International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 2 Issue 1 2011

ADAPTING THE SOFTWARE ENGINEERING PROCESS TO WEB ENGINEERING PROCESS

Sandeep Kumar Satyaveer Sangwan

Department of Information Technology,

M. M. Engineering College,

M. M. University, Mullana-Ambala (Haryana), India

Email: sandeepkumargarg@gmail.com, satyaveer.sangwan@gmail.com

Abstract: With the evolvement of web based applications, it has been found that the conventional software engineering models can not be used directly for the development of web based applications. The conventional software engineering life cycle models need to be adapted to cop up with the changing requirements of the web based applications such as the network intensiveness, time-to-market, availability, user concurrency, security etc. The paper identifies and analyses the various adaptations required in the conventional models to make them applicable for the development of web based applications. A model for the development of WebApps adapted from the conventional software engineering model has also been presented.

Keywords: Adaptations, Life cycle models, Migration, Software engineering, Web engineering.

1. Introduction

International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 2 Issue 1 2011

Software engineering deals with the systematic, disciplined and quantifiable approach to the development, operation and maintenance of software. With the advent of time, the web based applications are coming into play. It is usually observed that the existing software engineering models are not having all the features for the systematic development of web based applications. So, it was needed to adapt the conventional software engineering models to make them compatible for the development and maintenance of web based applications. To fulfill this need of developing the web based applications in the systematic form, a new approach of web engineering (WebE) comes into play. WebE deals with the use of sound scientific, engineering and systematic approach to the development, deployment and maintenance of high quality web based systems and applications (WebApps). But, it can be seen that the WebE model can be seen as the adapted version of software engineering model. The paper analysis this understanding and have identified the points in the software engineering models that can be handled to make the model useful for the development of WebApps. The paper also presents the adapted version of a WebE life cycle model.

The paper has been structured as follows. Apart from introduction in section-1, the section-2 presents the basics of software process models. The WebE process models have been introduced in the section-3. In section-4, various factors have been identified due to which the need of new approach to the development of WebApps has been felt. Section-5 proposes the adaptation of a software engineering model to the web engineering process model and presents a model for the development of WebApps as an adaptation from the software life cycle.

2. Software Process Models

A software process model [1, 2, 3, 4] is a conceptual representation of a process. It presents a description of a process from some particular viewpoint. Such models can be used to develop

more accurate and formalized descriptions of software life cycle activities. Software project utilize a process to enable execution of the engineering tasks to achieve the goal of delivering a software product that satisfies the user requirements. Processes so utilized frequently conform to a process model - a general process structure for the lifecycle of software development. The most influential and commonly used process model is the waterfall model, in which the different phases of requirements specification, design, coding, and testing are performed in sequence. This model was first proposed by Royce to suggest that there should be many distinct stages in a project execution. In their proposed work, the provision has also been made for feedback loops- from testing to design and from design to early stages of requirements [1].

Waterfall model has some well known limitations. The biggest drawback with the waterfall model is that it assumes, the requirements are stable and known at the start of the project. Unchanging requirements, unfortunately, do not exist in reality, and requirements do change and evolve [1] [4]. To remove these limitations, an iterative development model can be employed. The incremental process model uses the same steps as the waterfall process model but the steps are much smaller than the waterfall steps. In an iterative development, software is built and delivered to the customer in iterations – each iteration delivers a working software system that is generally an increment to the previous delivery. Requirements need not be completely understood and specified at the start of the project. The requirements can evolve over time and can be incorporated in the system in any iteration. Overall, iterative development is able to handle some of the shortcomings of the waterfall model, and is well suited for the rapidly changing business world, despite having some of its own drawbacks [2].

In order to overcome the limitations of the waterfall model, it was necessary to develop a new software development process model, which could help in ensuring the success of software project. One such model was developed tha
t incorporates the common methodologies followed in the waterfall model, as well as eliminated almost every possible/known risk factor from it. This model is referred as the spiral model or

Boehm's Model [3] .There are four phases in the Spiral Model which are: Planning, Evaluation, Risk Analysis, and Engineering. These four phases are iteratively followed one after other .However, one of the main disadvantage of the spiral model is that it requires highly skilled people and is more time-consuming and expensive [2] [3].

