

Managing the Web-based Software Product Development: Towards Feedback Loop Methodology

Gunjan Tomer

Indian Institute of management, Indore, India

Email: f10gunjant@iimidr.ac.in

Abstract

This paper comments on the challenges and constraints in the internet software development arena. Our study also reviews the methodologies currently adopted by various companies and based on the findings suggest an improved methodology for internet software development. We have adopted an exploratory study using qualitative method as studies in the Indian context are found to be scarce. Our findings indicate that in comparison to traditional software methodologies internet/web software development faces a different set of challenges. Based on a research conducted on IT professionals involved in internet/web software development, this paper finds out that current methodologies may lead in sub optimal internet software development due to lack of vision for serving the end users. Our findings, based on the qualitative data analysis suggest that the advantage of technological innovations has overshadowed the customer's perspective and expectations. The importance of analyzing consumers' requirements is not fully realized, thus not explicating the value proposition of the end product. In order to help a coherent, business-aligned software development, this paper proposes 'feedback loop method' which emphasizes the need for evolution among current methodologies in practice. The paper will be of interest to researchers, software practitioners and managers specifically involved in internet application and software development.

Keywords: Software development, Internet software, web development, development environment.

1. Introduction

The World Wide Web has evolved from being a communication medium for businesses to its present position where it has become the core of various business functions. Since then, the changes in the world of internet have been exponential and many a times unpredictable. Businesses are creating economic value from the information and access that is provided by the internet. The expanding usage of internet and 'www' has created many opportunities (e.g. E-Commerce) in business domain. The unrivaled growth of electronic commerce has increased the demand of internet software, thus extending the demand of software development for the web. However, this growth and pace of change in the web world have posed new challenges for the web application developers. Few of these challenges are identified as (1) high speed development to reduce time to market and (2) frequent upgrades to meet the changing demands. The traditional software

development methodologies, somehow, were not sufficient to provide the required level of flexibility and speediness in the development process. To address these challenges Internet software development has evolved over years and has created a different set of methodologies for development. These methods are influenced by need of time and the pressures exerted on software developers.¹

Software engineering is now facing two dynamic innovation streams² technological innovations and market innovation. Harshly shook by rapid technological development and highly volatile market environments, today's software development is under the constant pressure for swift and reliable development practices and enabling the product launch in reduced time. This time compression for time to market has also led to solutions such as automation of web development process. There are various platforms available which allow developers to get a structured linear development process which

is often template based and generate databases designs and interface pages.

The integrated environments³ provide assistance in development phases ranging from conceptual design to implementation details. Object-oriented Hypermedia Design Model (OOHDM) -Web⁴ Araneus, AutoWeb, JWeb⁵ and the rather recent full stack web frameworks such as Ruby-on-rails, Django and CakePHP tools belong to this integrated category. The implementation-oriented development environments tend to evolve into integrated ones. However, in the presence of employing these semi-automated environment and predominance of technologies, the business value and end user's requirements are often ignored. So the current internet software development industry is providing software products in minimum time to market and these products are getting obsolete at a much faster pace. There are no ways employed to understand what end user's are deriving from these software products. At ground level, often internet software is meant to leverage the business, for instance, development of an e-commerce website or a mobile application. So, we propose that the business proposition factor should be considered vital.

Businesses frequently use the concept of a value proposition⁶ to characterize the combination of end-result benefits and price to a prospective customer from purchasing a particular product. A customer will choose the product that offers the best value with regards to associated benefits and cost. So the idea fuelling this study is to give equal importance to requirement elicitation and value proposition. To develop stable, high value software keeping the end user or customer's perspective is equally important.

Despite ample amount of research in this area, explaining different innovations such as agile methodology⁷ and extreme programming, very few have actually succeeded in realizing the involvement of end users and consumers. Our study attempts to fill the gap by exploring the development process of internet software and propose an alternative internet software development methodology.

2. Theoretical background

2.1 Models of software development process

The primary reason for modeling the software development processes was to present a linear order of the stages involved. These models guided what phases should be carried out for a successful project. Taking a historical view, the very first model for software development was "code and fix" model which works on a simple approach of coding and

then fixing the errors subsequently according to requirement and design. However, there were some visible problems with this threadbare model. First, coding was done without any predefined design or modeling and fixes were costly and become even costlier as code-bases were completely unstructured.

