

A Review of Value Based Software Engineering and its Impacts

Muhammad Zahid Khan and M. N. A. Khan

*Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad,
Pakistan*

khanzahidkhan@hotmail.com, mnak2010@gmail.com

Abstract

Profitability is the main goal of every software organization. Software companies develop products to capture greater market place and gain the attention of targeted customers or stakeholder. In the past the software engineering field was not able to determine the stakeholder's value of the developing product as mostly systems were based on value neutral setting. With the emergence of Value based software engineering (VBSE) the traditional software engineering discipline is vastly changed as during product development the stakeholder's value proposition plays an important part and the development is done accordingly. In this article I cover some important aspects of value based software engineering regarding their impact in software reusability and quality. Different value based pricing criteria are discussed. Empirical research and value based selection of best suitable automated tools and existing Component off the shelf (COTS) and conflicts between different stakeholders of the same system are also taken into account. In the 3rd section Critical review of the all papers is performed and their plus and negative points are mentioned. At last the Conclusion and future work in value based software engineering field is presented.

Keywords: *Value Base Software, Reusability, Component Off-The Shelf, Stockholders Value Proposition, value base Empirical Analysis, Software Quality Achievement Framework, Risk Management*

1. Introduction

With advancements in the field of software engineering the "importance of value based software" has tremendously increased towards application of new methods and techniques in reevaluating business investments based on organization's expenses and profits. The value based software approach has altered some rules in traditional project feedback mechanism as it was based on value neutral settings; But due to the rapid change in technology, business demands, stakeholders requirements, market situations, competitiveness and profitability, pre-defined requirements, budgets, schedule and plans the need of adaptable mechanism is ever increasing.

The today's earned value systems cannot deliver the stakeholder value or business value rather they just track the cost and schedule of a project. To address these issues value based software approach emerges focusing on company's business value by improving efficiency of the under developing software economically. It is normally done through collecting the feedback of current expense of a project and its expected profit after examining these, adjustments been made depending upon these values. Whereas software engineering techniques which use value neutral settings frequently use the scare resources of project on those activities which has a negative impact on profit. So to respond to these needs a new engineering approach termed as value based software engineering (VBSE) is emerged. In order to incorporate these value integration in to new systems and

already existing systems so they can support and communicate with each other in a manner useful for both customers and the vender [1].

In early stages of software engineering decision tasks regarding under developing software were comparatively slight effect on software development expenditure, allocated budget, and schedule. So at that time mostly approaches was based on value neutral settings, but with the passage of time advancement has come in software engineering field and in this era software has a main influence on most systems cost, schedule, and value plus software decisions are inevitably tangled with system level decisions. In addition, value neutral software engineering principles and practices are not capable to deal with most of the causes of software project failure. Major Research studies conducted by Standish Group's CHAOS reports 1995 and 2001 discovers that mainly software project failures are caused by value oriented shortfalls for example lack of user input, deficient requirements, varying requirements, resources shortage, impractical expectations, vague objectives, and not viable time frames. Moreover, value neutral methods are inadequate as a basis of engineering discipline. Therefore the value base approach fills out that space which was left behind by the value neutral software engineering [1].

2. Literature Review

Software reuse is a process to implement or update the software system by using pre-existing software components [2]. Well defined organized software reuse method is very important for survival of any organization because due to this the overall production of any company can take a boom. The quality of the products will increase, development budgets and product delivery time will be minimized [3,4]. With the help of reuse organization can save it resources in term of human efforts, schedule constraints can be eliminated. By reproducing same items again and again enable organization to maintain quality among the reproducing items and they all known risks are dealt with a great sense of responsibility [4]. Fundamental part in value based software engineering is the software metrics.

Software metrics follow a systematic approach for measurement of any software development process a Result of this process estimation of resultant data and precise management of allocated schedule is performed, which help in better development process and greater output [5]. Following is the information acquired from metrics and with the help of this data several managerial decision are made for the achievement of best quality products. Important concepts of software reusability mention below:

- Measuring cost of software reuse.
- Measurement cost in present value concept.
- Software reuse risk factors

In first of software reuse the distribution is done in two different type mention below:-

- Reusability of items lacking of variation.
- Items reusability in terms of variation.
- In any project there is no generic method for risks removal only there amount and severity can be minimized. There are 2 major type of Risks mentioned below:

The decision between severity of risk is not a piece of cake as both risks have their own consequences and also they are used in software reusability [5, 9]. Method use for reusability should be an organized, pre planned and systematic because company profit depends upon this method. To ensure the quality of deliverable products and under

developing one's reusability technique should be well managed according to quality and standards for the betterment and goodwill of organization [7].