3. WebE Process Models

A WebApp (Web Application) encompasses complete websites, specialized functionality within websites, and information-processing applications that reside on the Internet, Intranet or Extranet [4]. WebApps deliver a complex array of content and functionality to a broad population of end-users [21]. It's not possible to fully specify what a WebApps will contain at the start of the development process because WebApps are rapidly changing. There are two major factors that should be considered the ability to maintain information and its structure and functionality [10]. These factors make WebApps development different from traditional software development. There are some aspects that make WebApps development difficult like complexity, continuous evolution, diverse group of users, developed in shorter time period and with smaller budgets [4]. By its unique features and nature, WebApps development is not an event but a process. Therefore, the ways of keeping the information up to date and once developed, ongoing updates becomes a continuous process [6].

To reduce the difficulty in building WebApps, better manage WebApps development, and to do it in a systematic and repeatable manner, there is need of a process that outlines the various phases, steps, and activities of WebApps development [11]. This process will consist of a set of controllable activities that need to carry out to develop the WebApps and then to keep its information current. Also, this process should be planned well and clearly define a set of steps

that developers can follow. Futher more, the process should be iterative to provide to the evolutionary nature of WebApps.

A suitable process model should help developers to address complexities of WebApps, minimizes risks of development, deals with the probability of changes and delivers the WeaApps within time, facilitate continual update and evolution based on feedback from users [11]. To develop a process model there is a need to identify various activities that needs to be carried out and organize them into some logical order. These activities are generally combined into phases. These phases are combined to form the overall development process as shown in Figure1 [23].The overall development process means the life cycles of WebApps. Generally, the steps followed in the development process models are same i.e. Analysis, design, implementation, maintenance and evolution [5].

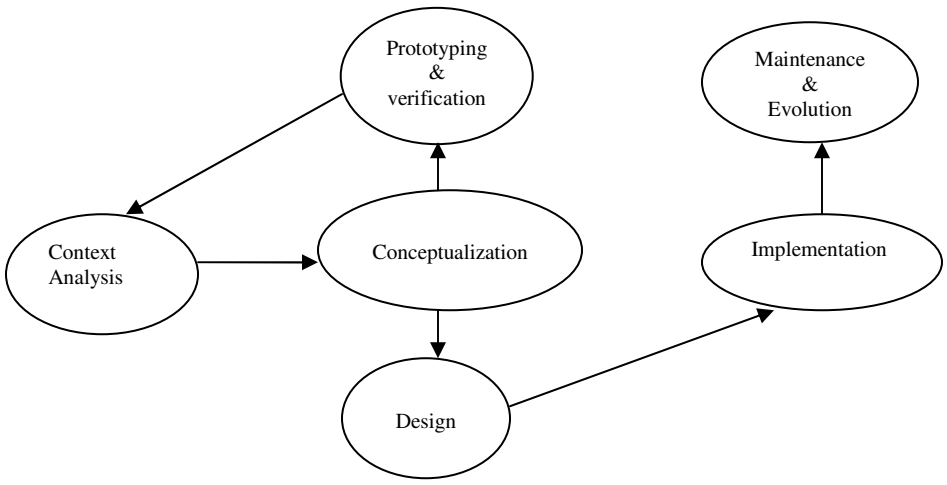

Figure: 1 Overall Development process for WebApps

For example, when developing a small application consisting of only a few Web pages, it may be sufficient to follow a reasonably linear development sequence starting with analysis followed by design, implementation and maintenance [6]. However, developing a large complex application in a relatively unknown application domain, in the implementation phase, any change in design can make the implementation process much more cost effective. In such a situation, it will be

needed to re-visit the design phase and check whether those changes could be accommodated. A process model that facilitates iteration would therefore be very beneficial [6]. Thus, it can be observed that there are various factors such as application domain, application scale and others that not only influence the set of activities to be carried out but also the way in which these are combine and undertaken.

The attributes of the WebApps decides which process model to be used. If immediacy and continuous evolution are primary attribute of WebApp, a WebE team might choose agile process model but if a WepApps is to be developed over a longer time period, an incremental process model might be chosen [4]. Figure 2 shows the small process model for WebE [4].Communication involves deep interaction and cooperation with the customer and encompasses requirements gathering and other related activities. Planning establishes an incremental plan for the WebApps. Modeling encompasses the creation of models that assist the developer and the customer to better understand WebApp requirements and the design. Coding and Testing combines both the code generation with testing that is required for uncovering errors in the code. In the delivery and evaluation phase delivers a WebApp increment is deliveredto the customer who evaluates it and provides feedback based on the evaluation.