The first structured approach to software development was the advent of Waterfall model. Waterfall models break down the process into distinct linear stages. These stages were hierarchical⁸ and each state worked as the input to the next stage. For example, completion of requirement analysis will lead to software design and so on. "Based on this linear process model, many modified models have been devised, such as 'incremental', 'spiral', 'concurrent', and 'evolutional' models. Prototype model was proposed to address issues of non descriptive requirements. In these cases a working prototype can be designed to give a preview of the final product. The concept of prototyping also gave way to evolutionary model which provides customer with initial operational software (also called minimum viable product or MVP) which can be further modified and upgraded as per the requirement. The spiral model also follows an incremental approach like prototyping and follows a cyclic development model⁹ instead of the linear model. The model reflects the underlying concept that each cycle involves a progression that addresses the same sequence of steps, for each of its levels of elaboration, from an overall concept of operation document down to the coding of each individual program.

Every model has its own significance and it is very difficult to establish any predominance. Development environment and nature of product should be considered for selection of an appropriate model. The effectiveness of a system-design method is influenced by a project's requirements, characteristics, user¹⁰ and designer satisfaction, and impact on the decision making process. The main objectives of traditional software development projects were to produce cost effective software that fulfills specific requirement. The development process involves standard models and technologies. These projects usually were time taking and were stable once deployed. Upgrades and new versions were rare and costly.

2.2 Internet software development

Growing internet usage and emerging e-commerce has given way to an entirely different wave of application development, internet software. The requirements and market conditions for internet software is entirely different from those of traditional software. Products should be launched faster with

subsequent upgrades and newer versions. So there is no final product for internet applications and development is a continuous process. Table 1 presents a comparison of both traditional software and internet software on the basis of their characteristics.

The current practices for internet development emphasize more on tools and technology. Mostly all products follow evolutionary and incremental approach. In studies, (see Refs 11 and 12) the most remarkable finding of his work was that getting a low-functionality version of the product into customers' hands at much faster pace and the quality (refer figure 1). It actually presents a stark contrast with traditional software development and implementation where a product delivered to the client was expected to be of high quality and stable. Some of the practices that are prevalent in the internet software industry are evolved due to time to market compression and environmental changes. Shelf-life of these software products is considerably lessened as technologies are changing at a faster pace and user demands are fluid.¹³ Component based development¹⁴ is also a very common practice in internet software development. This practice breaks software development into a series of smaller projects, each with a small requirement-set and a short development cycle. All these methodologies are demand based and evolved over time, mainly because of these fundamental differences between traditional software and internet software

2.3 Changes in environment and expectations

Change is inevitable. In high speed internet software industry, fast changes are inevitable. In 1960 when software industry was taking shape many methodologies evolved to develop standard and durable software products. In those times a stable, error free system was the requirement. The computer industry grew by double-digit percentages throughout the 1960s¹⁵ and by the end of the 1970s, nearly 100 percent of Fortune 500 firms substantially used computers. During this same period, the first commercial software application products arrived. Since then the growth of computing devices in all spheres of human life had become the reality.

In current web enabled world every technology is short lived. Software are getting obsolete and new releases are marked in very short span of time. It actually impacted the way software gets developed. Internet services never become 'final products' and are rather constantly revised¹⁶ and added with new features and functions in dynamic response to user feedback. It is obvious that internet service development is one of the active and dynamic areas of software development. Customized methodologies are adopted by companies and development tools have also evolved. There is a visible shift in the usage of programming language and other tools. The development environment has changed.