The Quality Achievement framework based on value is [11] resultant of stakeholder's Win-Win conditions [12]. Some value base software theories are also involved in this method [13]. It contains certain principles of process instance generation for the better achievement of stakeholder software quality requirements based upon risk driven approach. This frame work give insight of different system and process attributes with help of those we are able to do the process of identifying plus resolving the value conflicts of quality attributes (Q-attributes) during the risk analysis process, technology or architectural evaluation plans and milestone reviews. Using this approach schedule and cost are also dealt as quality attributes in under developing projects. This method also provides us the opportunity to use "the real earned value" and acquire the quality requirements by means of monitoring and controlling the overall progress.

Rather than applying the whole metrics for estimation of quality of software. The "value base" quality framework provides the facility to its users to gather the most important value propositions of users or customers which can be prioritization of various processes or instances, provide a difference between expect value and the actual value and also take into account the different quality criteria. Identification and resolution of different contradiction among different attributes is also done with the help of this. Different architectural and other domain related decisions are made with the help of this process as well [11].

Hang *et al.*, proposed an approach on the basis of risk occurrences, to improve the overall process frame work by embedding risk-based process into the software quality frame work to improve the usability of overall process [10]. Three different process strategies were applied in the enterprise resources planning each strategy covers specific market area and specific amount of stakeholders. Those different strategies are mentioned below:

- Strategy powered by Schedule
- Strategy powered by product
- Strategy powered by Market trend

Processes used in schedule powered are driven processes are trivial processes consists of small repeated circular iterations where as processes used in product powered strategy are consists of slightly bigger repeated circular iterations. Integration of risks based process with quality process framework provides several benefits like it can generate a precise output result .The most important operations of this process is identification of system users or customers, analyzing different working cases from business point of view and communication of main developers with the different project related people like customer, users and stakeholders [10].

The main purpose of business case analysis is to collect crucial system users requirement and their value perception of the under developing product. Software quality risks and schedule risks are identified in risks analysis. Business analysis is strongly connected with risk analysis as potential risks in any project are prioritized according to their business cases [10]. In case of schedule powered risk domination over risks powered by quality then strategy powered by risks will be used. but when quality powered risk domination occur over schedule one then quality powered strategy is used, and when at certain stage when both values are same then encapsulation of schedule power parts is done and the process is applied somewhere else in the product [10].

As we know in software field change is inevitable so one decision cannot be applicable for all times, thus project staff is responsible for constantly check and supervise the output to maintain solid knowledge of the system and to accommodate changes in the system [16].

The process of small up gradation between different sections of a company when takes place the strategy powered by schedule is implemented. Examples of such technique are addition or deletion of certain entities or tables, updating of different entities etc these actions have high priority most of the time so they might be needed without any delays. So these kind of system functionalities got high priority and stakeholders demands these to be operational as soon as possible [10].

The quality process framework is tailored to act according to the given business scenario and ensure quality and corrective actions. When this process is applied the system users willingly tolerate little discrepancies in quality and can wait for achievement of those till full system operation functionality [10].

When a high version updating/up gradation is required after collecting different updating/up gradation request from various section the approach powered by product is applied. Quality attribute of updated product has more influence over reaching the deadline. Risks powered by quality got more priority then risks powered by schedule. These requirements are collected from many sources so there is a conflict possibility among different quality entities [10].

In case of product up gradation is based upon market trend powered or on competition basis, products Market trend driven strategy is used. The most important and very basic operation is to gain attention of everyone in the market and cover greater part of it. This strategy is the combination of both schedule powered strategy and product powered [10].

By embedding risk based process strategy decision making method into value based software quality achievement process framework; enable us to alter the process for use in different business cases. With the help of this approach flexibility in the process can be achieved. Comparison of business cases and different risks resolving techniques can help in a selection of accurate strategy [10].

Pricing strategy is very critical in any market place it's either boost sales or destroy demands in a fraction of time. Announcement of price for any newly developed product is a critical task for an organization which has to be taken at launch time of any product. From a marketing perspective, the main purpose or aim of price policy is to define an amount for a product which is the financially profitable for the company as well as represent the value perceived as customers given value to that product to obtain certain financial benefits or aims set by vender to get maximum level of investment from the product [14].