For small WebApps the waterfall model can also be applied .But for developing large and complex WebApps requiring iterations ,the spiral model is considered to be more appropriate [6][12].

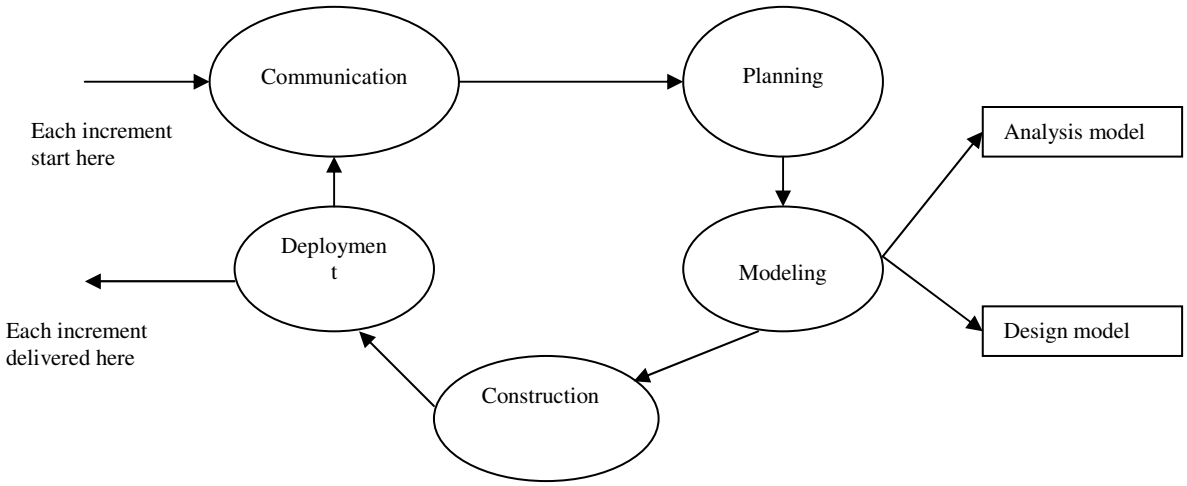

Figure: 2 Simple Process Model for WebApps

4. Factors affecting the evolvement of WebE

In this section, the factors have been identified that causes the requirement of a new engineering approach for WebApps i.e. even through the conventional software engineering was already there. In today's scenario, Internet and World Wide Web have changed the culture of information systems and the manner in which information technology is applied to solve business problems. It has brought about a revolution in information processing, system design and architecture. The scope of the systems has become larger, the number and variety of users have increased, adding complexity to user's requirements. For this reason, security of information has become a prime requirement in every aspect of web application, demanding more engineering rigor in the development of web applications..

WebE and Software Engineering are similar in terms of engineering discipline. Like Software Engineering, WebE has components such as analysis, planning, modeling, construction, deployment, testing and so on. The engineering components are almost identical to

Software engineering but have Web and Internet features. WebE uses internet and web technologies and Software engineering principles for systematic development of WebApps.WebApps have many attributes which are different from the attributes of traditional software systems [4] as given below in Table 1.

Table 1: Attributes differentiating WebApps from traditional software systems

1. Network intensiveness	2. Concurrency
3. Unpredictable load	4. Performance
5. Availability	6. Data driven
7. Content sensitive	8. Continuous evolution
9. Immediacy	10. Security
11. Aesthetics	12. Usability
13 Functionality	14. Efficient and reliable
15. Compatibility	16. Interoperability

By nature, WebApps are designed for a number of unknown users, residing at remote locations, whose requirements are not precisely known. WebApps are always under the continuous evolution due to technology upgradation and business dynamics. The frequency of changes and modifications are very high. In view of this, WebApps design must be extremely dynamic and flexible. Because of the nature and distinct requirements of WebApps, there have certain characteristics which differs the WebApps from traditional software systems [14][16]:

:

- WebApps are document-oriented containing static or dynamic Web pages.
- Open modularized architectures and rapidly changing technologies.