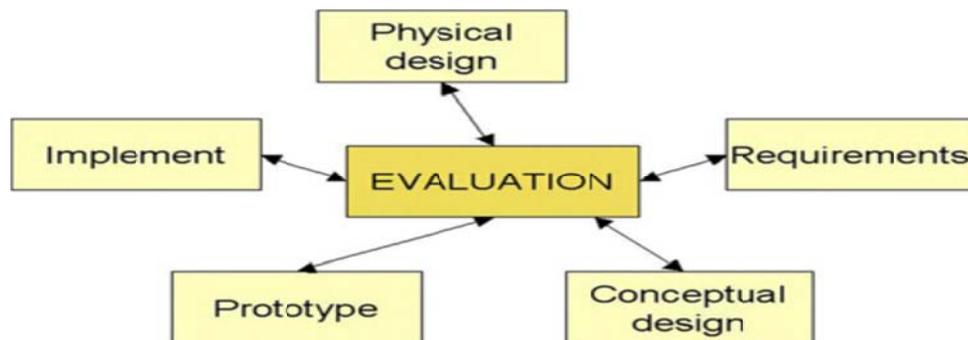


Figure 1: Current internet software development model in practice^a

^a Adapted figure (see Ref 30)

Table 1: Differences between traditional and internet software product development

Attributes	Traditional software	Internet software
End product Quality	High quality software/cost effective	Low functionality product
Time to market	High	Low
Development tools and technologies	Tools are programming languages, use cases, CASE tools	Multimedia, components, Ajax, HTML, XML, Ruby on rails, PHP.
Size of development teams	Large project team/Highly experienced	Smaller teams/ lesser experience due to constantly changing technology
Software development models	Traditional development models	Rapid development, Agile/ Xtreme programming
Product specification	Few reusable code, legacy involved	Reusable blocks, simple application
Requirement specification	Mostly rigid user requirements	Fluid user requirements

There are a lot of new avenues in application development with increased level of automation requirement and emerging web based markets. There are many new areas where software utilities are developing. It is a wide arena of search engines to internet shopping and so many web based applications. Computing and processing power has also risen exponentially. It has actually scaled up software development industry and has created many possibilities. Business conditions have changed to a great deal due to the advent of the internet and new business models are emerging. Internet applications have changed the way business is done, hence propagating new need in terms of software development. Strategies are evolving to collaborate business needs and IT. Software and applications are not just means to automate or improve efficiency of business, they are the new business. These developments are creating new markets and new needs. It has also affected the way organization works. So expectations and the environment both have evolved and dynamics of technology and markets determines the direction in which software development will go in future.

2.4 Automation in development

There are lots of standards and development frameworks that support automation in development by generating templates and providing information modeling. These platforms also enable conceptual design and implementation support.

The Araneus uses ER design for information modeling and Araneus Data Model (ADM) for navigation design. The JWeb is based on HDM¹⁷ which is based on ER and consists of structure design, navigation design, and presentation design. The AutoWeb uses HDM-Lite, an adaptation of HDM, especially tailored for Internet business systems. By contrast, the OOHDM-Web uses a conceptual model, navigation model, and interface model according to an OO methodology called OO-HDM.⁴ The OO-H CASE tool provides a class diagram, navigational access diagram (NAD) and abstract presentation diagram (APD) which enables a better conceptualization of product design and development.

These automation models have also skewed the focus towards technology rather than relevance and market value of software. However, these models can be

very effectively utilized in reducing time in design and code level so that a greater time can be devoted in understanding the business need and ensuring alignment to what customer expect and demand. It is essential that these models are always utilized with the larger objective of maximizing the value proposition of the end product.

Sample and Procedure

The participants of this study are working professionals from software industry who are involved in internet software and website development domain at various levels. The sample included developers, project managers, entrepreneurs running internet startups and end users. An email invitation was sent to 55 professionals and out of them 41 agreed to participate in the study.

To understand the process and models followed for development of software, semi structured interviews were employed. Interviews were framed to capture few key elements like flow of processes, hierarchy of different stages and their interdependency, decision factors for design and deployment.

All participants had more than three years of work experience in software development firms. Size of project groups varies from 6 members to more than 20 members and typical project duration was 3 to six months.

3. Interview protocol

One of the two authors has conducted each interview. Each interview was 45-60 minute semi-structured interview. There were multiple meetings with the respondents thus there were lots of follow up interviews. The language for conducting of interviews was English as English was the language of communication in all the organizations selected. The interviews were transcribed within 24 hours after interviews were taken.