Project managers or price determining bodies usually deploy price for newly developed product in terms of "how much value a customer is giving to the product", in contrast cost base price policies are define at vender's own given value to the product so these are valid only for a very short interval. Whereas Value base price policies are made on the assumptions that at what extent a customer will go for the new product instead of its development cost [14].

The most widely used method for software pricing is cost based as it relies heavily on information from the cost accounting system. The production of operating results, budgets, and financial statements are done through this data. Managers took pricing decision of the software products on these data to get maximum profit from market [14].

The elementary drawback of cost based policies is they are set according to vender's own given value to the product which mostly represents the total financial expenditure done during the development of product. At first, a unit cost is dependent on volume. Fixed cost/per unit is a specific number which varies according to the volume of every project. The allocation process can be consists of calculating total working hours or by usage of substitute metrics etc, but maintain accuracy in these is not a piece of cake. Hence costs of different product could not be discrete rather a very hectic task of examining and changing the policies [14].

Moreover cost related policies are measured in form of circular loop theme, represents logic depended upon single unit/ volume. Sale volumes in this case will be the exact

decision makers of price. If the managing crew is incompetent of making the right decisions then different price policies suffered in various market places. Due to the circular nature of cost based pricing a pricing problem may occur which results in a problem of over pricing or under pricing in different places. [14].

Software companies often adopt pricing strategies from requirements points of view to encircle the total costs and maintain a handsome profit criteria mostly based upon detriment of their customer relationships. Value given by a user to any product is the main source of income for a company as a customer is willing to acquire that product if the price of the product is equal to the value perceived by customer. So in this regard most important and critical point in customer's given value to the product is the correct actions an organization has to take and figure out correctly and precisely the customer perception of the product value. To achieve this value different tradeoff decisions have to be implemented to maintain a fair balance among company's own profit and customer's value of the product. Executive of a company must be aware of these tradeoffs [14].

Software pricing depends upon several factors due to the different trends of market and mind set of the stakeholders. In Value based software the customer's perception and demands got more priority than other price determining factors [14]. Value based software pricing decisions are mostly made on the basis of stakeholders perception and their priority towards the under developing product as well as different market scenarios. There are different kinds of stakeholders; to capture their attention different types of value based pricing strategies are generated [14]. Different kinds of stakeholders and pricing strategies are:

- Quality sensitive
- Customers with high degree of price sensitivity
- Low degree price sensitive customers
- Price signaling
- Competitive pricing
- Reference pricing

Quality sensitive customers/stakeholder's main aim in acquiring a software product is its quality rather than its purchasing cost. Conversely customers with having low degree of price sensitivity can tolerate some quality attributes. When time is the major decision factor of new products pricing then price signaling strategy is used, Innovators having greater amount of cost mostly on searching for new idea, trend or technique and having a high moral do not invest heavily on evaluation of alternative products. Reference pricing is the variant of signaling price policy, in this policy; product buyers invest heavily on search tasks and risk perception as compare to innovators [14].

A mutual reference point is needed for the calibration of different values regarding to cost and quality affiliation. Deployment of a reference pricing strategy can generate tremendous results in competitive market conditions. Software organizations frequently introduce competitive pricing strategies to capture greater market place [14].

Empirical research has become the building block of new trends and techniques in software engineering field. Software engineering is mostly related with practicalities than theoretical hypothesis. Facts provided by empirical researches make it possible to evolve and draw important results for future research in various engineering fields and new development architectures can be designed as out of this study. Similarly in value base software engineering, the architectural, entity base and functional level connections between several research and designs in a given development process can elaborate many customers requirements some of them maybe contradict with each other. So for their resolution specific planning policies should be applied [15].

An empirical value base framework is presented in this article, for collecting and examining the win-conditions of intended user for the comparison of different advantages and disadvantages of various empirical ongoing researches. The value base empirical

research frame work (VBERF) emerges from benefit realization technique [15]. The value based empirical research framework (VBERF) proposed by Biffi *et al.*, builds on the realization approach of value base software [17].