- Multi-user capability: A WebApps works on a network and must serve the diverse needs of users.
- WebApps contain rapidly changing contents i.e. text, graphics, audio and video clipping.
- WebApps demand aesthetics attribute i.e. good look and feel.
- Developed in shorter time frames and with smaller budgets.
- WebApps demand 24-hour access availability.
- WebApps are generally more unpredictable and not as well understood as software.
- Understanding of additional disciplines required for WebApps, such as hypertext, graphic design, information architecture
- Fine-grained evolution and maintenance
- Rapidly evolving implementation environment, encompassing various hardware platforms
- Main features of these systems are the navigational structure, the user interface and the personalization capability.
- Many different kinds of stakeholders participate in the development process: analyst, customers, user, graphical designers, and marketing, multimedia and security experts

Also, some difference between the characteristics of early, simple WebApps and the advanced WebApps given below in Table 2 [5][8][12]. The scope and complexity of current WebApps vary widely from small scale, short-lived services to large-scale enterprise applications distributed across the Internet and corporate intranets and extranets. As WebApps have evolved, the demands placed on Web-based systems and the complexity of designing, developing, maintaining, and managing these systems have also increased significantly.

Table 2 Characteristics of simple and advanced WebApps**.**

Simple WebApps	Advanced WebApps

• Simple Web pages • Less emphasis on aesthetics/user interface • Information content static • Simple navigation • Stand-alone systems • High performance wasn't major requirement • Developed by a single individual/small team • Used for information dissemination is non core applications	• Complex Web pages • More emphasis on aesthetics, user interface • Information is dynamic • Complex navigation • Integrated with database and other systems • Required high performance • Requires a large development team • Deployed in mission critical applications

With the increasing maturity in the web based systems, the complexity of WebApps is increasing. WebApps are now made to deliver a complex array of content and functionality to end users. Thus of conventional software development approaches for WebApps development are causing some problems such as

- WebApps are not delivering desired performance and quality.
- Development process becomes increasingly complex and difficult to mange and refine, and
- The development exceeds over run the projected budget and schedule.

The primary causes of these failures are a lack of vision, narrow goals, a inconsistent design and development process, and poor management of development efforts .A recent survey on Web-

based project development made by the Cutter Consortium highlighted problems plaguing large Web-based projects [9] [11][12][17][20].

Table 3: Problems plaguing large Web-based projects

Problems	Percentage of time
Delivered systems didn't meet business needs	84%
Schedule delays plagued the projects	79%
Projects exceeded the budget	63%
Delivered systems didn't have the required Functionality	53%
Deliverables were of poor quality	52%

In order to avoid WebApps failures and achieve greater success in development, there is an essential need for disciplined approaches and new methods and tools for development, deployment and evaluation of WebApps. Thus, large, high performance advanced WebApps development is a complex task. It demands a sound and disciplined process and requires consideration and systematic approach, rather than ad hoc approaches [10][17].

WebE advocates a process and a systematic approach to development of high quality WebApps and aims to bring the potential chaos in Web-based system development under control, minimize risks, and enhance maintainability and quality. It is rapidly emerging as a new discipline in its own right [10][11].

5. A WebE Process adapted from Software Process Model

In this section, the various changes and adaptations required in the conventional software engineering process have been identified to make it useful for the development of WebApps.The resulting process is known s WebE process. Software process includes all of the activities involved in the software development. The high-level activities of software specification, development, validation, and evolution are part of all software processes. While WebE process adopts and encompasses many software engineering process and principles, it incorporates many new approaches, methodologies, tools, techniques, and guidelines to meet the unique requirements of WebApps i.e. highly distributed, much more modular, and frequently modified. Developing WebApps is significantly different from traditional software development and poses many additional challenges. As discussed in the previous sections, there are differences in the nature and life cycle of Web-based systems and software systems and the way in which they are developed and maintained.

WebE is neither a clone, nor a subset of software engineering, although both involve programming and software development. While WebApps uses software engineering principles, it encompasses new approaches, methodologies, tools, techniques, and guidelines to meet the unique requirements of WebApps. Difference between WebE and Software Engineering has impact on development process as given below:

Table 4: Difference between WebE and Software Engineering

Software Engineering	WebE
• Software system has small user	• WebApps has large user range

range. • User Requirements are specific • Growth and change is small • Development budgets varies in a wide range according to the size of the company • Development time is longer • Hardware and Software environments constraints are specific. • Design and development expertise is few. • Security and legal issues are not much important • Less emphasis on user interfaces	• User Requirements are changes with time • Rapidly changing • Development budgets are small. • Development time is small. • Hardware and Software environments constraints are not specific • Design and development expertise is available in wide rang • Security and legal issues are not much important • . More emphasis on user interfaces.