The interviews were transcribed and analyzed to understand the outlining factors. The questions were organized in a way that allows participants to share their work and projects and to discuss about current trends and technologies. There were questions targeted at understanding the life cycle of product development and whether participants have noticed any adaptation in development processes. Our interviews tried to capture three fundamental dimensions. The first set of questions aimed at understanding the software development process. (1) To understand their priorities and protocols regarding the process we asked them questions such as what are the challenges you face with internet software development and how do you think you

cope up with them. The second and third sets of questions were centered to (2) obtain knowledge on their understanding of most significant parameters in internet software development (3) understanding their pattern of usage of development technologies. All these responses helped us to observe and understand their opinions and priorities.

We took prior permission from the respondents to participate in our study. As per the directives shared by relevant studies (see Ref 18), we shared with them that the responses will remain anonymous and only aggregate level of results will be shared on a public forum. We also ensured that they understand that the participation is voluntary and they can choose to withdraw from the study at any moment of time. However, we found most of our respondents very cooperative and forthcoming in sharing their views.

4. Data Analysis

Data analysis was conducted on the transcribed data and data codes and categories were generated. We followed a systematic process for the data analysis. We, after reading all the transcripts thoroughly, created blocks of narratives on the basis of similarity of concepts discussed in them. Based on the similar code blocks we created theoretical categories, which represented the emerging theory in the blocks. These theoretical categories were then revised and modified as more data was coded. We finally collated all the theoretical categories which contribute in building our understanding of the phenomenon. In the findings section, we have discussed each theoretical category independently and have drawn theoretical implication based on their interpretation.

5. Findings

5.1 Shorter time-to-market

One of the most crucial challenges in internet software development is the reduced "time-to-market". Most of the respondents discussed the time pressure experienced by them in designing and deploying their projects. They explained that how stages of development are fluid and often merge into each other. So, there are usually no boundaries between the development life cycle stages and in order to reduce turnaround time, efforts are made to run multiple processes together.

5.2 Dependence on tools and technologies

The responses were more centered towards task efficiency, use of latest technology and differentiation based on multimedia content. One

participant who was working as a project manager shared that his team is very competent in delivering quality internet software.

“...we roll out internet software and website in very short time...we make them good because our team is competent in all set of skills...back end optimization...designs with HTML5 and JS and brilliant animations”

Most of the participants were talking more about how technology has changed the process of software development. Many of them excitedly referred upcoming technologies which will change the market at a large level.

“...you must have heard about SQL but noSQL is the future...there are so many up gradations in the coming time which will be like a paradigm shift in internet software development. If you can play well with all these tools and technologies you are sure to get a successful product”

These responses hinted towards over reliance on advances in technology rather than the proposed value of the end product to the users.

5.3 Lack of focus on requirement elicitation

Requirement elicitation is considered to be the most crucial phase in a typical software development life cycle. It not only prescribes the specifications of the end product, but also ensures congruence between the client's expectation and the deliverables. We found that in the case of internet software development, requirements are very fluid and ambiguous.

“Every day so many internet applications and software enters into the market. Every one promises to offer best functionality and performance...there are so many features and so much competition...users tend to get confuse...sometimes these apps are creating needs among users which even they don't realize before using the product. So most of the time what they (end users) need is so much in flux.”

Many respondents cited lack of time as a reason for not focusing much on requirement elicitation. They mentioned that they experience much pressure for releasing the product in the shortest time possible.

“...it depends much on the end users', what is their profile, how well versed are they with the technology. So, sometimes even a good product fails because it is not appropriate for the target audience. We are always short on time, unlike traditional software development where much time and effort is given for eliciting requirements and documenting it.”

5.4 Lack of emphasis on business value proposition

The traditional software development environment offers more information on the requirements of the final software. There was a better level of certainty about the kind of end users. The internet software development has created new challenges of understanding the end user. The end user can be anyone from a 16 year college student to a 45 year old woman. The significance of this end user has increased much in the past years as they play a major role in determining the success or failure of a product. However, we found that the development teams working on these internet projects perceives that the technology is the *force majeure* of the development process. In many instances, it was very evident that the development community is deeply engrossed with technical aspects that they are naturally insulated from the world they are serving - the end users. Some of them explained that most requirements are taken on observations and hypothesis of few team members. There were many responses which started from I think or I believe. It is not a common practice to conduct a formal market study before product development.