Value base software engineering key objective is integration of articulated stakeholders value propositions (SVPs) into the full spectrum of software engineering principles and practices so as to increase the return on investment. But the stakeholder's value propositions are constructed on basis of different prototypes; suppositions from different models are usually mismatched, differing or uneven sometimes [23]. Main system stakeholders such as users, developers, acquirers, and maintainers utilize a variety of models like process, product, property, and success to articulate their stakeholder value propositions (SVPs) which can easily lead to inconsistency [23]. Software system development is naturally hard. The four crucial difficulties which are faces during every development process are [23]:

- Complexity
- Conformity
- Changeability and
- Invisibility

Still remain valid in developing large software systems. Inconsistent or conflicting stakeholder propositions (SVPs) are bound to arise in almost every development process. These inconsistencies and contradiction, if not properly addressed, would result in value reduction or even the destruction for software asset, thus threatening the main objective of value creation in value base software engineering (VBSE) [23]. In value neutral software development system the inconsistency management between different entities, system and modules got significant importance and research is done from the very beginning of software engineering [23]. Discrepancies in the contradicting stakeholders value proposition (SVPs) of five different kinds are defined and discussed. All are based upon contradiction of different model assumption along with each other. The defined value propositions are:

- Complementary inconsistency
- mutually exclusive inconsistency
- incompatible inconsistency
- precedence inconsistency
- Disagreeing inconsistency

For detection of above mentioned inconsistencies five algorithms are presented in the paper those could be useful for detection and removal of inconsistent Stakeholder value propositions (SVPs). For those algorithms there is one condition which should be met before the execution of any algorithm, which is the set of assumptions for models used by stakeholders to express their "SVPs", can be represented as formulas of the first order logic and some predicates like mutually exclusive, synonymous and antonymous can be pre-specified and available to the algorithms. With help of these value-based developments can give tremendous economic benefits to any organization adopting the above mention phenomena [23]. For project complexity management, project managers requirements engineers, and other project participants need best suitable requirements tool. There is a verity of different requirements tools available in the market each having different characteristics, capabilities, concepts, and terminologies. The Selection of most appropriate and precise tool is the rightful responsibility of project manager [24]. Tools are designed to reduce human effort, speedup the undergoing process, increase the overall efficiency, achievement of accuracy and precision, reduction of resources and for Automated tools are very helpful and play a vital role in ensuring consistency among different components, completeness and conformance to different quality and development standards. Tools provide and boost a shared and systematic workspace atmosphere, boost the improvement process allow exactness to be simulated shrinks

development time and costs, especially in maintenance increases productivity makes structured techniques practical it [24].

Iqbal *et al.*, [25, 26] proposed performance metrics for software design and software project management. Process improvement methodologies are elaborated in [27, 28] and Khan *et al.*, [29] carried out quality assurance assessment. Amir *et al.*, [30] discussed agile software development processes. Khan *et al.*, [31] and Khan *et al.*, [32] analyzed issues pertaining to database query optimization and requirement engineering processes respectively. Umar and Khan [33, 34] analyzed non-functional requirements for software maintainability. Khan *et al.*, [35, 36] proposed a machine learning approaches for post-event timeline reconstruction. Khan [37] suggests that Bayesian techniques are more promising than other conventional machine learning techniques for timeline reconstruction. Rafique and Khan [38] explored various methods, practices and tools being used for static and live digital forensics. In [39], Bashir and Khan discuss triaging methodologies being used for live digital forensic analysis.

3. Critical Analysis

| Ref | Issues Discussed | Strength(s) | Limitations |
|-----|--|---|--|
| [1] | Four different procedures for evaluating different financial concepts like cost investment, present value concept, risks factors and decision trees and their role is argued and its relationships. Role of techniques in software economics. | 1. Increase productivity and improve quality. 2. Reduce development cost and time. 3. Save project resources | 1. Unplanned reuse technique will require additional budget, schedule and other resources. 2. All the mention methods only apply in specific conditions they cannot be generically applicable. |
| [2] | Value base software quality framework is tailored to project three business cases. 1. Product powered approach. 2. Schedule powered approach. 3. Market- trend powered approach. Risk base process Strategy decision making approach embedded into the VBSQA framework; to improve applying flexibility of VBSQA framework. | 1. It gives a practicable solution to a flexible process creation. 2. provide different business cases to cope with different business requirements | 1. Every strategy should be applied to a specific set of conditions so a rightful and precise method selection decision is required, which demands extra time and budget allocation. |
| [3] | 1. To Capture different kinds of stakeholders' different type of value based pricing strategies are discussed. | 1. A rigid price taxonomy that shows the interactions among user Needs, organization goals, and different price policies. | 3. Value base price policy is market specific as there is verity of customer perceptions. So choosing a target market is the great challenge and then determination of stakeholder's perception of value in the product. |
| [4] | 1. Empirical study planning issues. 2. Value based empirical planning research frame work. | It provides the following benefits: 1. idetification of Success critical stakeholders 2. Identify the clashes among different stakeholder's value propositions 3. Deliverable that do not give customer's value proposition or identification of missing stakeholder's value propositions. 4. Provide additional initiative for improvement | 1. Conducting an empirical research requires additional budget, time and organizational resources. So time, cost and schedule constraints are the main drawbacks of this approach. |