There are also similarities between Web Development and Software Development [9].These includes need for methodologies, requirement elicitation, programming, testing,and maintenance of those parts that deal with programming and functionalities.

Due to the network intensiveness nature of WebApps, WebApps run on the Internet, an intranet or an extranet to meets the needs of diverse groups of users. A large number of users may access the WebApp at one time, there may need concurrency feature. Loads on WebApps are unpredictable because the number of users may vary by orders of magnitude from day to day. WebApps is, therefore, content–driven.. WebApps is to be use hypermedia to present text,

graphics, audio and video content to the end-user. In order to protect sensitive content and provide secure modes of transmission, strong security measures must be implemented throughout the infrastructure that supports a WebApp and within the application itself.

The quality and aesthetic nature of content remains an important determinant of the quality of a WebApp. Unlike conventional application software that evolves over a series of planned, chronologically-spaced releases, WebApps evolve continuously. When an application has been designed to market or sell products or ideas, aesthetics may have as much success as technical design.A good attractive design of the application generates appropriate response from users. The use of colours, presentation, layout, and creation of guiding buttons and domain-specific icons are the main requirements of WebApps.All these needs for the development of WebApps are fulfilled by the WebE. In view of the nature of the Web and WebApps , WebE is bound to be a multidisciplinary field, with encompassing inputs from diverse areas such as human-computer interaction, user interface, systems analysis and design, software engineering, requirements engineering, hypermedia engineering, information structures, testing, modeling and simulation and project management, as well as social sciences, arts and graphic design as shown in figure 3 [5][9] [10][15][17]

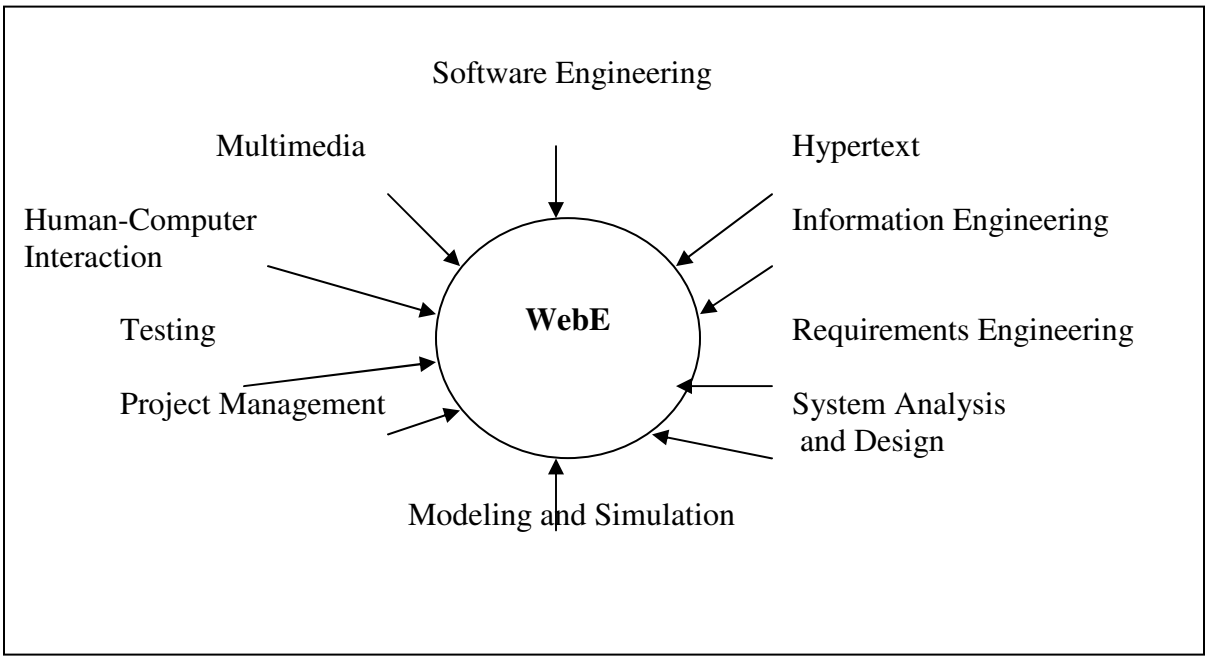

Figure: 3 WebE- A multidisciplinary fields

Thus, there are major differences in the development process of conventional software systems using software engineering and WebApps using WebE .Table 5 shows the major differences between the software activities and WebE activities [4][7][14][17][19][22].