“..I think people don't like filling feedback forms...”

“...I know people like viewing premium product at top of the list”

This allows them to overestimate the technology proficiency of users and thus designing effective but complex systems. On the basis of this study internet software development can be characterized on several criteria

6. Feedback loop method: A new effective model

As adding new features has become the driving factor of software development in internet time, lots of critical factors regarding the utility of the product are often ignored. As we have discussed methodology is adaptable and has evolved out of expectations of quick development and matching the technology, development is woven around use of latest tools and technology. The business value of the software and its congruence to business functions is often ignored. Recent development suggests that web applications and internet software have also commoditized.¹⁹ The internet websites and software are often treated like a product as there is sufficient amount of consumer interaction.

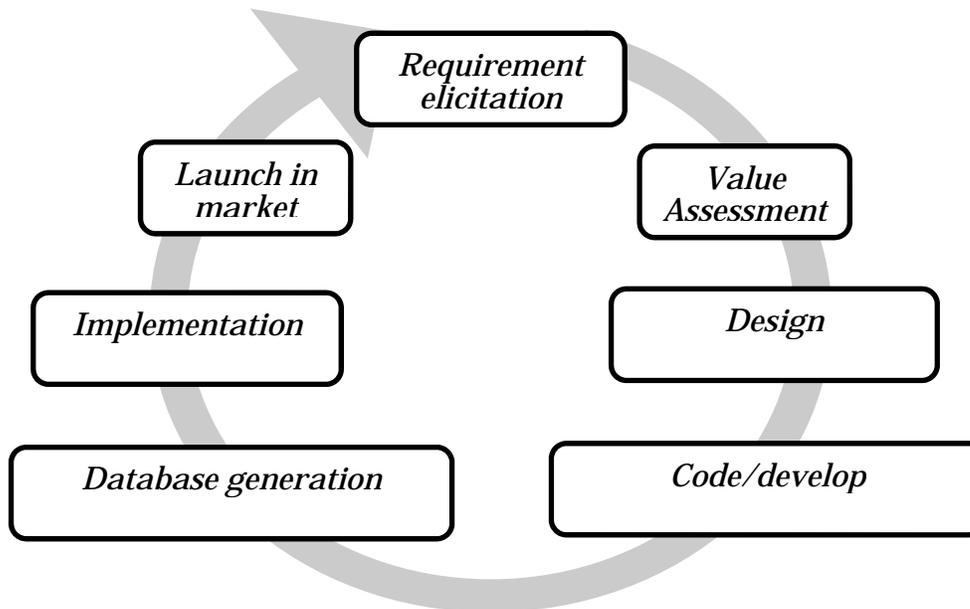


Figure 2: Feedback loop method: Continuous process of internet software development

Customer co-creation and customer implants are very common practices in internet software development companies.

Another issue with current methodology is that requirement elicitation and requirements specification phase is either ignored or not conducted thoroughly, reason being lack of time. Requirement specifications often help in understanding the customer associated value of the product which can lead to improvement in future versions and add more value to end users.

This new model therefore puts forward, two additions in the current model. Figure 2 illustrates the suggested stages in the development life cycle. Requirement elicitation phase combined with measuring customer's value²⁰ derived and alignment to business functions. Customer's analysis (requirement-benefit) is done in early stages.

One noticeable thing in the proposed model is that it is a continuous process. The focus nowadays is on launching the product in the market in earliest time possible and then improving on quality in subsequent releases. However, it also raises questions that whether consumer value assessment is feasible with fast launch and high speed development. To address the challenge of fast launch in market technologies like agile software process ASP (see Ref 21 and 22) and eXtreme programming (XP) can be employed.²³ Automation of development and overlapping release cycles can also facilitate high speed release.

It also ensures high quality by continuous feedback and quality assurance practices are candidates for compression in cycle time reduction.²⁴

The model accommodates itself to collaborative dynamics of business and technology and presents a framework to roll out relevant, 'high on value' software which is sustainable and technologically sound.

7. Limitation and scope

The study is limited to understand different aspects of methodologies involved in current internet software development. It does not deal with the effectiveness or failure of any particular technology and all prevalent technologies can co-exist with proposed model without any limitation. This study also explains the importance of customer analysis in the context of value associated with the product, however, there are no strategy suggestions on how we can improve our value proposition. Further studies can address issues like how to improve the value proposition of a web site or other internet software.

It can also be quantitatively supported by surveying internet users to understand factors that improve the value associated with particular software. This study can guide the development of any standard that can work as an efficient and adaptive methodology in internet software development context.

8. Contribution and implications for practice

The traditional software development methodologies mainly concentrate on developing durable, error free, high quality software. The requirements were specified with certainty from the customer so that there were no ambiguities in the product specification. In addition to this, customers require software that will not need newer versions in the near future and will be equally efficient for a longer period of time. New versions were considered costly and were released only when there was a major change in requirement at customer's end.

The current study contributes in several ways. First, it contributes to the software development literature as internet software and its impact on evolving methodologies is a less researched area. The advent of the internet age and internet based software brought a major shift in the way software is developed. There were lots of constraints and variances that has lead to an entirely different development methodology evolving over time.

In order to launch technologically advanced software in a short period of time more emphasis was given to tools and platforms to develop software. This approach actually ignores the customer's value derived from the products.²⁵ The behavior and intentions of customers can be positive if a better value proposition is offered with good service quality. While the web service quality²⁶ is an important driver of behavioral intentions, its indirect effect on customer satisfaction is equally important, if not more so, in promoting usage. Previous studies in the software development area also emphasize on the significance of the goal or objective of development rather than the methodology itself.²⁷ Our findings converge to these findings in the internet software development context.

In many Internet software development projects "requirements are fuzzy". In such situations having close access to customers helps in the "prioritization of features based on customer's demands".²⁸ The current methodology also doesn't spend time in capturing all requirements and requirements collection is left to feedback stage and changes are made in newer versions. However, this approach can only sub optimally utilizes the leverage of the internet age. In the era of internet, applications websites and software development plays a more crucial and determinant role in business. There is business models developed on the internet and these software plays more than a role of an application. Web technologies properly deployed²⁹, they may well permit companies to cost-effectively increase their productivity and, ultimately, their competitive

advantage. This makes the issue of development methodologies more relevant and challenging.

This paper also contributes through proposing a framework to develop internet software more effectively taken into consideration all the constraints imposed on internet software development. The new model proposes evolving requirement elicitation to be an important part of the process and before moving to design phase it suggests value assessment of the product so that our design and implementation offers a high value proposition.

References

1. Baskerville, R., Ramesh, B., Levine, L., Pries Heje, J., & Slaughter, S., Is internet-speed software development different? *IEEE software*, 20, No. 6, 2003, pp. 70-77.
2. Kakihara M., Strategizing Software Development: Strategic Management of Internet Service Development, Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research, 2006, pp. 37-43.
3. Lee Choongseok, Choi Byounggu, Lee Heeseok, A development environment for customer-oriented Internet business: eBizBench, *The Journal of Systems and Software*, Vol. 72, No. 2, 2004, pp. 163-178.
4. Schwabe, D., Rossi, G., The object-oriented hypermedia design model, *Communications of the ACM*, Vol. 38, No. 8, 1995, pp. 45-46.
5. Garzotto, F., Mainetti, L., Paolini, P., Hypermedia design, analysis, and evaluation issues, *Communication of the ACM*, Vol. 38, No. 8, 1995, pp. 74-86.
6. Keeney Ralph L., The Value of Internet Commerce to the Customer, *Management Science*, Vol. 45, 1999, No. 4, pp. 533-542.
7. Breivold, H. P., Sundmark, D., Wallin, P., & Larsson, S., What does research say about agile and architecture? In (ICSEA), Fifth International Conference on Software Engineering Advances, *IEEE*, 2010, August, pp. 32-37.
8. Pressman, R.S., *Software Engineering: A Practitioner's Approach*. New York, 2004, NY: McGraw-Hill.
9. Boehm Barry W., A spiral model of software development and enhancement, *Computer*, Vol 20, No., 9, 1987, pp. 61-72.
10. Mahmood Mo A. System Development Methods- A Comparative Investigation, *MIS Quarterly*, Vol. 11, No. 3, 1987, pp. 293-311.
11. Iansiti, M., & MacCormack, A. Developing products on Internet time. *Harvard business review*, Vol. 75, No. 5, 1997, pp. 108
12. MacCormack, A. How internet companies build software. *MIT Sloan Management Review*, 42(2), 2001, pp. 75-84

13. Mellis W., Loebbecke, C., & Baskerville, R., Requirements uncertainty in contract software development projects, *Journal of Computer Information Systems*, Vol. 53, 3, 2013 pp. 97-108
14. Baskerville Richard, Balasubramaniam, Ramesh, Linda Levine, Jan Pries- Heje, High speed software development practices: what work what doesn't, IT Pro, *IEEE Computer Society*, Vol. 8, No. 4, 2006, pp. 29-36.
15. Cortada James W, Researching the History of Software from the 1960s, *IEEE Annals of the History of Computing*, Vol, 24, No 1, 2002, pp. 72-79.
16. Offutt, J. Quality attributes of web software applications. *IEEE software*, Vol. 19, 2, 2002, pp. 25-32.
17. Bochicchio, M., piano, R., & Paolini, P., JWeb: an HDM Environment for fast development of Web Applications. In *Multimedia Computing and Systems*, 1999. *IEEE International Conference*, Vol. 2, 1990, July, pp. 809-813. *IEEE*.
18. Podsakoff, P. M., MacKenzie, S. B., Lee, J. Y., & Podsakoff, N. P. 2003. Common method biases in behavioral research: a critical review of the literature and recommended remedies. *Journal of applied psychology*, Vol. 88, No 5, 879-903
19. Ramesh, B., Pries-Heje, J., & Baskerville, R., Internet software engineering: A different class of processes. *Annals of Software Engineering*, Vol. 14, 2002, 169-195.
20. Kotler, P., Armstrong, G., Principles of Marketing. 1996, Prentice-Hall.
21. Aoyama, M., Web-Based Agile Software Development, *IEEE Software*, November 1996, pp. 56-65
22. Chuang, S. W., Luor, T., & Lu, H. P., Assessment of institutions, scholars, and contributions on agile software development (2001-2012). *Journal of Systems and Software*, 93, 2014, pp. 84-101.
23. Beck, K., *Extreme Programming Explained: Embrace Change*, 2002, Boston: Addison-Wesley
24. Wetherbe, J. C., Frolick, M. N., Cycle time reduction: concepts and case studies, *Communications of the AIS*, Vol. 3, No. 4, 2000, 1.
25. Ramrattan, M., & Patel, N. V., Web-based information systems development and dynamic organisational change: The need for development tools to cope with emergent information requirements. *Journal of Enterprise Information Management*, Vol. 23, No. 3, 2010, pp. 365-377
26. Udo Godwin J. , Bagchi Kallol K, Kirs Peeter J. An assessment of customers' e-service quality perception, satisfaction and intention, *International Journal of Information Management*, Vol. 30 No. 6, 2010, pp. 481-492
27. Iivari, J., Hirschheim, R., & Klein, H. K., A dynamic framework for classifying information systems development methodologies and approaches. *J. of Management Information Systems*, Vol. 17, No. 3, 2001, pp. 179-218.
28. Ramesh, B., Pries-Heje, J., and Baskerville, R., Internet Software Engineering: A Different Class of Processes, *Annals of Software Engineering*, Vol. 14, 2002, pp. 169 – 195.
29. Andriole, S.J., Business impact of web 2.0 technologies, *Communications of the ACM*, Vol. 53, No. 12, 2010, pp. 67-79.
30. Di'az Paloma, Aedo Ignacio, Towards efficient web engineering approaches through flexible process models, *The Journal of Systems and Software*, Vol. 80, No. 8, 2007, pp. 1375-1389.