| | | | |
|------|--|---|---|
| | | of deliverables value. | |
| [5] | <p>1. Difference between traditional earned value system and stakeholder value.</p> <p>2. Value based monitoring and control system.</p> | <p>1. It facilitates organizations to systematically monitor and control the rapidly occurring risks to project failure by not delivering the expected value.</p> | <p>1. Value based monitoring and control (VBMN) is applicable at a small scale</p> <p>2. It is still not integrated with all fields of software engineering.</p> |
| [6] | <p>1. A value based approach for determination of quality investment.</p> <p>2. Integration of different models takes place for achievement of optimal quantitative quality investment level.</p> | <p>1. These value based quality models provide approximate judgment of relative quality Investment level.</p> <p>2. Due to this approach rapidly changing business requirements, market situations, unclear or ambiguous stakeholder value propositions and different risk occurrences can be controlled.</p> | <p>1. Only large and well established organizations apply this model as they got greater resources and stable requirements. Small companies would preferably spend time on development rather than spending extra time on model calculations.</p> |
| [7] | <p>1. A frame work for the estimation of cost plus benefits of an inspection process is introduced, based upon software system and its parts.</p> <p>2. Benefits of inspection process.</p> <p>3. Different cost benefit related approaches.</p> | <p>1. A simplified and detailed model of inspection related benefits from which qualitative evaluations of inspection processes can be drawn.</p> <p>2. Inspection minimizes the risk of project failure by detecting and removing any defects at every phase of the software development life cycle.</p> | <p>1. The presented approach is not able to give practical guidelines to how optimize the inspection process.</p> <p>2. Assumptions are made during this approach so it is also specific not generic model.</p> |
| [8] | <p>1. Role of component off the shelf products (COTS) in software industry.</p> <p>2. Advantages of using COTS product.</p> <p>3. Value based process decision frame work</p> | <p>1. A value based set of processes can save large investment and steers the project towards successful cost effective integration.</p> | <p>1. Complexity of the project increases as we imbed several COTS products into the system.</p> <p>2. Architecture becomes weaker and ambiguous.</p> <p>3. Maintenance of COTS integrated systems is also a big challenge.</p> |
| [9] | <p>1. Different stakeholder's value prepositions are discussed.</p> <p>2. five different type of antagonistic are presented</p> <p>3. Algorithms are proposed to capture them.</p> | <p>1. Gives an understanding of various stockholders and their values about products.</p> <p>2. Removal of contradictory stakeholders' value proposition can be done with the help of these presented approaches.</p> | <p>1. These approaches are specified as its captures only few customers not all.</p> |
| [10] | <p>Usefulness of tools in software engineering.</p> <p>2. Value based approach for selection of appropriate tool is discussed.</p> <p>3. Tool rating model development is also discussed.</p> | <p>The rating model can provide a better chance of selecting the most precise tool.</p> <p>2. Project managers can use this approach for choosing best available and project relating tool to improve the performance and quality of the software product.</p> | <p>A large verity of commercial and open source tools are available in the software industry. Comparison of those tools is very expensive and time consuming task.</p> <p>2.</p> |

4. Conclusion and Future Work

In this report different value based software techniques, approaches are discussed and their usability in software industry is taken into account. The software or component reusability is the fastest way of delivering software product to intended customer or market; it provides a handsome amount of savings to any organization in term of money, time and other resources. Tailoring value based processes according to different business condition can also improve the quality of product and productivity. Now-a-days determination of stakeholder value of the product is very critical for any software success so for this purpose empirical research and cost-benefit processes are used. Software pricing is also the most important factor as it depicts the profit or loss of the company so it should be prepared very carefully according to the customers or stakeholder's value perception of the product as there are various kinds of customers so pricing policy should be applied accordingly.