Table 5: Difference between Software engineering activities and WebE activities

Software engineering activities	WebE activities
• Requirements analysis and definition. • Software scoping and determining its boundaries • Factoring software in different components and configurations for development, testing and	• Requirements specification and analysis • Web-based system analysis and design • Web development methodologies and techniques • Migration of legacy system to Web

integration • Planning, scheduling, execution, monitoring and control of software development • Testing at all stages and phases for quality assurance as required by the customers • Documenting the software for its uses • Implementation through demonstration, installation and execution at customer's site • Change management in pre and post implementation phase • Resource and effort estimation and its management • Risk assessment and its management • Development process management to remain within cost, time and budget • Project management for achieving software goals	environments • Web-based real-time applications development • Web-based multimedia application development • Testing, verification and validation techniques and tools • Quality assessment, control and assurance • Management of access to applications and privileges • Configuration and project management • "Web metrics" - metrics for estimation of development efforts • Performance specification and evaluation • Update and maintenance • Development models, teams, staffing • Human and cultural aspects • User-centric development • Graphics, animation and streaming • Copyright, legal and social aspects

It can be seen that the activities in the software engineering models requires some changes and adaptations to be useful for the development of WebApps as shown in figure 4.

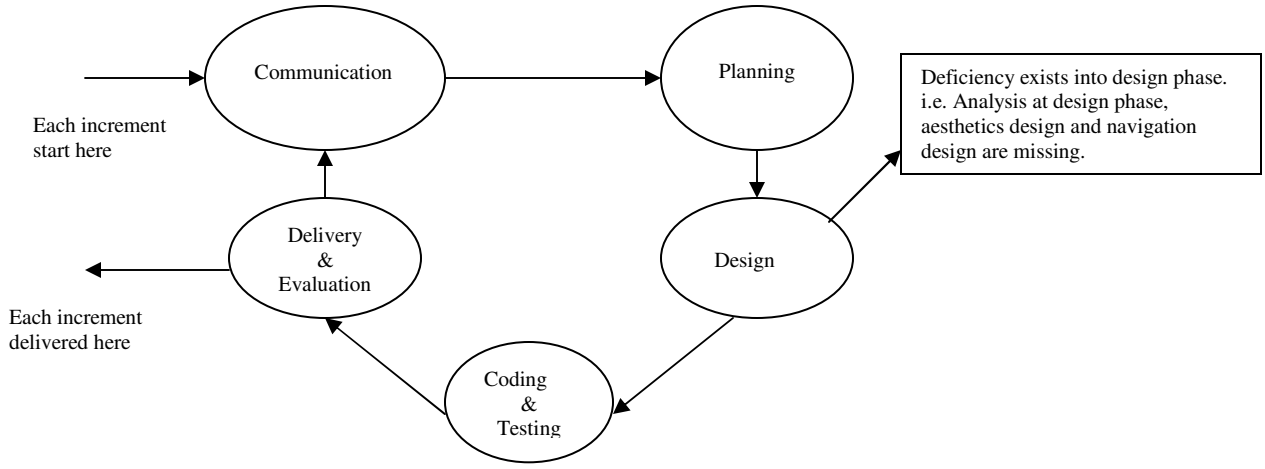

Figure 4: Software engineering process model

The resulting model with adapted group of activities is shown in figure 5.In this model; all phases are almost same as software engineering process model. But, at modeling phase, analysis modeling is required and at design model, aesthetic and navigation designs are also required. Analysis activities help to understand the detailed requirements that will allow satisfying user needs. Analysis models look at content, interaction, function, and WebApp configuration. Content analysis identifies the full range of content to be provided by the WebApp.Content includes text, graphics and images, and video and audio data. Interaction analysis describes the manner in which the user interacts with the WebApp Functional analysis defines the operations that will be applied to WebApp content and describes other processing functions that are

independent of content but necessary to the end user. Configuration analysis describes the environment and infrastructure in which the WebApp resides. Aesthetic and navigation design are missing in software engineering process model at design phase. But, these are necessarly required at design model in WebE process model .Because aesthetic design describes the look and feel of the WebApp. Aesthetics are an important and integral feature of WebApp design. A good attractive design of the application generates appropriate responses from users. It includes colors, layout, text size, font and placement etc. Navigation design represents the navigational flow between content objects and for all WebApp functions.

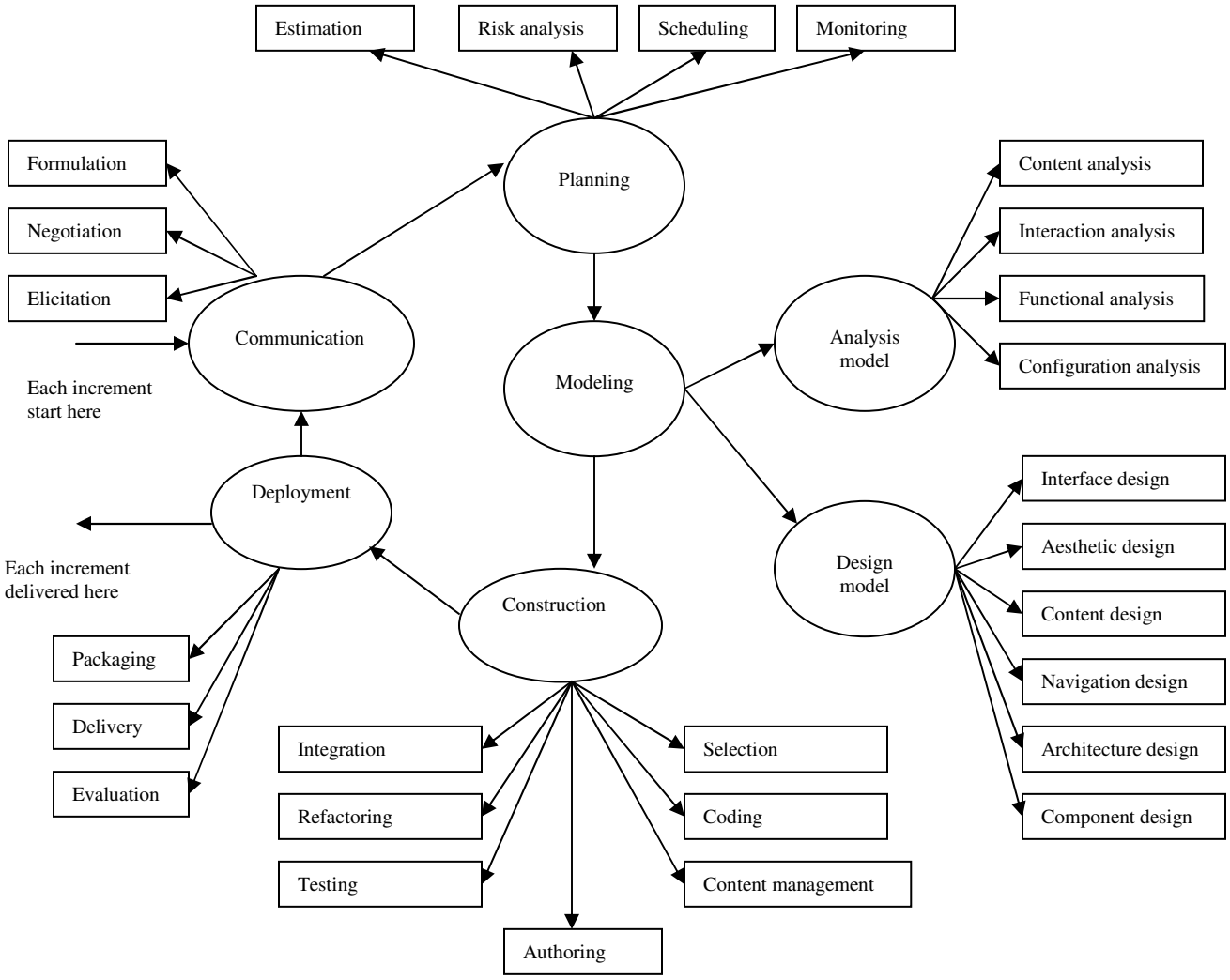

Figure 5: Resulting WebE Process Model

The advantage of resulting WebE model is that the software engineers habitual of working in software engineering methodology can easily adapt themselves to the development of web based applications. This model helps the developers to address the complexities of web based applications, minimizes the development risks and delivers the web applications within time.

6. Conclusions and Future Work

In this paper, the work has been done to identify and analyze the various factors causing the need of new approach to the development of WebApps and further a WebE model has been presented as the adaptation of conventional software engineering model. It has been found that the existing software life cycle models can be adapted to cop up with the need of web based system development. The presented WebE model reveals that the process for the development of web based systems can be considered as an adaptation of existing software engineering process only. The advantage of this understanding is that the software engineers habitual of working in software engineering methodology can easily adapt themselves to the development of web based applications. Our future work deals with the development of more precise and efficient adapted model for the development of WebApps and to also work on the model for the systematic development of semantic web based applications.

References

1. Marciniak, J.J. (ed.), 'Process Models in Software Engineering', Encyclopedia of Software Engineering, 2nd Edition, John Wiley and Sons, Inc, New York, December 2001.
2. Jalote, P, Palit, A, et al, 'A Process Model for Iterative Software Development', Journal of Systems and Software, 2003.
3. Boehm B, 'A Spiral Model of Software Development and Enhancement', ACM SIGSOFT Software Engineering Notes, ACM, 11(4):14-24, August 1986.
4. Pressman R.S., 'Software Engineering: A Practitioner's Perspective', 5th ed.,McGraw- Hill, New York, 2000, pp. 769-798.

5. Ginige, A, Murugesan, S, 'Web Engineering: A Holistic, Disciplined Approach to Web-Based System Development', 12th International World Wide Web Conference' 03, Budapest, Hungary, 2003.
6. Ginige, A., 'Web Engineering: Methodologies for Developing Large and Maintainable Web Based Information Systems', Proc IEEE International Conference on Networking the India and the World CNIW'98, Ahmedabad, India, Dec 1998.
7. Kuhnke, C, Schneeberger, J, et al, 'A Schema-based approach to Web engineering', In Proc. Of Hawali International Conference on System Scieneces-2000.
8. Ginige, A & Murugesan, S, 'Web Engineering: An Introduction', IEEE MultiMedia, vol. 8, no. 1, Jan.-Mar. 2001, pp 14-18.
9. Deshpande, Y, Murugesan, S, Ginige, A, Hansen, S, 'Web Engineering', Rinton Press, Journal of Web Engineering, Vol. 1, No.1 (2002) 003-017.
10. Murugesan, S, et al., 'Web Engineering: A new Discipline for Development of Web-Based Systems', WebEngineering2000, LNCS 2016, pp. 3-13, 2001.
11. Ginige, A, 'Web Engineering: Managing the complexity of Web Systems Development', Proceedings of SEKE 2002, Italy, ACM.
12. Ahmad, R, Li, Z, Azam, F, 'Web Engineering: A New Emerging Discipline', IEEE International Conference on Emerging Technologies, 2005,September 17-18, Islamabad
13. Gaedke, M, Graf, G, 'Development and Evolution of Web-Applications using the Web Composition Process Model', International Workshop on Web Engineering at the 9th International World-Wide Web Conference (WWW9), Amsterdam, The Netherlands, May 15, 2000.
14. Lowe, D, Henderson-Sellers. B, 'Impacts on the development process of differences between web systems and conventional software systems', in SSGRR 2001: International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, (L'Aquila, Italy, 2001), Scuola Superiore Guglielmo Reiss Romoli, 21.
15. James, E, Whitehead, Jr, 'A Proposed Curriculum For A Masters in Web Engineering', Journal of Web Engineering, Vol. 1, No.1 (2002) 018-022

16. Deshpande, Y, Hansen, S, 'Web Engineering: Creating a Discipline among Disciplines', Special Issue on Web Engineering, IEEE Multimedia, 8 (2), April-June, 2001, pp. 82-87.
17. Ginige, A, Murugesan, S, 'Guest Editors' Introduction:The Essence of Web Engineering-Managing theDiversity and Complexity of Web Application Development', IEEE MultiMedia, vol 08, no.2, pp. 22-25, April-June 2001
18. Weippl, E, Ibrahim, I, et al, 'Web Engineering For Intranets: Rethinking Software Engineering', Fourth International Conference on Information Integration and Web-based Applications & Services (iiWAS2002). 2002. Bandung, Indonesia.
19. Ginige, A, Murugesan, S, 'The Essence of Web Engineering: Managing the Diversity and complexity of web Application development', IEEE Multimedia, 8(2), 2001.
20. Kappel, G, Michlmayr, E, et al, 'Web Engineering-Old wine in new bottles?', 4th International Conference on Web Engineering (ICWE2004), Munich, Germany, July 2004.
21. