and challenges for green world. An efficient and formal approach of a soft computing body can overcome the upcoming challenges. The study presented latest considerable issues of an efficient software developing process for near future smart world. Selection of a Software Development Life Cycle (SDLC), specifying approach for a software development project and recommendation for future characteristics of software are presented respectively. Overall, study is developed for better understanding on ontology, DSLC and requirements for the software of next decade and quality control. Developed bridge among four components is generated for running a successful soft computing firm in sustainable energy consuming world.

Keywords – SDLC, Ontology, Software for Near Future, Green Computing, Software Quality Control.

I. INTRODUCTION

Software reusability and updating issues become more challenges day by day due to rapidly upgrading technologies, public demand changing, portability features, increasing rate of users and social concern on green technology. Today’s software can survive in next time or how can it will update interface, and can be more environment friendly including usability and accessibility need to consider during the time of requirement analysis. Clients can recognize what is required for present situation but developing team need to predict what will be happened in near future, so guidance to client is required [fig-1]. Discussion mainly focused on four major parts of software developing process in the following four sections of the article: SDLC, ontology in soft computing, features of future software and followed by quality control in the software developing process. Approaching automation system which will be controlled from internet devices or from the physical variables by using sensors, further decision making from cloud data analysis will be common characteristics of the software. A heterogeneous dimensionality and requirement changing behavior demanded for flexible design tactic. So view of these intentions, we have suggested flexible requirement supported with complex functionality of SDLC for large projects. During the period of updating or reusing well organized documentation with graphical representation is more useful and efficient, and hence ontology is recommended. Our concern on green ICT is absorbed too besides ensuring quality control of software. Reducing software failure is aimed from the guidelines of the research.

II. RIGHT CHOICE OF METHODOLOGY

Software companies traditionally follow methodologies for successfully developing a system known as Software Development Life Cycle (SDLC). Researchers approach different SDLCs and one is not suitable for all applications; fixing a methodology depends on the size, duration, time, complexity, risk and user requirements. Some special characteristics are observed from the requirements of user in this time: rapidly changing mentality with the progress of technology. Special requirement loop is introduced [fig-1] including some traditional phases like: requirement analysis, Planning, designing, coding and implementation, testing, deployment and maintenance. Also most of the cases sequences are same too; while one methodology combine 2 steps in a single step, others encourages to splitting more stages on demand. Main differences are observed loopback facility in the stages as well as version flexibility. A classical approach is waterfall model [28] and extended modified waterfall model [29] to ensure successful project and it is not fixed for any special software. Prototyping uses to identification component and specifying layout oriented software like web applications within clients, RAID is suitable for small and short time project having less complexity and budget. On the other side if anyone wants to start partial working on ongoing project can chose incremental model. A combination approach for waterfall and Prototyping with risk analysis is spiral model [28] [29]. A methodology focusing on coding for the rapidly requirement changing software project named Extreme Programming (EP) model while Scrum Model (SM) focuses on interactive for that kind of software [30]. We expected development requirement ontology for fixing all requirements according to the following flow diagram [Fig: 1] and followed by the other ontologies from the selected SDLC.

III. FORMAL SOFTWARE DEVELOPING APPROACH

Ontology is a formal design approach for software development with multiple dimensions in multiple subject areas and domains. There are plenty of definitions available based on the area or domain for ontology. While someone developed definition for the general view others preferred domain oriented definitions. Ontology specified by shared understanding [1] [2] among agents of the software development life cycle, and it is a systematic science for describing axiomatic theories [3] mentioned by researchers. Moreover, ontology focused for inter - agents communication to make efficient interoperability [6]. View of text mining it is defined by a set theory which used to identify pattern for searching application
development [4], same way it is mentioned to specify vocabulary for shared domain including all functions,

...standardize knowledge representation for web based application is a specific ontology. Digital ontology can be developed for large scale research field or some common software designed tools, but if every individual applications till now working on manual ontology because it is specific for each application. Ontology also designed for partial part of an application such as data characterized (uniqueness, constrains) [11], data cleansing [12], statistical analysis [13], clustering and visualization [14] and knowledge integration [15]. Other researches focused on “requirement engineering and reusability” [16], “reusability and post implementation management” [17], and managing by sharing, planning and domain analysis [18]. Data sharing for agriculture based application [19] specified the users’ data sharing format.

Individual ontology is specified for next generation software to easily adapt flexible user requirements [fig-1], ensure reusability, consideration green computing. And ontology should specify [fig-3]:

- Formal documentation which includes all protocols from the beginning to post implementation activities in a software development project.
- It will specify the domain and standard information format to share among the agents.
- The ontology should be specific, measurable, and realistic for that application.

An effective Planning ontology [35] includes guidelines for all other ontologies like: requirement analysis, design, implementation, testing, maintenance and re-usability scope be well defined and specified individually. Maintaining a developed ontology for the lifetime of the project as well as update for enhances re-usability ensure quality and reduce failure rate. All above concerns are aligned with green computing and that are recommended too.

### IV. SOFT COMPUTING FOR NEAR FUTURE WORLD

The progress of technology and advantage applications in internet rapidly updated the traditional process in corporate, business, service, governance and personal life. Life become faster and competitive in the global village, so minimizing digital gap is a vital issue in developing country. A software developing company needs to do research on upcoming technology and how easily reuse the current resources for future service. A qualitative and quantitative analysis showed how fast technology changes and summarized by near future system will be “Fixed mobile convergence seamless meaningful cross-channel customers experiences” [20]. Moreover, researcher coated “Internet will be like electricity-less visible but more deeply embedded in life for good and ill” [23], it has noticed how it will effect in our social, professional and political life in 2025. Currently, in 2017 we already experienced by social media, according to Paul Jones founder of ibiblio.org and professor at the University of North Carolina “Television let us see the Global Village, but the Internet let us be actual Villagers [22].” People can communicate in very cheaper and without geographical

![Fig. 1. Requirement flow chart for developing requirement ontology](image-url)
boundary. A study [21] showed the specific and measurable prediction of next computing progress and concluded by bitcoin and Blockchain technology in 2027 by progressing, 3D printer, storage and internet for all things, wearable internet, vision in new interface, driverless car, AI and white collar jobs including smart city from this era. Discussion on huge amount of possibilities and recommendations from academicians and professionals on the statistical survey decided that in 2025 the definition of job position should be updated due to the commonly available robots and AI system in personal and professional life [24]. Definitely energy consumption will be a great issue and every system will introduce green technology and software engineers should consider green computation for data structure, algorithm, machine learning [25] for sustainability. Faster growing of internet cloud computing become more popular, reliable as well as essential for near future internet society and load balancing [26] algorithm designed for energy saving.

A- Considerable Issues for Scheming Software

- Internet based automation: Smart city is leading of internet oriented automation system, resulted increases big data and cloud depended software having multimedia input/output and processing facilities. A simple device for today would be smart in next year, so advance design thinking required being success in business goal and ontology approach is recommended to leading quality.

- Green Computing: computation power and smart device users are increasing, consumption of power becoming high rate though huge number of people has no scope of using electricity in the world till now. A cloud environment will be common platform for sharing resources and saving energy, moreover how fast and less computation time make access of cloud is thought of green computing too. Research on reusability of software, efficient algorithm and data structure design additional with resource sharing and other recommendations like other fields; electrical and civil. Research ontology is included [fig-2] for adaptation of new technology and user requirements in flexible way to control the quality of communication among stockholders, service of the company and quality of the product.

V. REDUCE SOFTWARE PROJECT FAILURE RATE AND ENSURE QUALITY CONTROL

Software Quality Control (SQC) is a well-known terminology which was introduced from the beginning of the software engineering or software engineering introduced to ensure quality of the software [31] in 1968. This was aimed to reduce the failure rate of software project. All stakeholders or agents of the software which would going to be develop are responsible to make success the project that means has variety of dimensions including business goal, technical functions as well management procedure. Few general areas we required or demanded to discussion here before going to the main topic such as:

- Meaning of software failure, current failure rate of the software project and what are the reasons behind that failure rate.
- Who are the Stakeholders and how they are responsible to ensure quality of the software project.
- What do we mean by the term “Quality of the Software and what are the measures”.

Software is not only a package of programs which performs a set well-defined tasks and having documentation for installing, maintaining as well as upgrading. Software quality control [36] [37], includes activities before developing (pre-development), during the period of developing (in-development) and after developing (post-development) because software project may fail before starting, during developing and after developing respectively.

A software project totally failure means it does not functionally working for desired goal of the system which will not be used moreover. It is very clear that it will be not ready for running, and project should be closed. But partial failure identified for so many reasons and it can be summarized by two broad areas as mentioned below:

- System accepts appropriate inputs and generates desired output by performing required operations but project time and/ or cost increased.
- System may be has limitations in input, and/ or process and/ or output but it purposes partial work for the organization.

Assigned software development project by the organization to a developing company for their own business goal or for third party, all are parts of agents. Sometimes developing has done in house too. The person who will identify requirements; in details; who will specified input, output and process; who will guide the technical requirement analysis and who will do coding for the systems and so on are the agents in the project developing resource persons. They are not from in a single umbrella based on knowledge, personal weakness or strength or even from different working environment. Estimated time, cost and plan should be reasonable and follow-up accordingly are the functionality of the project.
management team. A lot of scopes may be observed from the historical analysis that weakness of the mutual understanding causes the partial failure. Technical and nontechnical persons mutually write the successful story of a project but technical person mostly responsible for a failure project. Because technical committee is considered expert body and their efficiency measured by the success rate of the projects. So management ontology is in the central of the figure [fig-2] which will be success on the participation of others.

Recently study “Chaos report” [32] mentioned the statistics of software failure rate and summarized that 66% technical project failed partially or totally and it is same for last five years, moreover 17% of large IT projects so low quality that company face threat for self-existence. Another survey from “Innotas” [33] has done in same period and their failure increases 50% to 55% from the year 2013 to 2015 respectively. Both are more than 50% indicates that it is a common phenomenon in software developing project throughout the world; especially it is more for the mega projects.

A software may be fail in any stage of the software development lifecycle but most of the case it is recognized in developing or testing stages and immediate solution given by increasing budget and time; another general perception developed that an extended project need to extend more time. Why do you need to increase project time? Is it for immature planning and estimation, business policy or primary motivation, or any uncertainty happened during the project? Though most of the cases later one mentioned to extend a project but real case is in the first two questions.

A few common reasons can be highlighted as follows:

- Mutual understanding problems on requirements among the agents.
- Immature planning and budgeting.
- Failure to measure scope and risk for resources on technical, economical and manpower.
- Lack of knowledge on the prediction of future technology.

Reluctantly misconception happens on requirement engineering; it is observed that few client cannot express their requirement properly if there is not an existing sample; otherwise they follow one sample and started work on that, after a while time they want to modify by following another new sample. Expertise from service provider can play active role to identify proper requirements for that software also can focus on the features of near future technology including upcoming impacts due to the progress of technology. Hosting company can take easy and best decision and it will diminish risk of requirement changes as well as failure rate of software.

Nostalgic mentality is not suitable in software developing firm, if any company run properly in the field of competitive market need to update their process regularly and accept latest technology for the updated service. Technical knowledge regarding IT application or updating technology is not required for host but developing firm need to make clear idea about the pros and cons of desired application for long term goals and should be inform to the client. More choice option can make more satisfaction to the client and may increase its’ reliability.

A good bridge among the agents is possible in only one condition to follow the plan, modify the plan or execute the plan efficiently. Faster communication and decision making is only possible when the bridge having responsibility, liability and respect to each other with in a formal approach. Lowering cost and time can provide quality product [34], a way to reduce software failure rate. It leads that it is not only depend on the each person or authority but also weak bridge is responsible to make failure the project.

There is no comments against the reasonable time and cost estimation required to ensure successful software project; technical knowledge with alternative all possible solutions can suggest the better scope of successful live project. Moreover, standardizations are recommended like ISO9126.CMMi, ISO9001: 2000 [36] and ISO/IEC 90003 [37] to ensure standard of the product and also appreciated for certification from legal body.

VI. CONCLUSION AND FUTURE WORK

In this article we showed the prediction of near future computing characteristics for software development to satisfy the challenges of green computing, rapidly progress of technology, increasing data and changing platform. Our proposed method is ontology [fig-3] based design can
make a company sustainable, globalization and competitive in the software market. We also recommended for standardizing to ensuring Quality of Service (QoS), by maintaining standardization. This paper opens a free space of research on green computing: designing new data structure, algorithm, load balancing and in cloud so on.

REFERENCES

AUTHORS’ PROFILES

Mr. Kamal Uddin Sarker (ksarker@mec.edu.om) working in Computing Department, Middle East College (www.mec.edu.om), Muscat, Oman. He has research on Software Engineering, Knowledge Based System Design and Data Mining, computational biology and usability and accessibility of website beside 14 years teaching experiences in his field. He is now working on PhD research in the field of software engineering at UMT (Malaysia) and he was awarded BSc in CSE from RUET and Masters in IT from DU, Bangladesh also pgCert in HEP from CU, UK.

Dr. Aziz Deraman (a.d@umt.edu.my), received his Bachelor from UKM in 1982, Master from Glasgow University in 1984 and PhD from University of Manchester Institute of Science and Technology (UMIST) in 1992 and has served National University of Malaysia (UKM) since 1984. He is professor of Software Engineering specializing in software process, management and certification. He has held various academic administrative positions such as head of Computer Science Department (1985-1988), Deputy Dean of IT Faculty (1992-1995), Deputy Director of Computer Centre, UKM (1995-2001) and the Dean of the Faculty of Information Science and Technology (FTSM) (2001-2007). Currently he is the deputy vice chancellor of University Malaysia Terengganu. He also affiliated with various organizations and industries as an adviser, panel as well as a resident consultant in information technology. His research mainly focuses on IT strategic planning, software management and certification, medical computing and community computing. He has served as a consultant to several small and medium enterprises, large organizations, and government agencies. For his research efforts, Prof. Aziz has published more than 100 articles and reports in journals, proceedings, books and other media both local and international and has been awarded the Unesco Fellowship, AIT-Bangkok (1988) and Senior Scientist Visit Program to Usbekistan - Renong Bhd (1995).