For the success of every newly developed product conflict resolution among different stakeholder value propositions must be performed. Value based software is mostly influenced by customer's value of the product so quality product will capture more market than a normal one. Recently user requirements and market conditions are rapidly changing, customers demand working product in minimal time so to compete with these business needs automated tools and component off the shelf (COTS) are used to deliver a quality product in less time and development cost.

As Value based software engineering is quite new emerging field in computer science so there are various areas open for research and development. There are many challenges lies in each paper I discussed like the improvement of pricing strategies to attract almost every kind of stakeholders or integration of cost-benefit process with other estimating models to predict more accurate results. A real time stockholder value proposition repository can be created which regularly checks changes in value propositions and deliver the result to all project customers for their future development.

References

- [1] B. Boem, "Value based software engineering overview and agenda", Springer, (2006).
- [2] A Quality Metric for Component Based Software Products K. S. Jasmine, And R. Vasa tha, 2007, Venue: Products", proceedings of World Academy Of Science, Engineering and Technology, Vol 23, ISSN.
- [3] Software Reuse - Facts and Myths Kevin D. Wentzel., (1994).
- [4] "An Empirical Investigation of Software Reuse Benefits in a Large Telecom Product Parastoo Mohagheghi and Reidar Conradi", ACM Transactions on Software Engineering and Methodology (TOSEM) TOSEM Homepage archive, vol. 17, no. 3, (2008) June, Article No. 13.
- [5] "Value Based Software Reuse Investment By John M. Favaro", Kenneth, R. Favaro, Paul F. Favaro, Annals of Software Engineering archive, vol. 5, (1998), pp. 5-52.
- [6] Cost Estimation Model for Reuse Based Software Products Jasmine K.S, Dr. R. Vasantha. Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, Hong Kong, (2008) March 19-21.
- [7] K. Yadav, A. Maharshi and V. M. Shrimal, "Optimizing value base software reuse investment", ijarcsse, (2012) July. www.ijarcsse.com.
- [8] Assess Reuse Risks and Costs.
- [9] "A New Process Model for Reuse Based Software Development Approach" Jasmine K.S, Dr. R. Vasantha Proceedings of the International MultiConference of Engineers and Computer Scientists 2008, IMECS 2008, Hong Kong, vol. I, (2008) March 19-21.
- [10] J. Lu, L. Huang, H. Hu, J. Ge and B. Boehm, "Tailor the Value-Based Software Quality Achievement Process to Project Business Cases", Proceedings of Software Process Workshop (SPW) 2006, LNCS, (2006) May.
- [11] L. Huang, "A Value-Based Process for Achieving Software Dependability", Proceedings of International Software Process Workshop, Beijing, China, LNCS, Springer Verlag, (2005).
- [12] B. Boehm and W. Hansenzz, "Understanding the Spiral Model as a Tool for Evolutionary Acquisition", CrossTalk, (2001) May.
- [13] B. Boehm and A. Jain, "An Initial Theory of VBSE", A. Aurum, S. Biffel, B. Boehm, H. Erdogmus, and P. Gruenbacher, Value-Based Software Engineering, Springer Verlag, (2005).

- [14] R. Harmon, D. Raffo and S. Faulk, "Value-Based Pricing For New Software Products: Strategy Insights for Developers", (2005).
- [15] S. Biffi and D. Winkler, "Value-Based Empirical Research Plan Evaluation", Empirical software engineering and measurement, (2007).
- [16] B. Boehm and R. Turner, "Balancing Agility and Discipline", Addison Wesley, (2004).
- [17] B. Boehm, "Value-Based Software Engineering: Seven Key Elements and Ethical Considerations", Biffi et al. (eds.) "Value-Based Software Engineering", Springer, (2005).
- [18] "Value-Based Feedback in Software and Information Systems Development Barry Boehm", Li Guo Huang, USC, (2002) June.
- [19] "How Much Software Quality Investment is enough: A Value-Based Approach LiGuo Huang Barry Boehm Computer Science Department University of Southern California Los Angeles", CA 90089-0781 {liguohua, boehm}@sunset.usc.edu. Software, IEEE, vol. 23, no. 5, (2006), September-October.
- [20] O. Laitenberger, "Cost-effective Detection of Software Defects through Perspective-based Inspections", PhD thesis", University of Kaiserslautern, Germany, www.iese.fhg.de, (2000) May.
- [21] S. Biffi and M. Halling, "A Value-Based Framework for the Cost-Benefit Evaluation of Software Inspection Processes, (2002).
- [22] Y. Yang, Dept. of Comput. Sci., Univ. of Southern California, Los Angeles, CA, USA, Bhuta, J.; Boehm, B.; Port, D. N. Value-based processes for COTS-based applications, Software, IEEE, vol. 22, no. 4, (2005).
- [23] "Capturing Antagonistic Stakeholder Value Propositions in Value-Based Software Development", Du Zhang Department of Computer Science California State University Sacramento, D Zhang - Proc. of 22nd International Conference on Software, (2010).
- [24] "Value-Based Selection of Requirements Engineering Tool Support", Matthias, Heindl, M. ; Siemens Program & Syst. Eng., Vienna ; Reinisch, F. ; Biffi, S. ; Egyed, Software Engineering and Advanced Applications, 2006. SEAA '06. 32nd EUROMICRO Conference, (2006).
- [25] S. Iqbal, M. Khalid and M. N. A. Khan, "A Distinctive Suite of Performance Metrics for Software Design", International Journal of Software Engineering & Its Applications, vol. 7, no. 5, (2013).
- [26] S. Iqbal, M. N. A. Khan, "Yet another Set of Requirement Metrics for Software Projects", International Journal of Software Engineering & Its Applications, vol. 6, no. 1, (2012).
- [27] M. Faizan, S. Ulhaq and M. N. A. Khan, "Defect Prevention and Process Improvement Methodology for Outsourced Software Projects", Middle-East Journal of Scientific Research, vol. 19, no. 5, (2014), pp. 674-682.
- [28] M. Faizan, M. N. A. Khan and S. Ulhaq, "Contemporary Trends in Defect Prevention: A Survey Report", International Journal of Modern Education & Computer Science, vol. 4, no. 3, (2012).
- [29] K. Khan, A. Khan, M. Aamir and M. N. A. Khan, "Quality Assurance Assessment in Global Software Development", World Applied Sciences Journal, vol. 24, no. 11, (2013).
- [30] M. Amir, K. Khan, A. Khan and M. N. A. Khan, "An Appraisal of Agile Software Development Process", International Journal of Advanced Science & Technology, vol. 58, (2013).
- [31] M. Khan and M. N. A. Khan, "Exploring Query Optimization Techniques in Relational Databases", International Journal of Database Theory & Application, vol. 6, no. 3, (2013).
- [32] M. N. A. Khan, M. Khalid and S. Ulhaq, "Review of Requirements Management Issues in Software Development", International Journal of Modern Education & Computer Science, vol. 5, no. 1, (2013).
- [33] M. Umar and M. N. A. Khan, "A Framework to Separate Non-Functional Requirements for System Maintainability", Kuwait Journal of Science & Engineering, vol. 39, no. 1B, (2012), pp. 211-231.
- [34] M. Umar and M. N. A. Khan, "Analyzing Non-Functional Requirements (NFRs) for software development", IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), (2011), pp. 675-678.
- [35] M N A. Khan, C. R. Chatwin and R. C. Young, "A framework for post-event timeline reconstruction using neural networks. digital investigation", vol. 4, no. 3, (2007), pp. 146-157.
- [36] M. N. A. Khan, C. R. Chatwin and R. C. Young, "Extracting Evidence from Filesystem Activity using Bayesian Networks", International journal of Forensic computer science, vol. 1, (2007), pp. 50-63.
- [37] M. N. A. Khan, "Performance analysis of Bayesian networks and neural networks in classification of file system activities", Computers & Security, vol. 31, no. 4, (2012), pp. 391-401.
- [38] M. Rafique and M. N. A. Khan, "Exploring Static and Live Digital Forensics: Methods, Practices and Tools", International Journal of Scientific & Engineering Research, vol. 4, no. 10, pp. 1048-1056.
- [39] M. S. Bashir and M. N. A. Khan, "Triage in Live Digital Forensic Analysis", International journal of Forensic Computer Science, vol. 1, (2013), pp. 35-44.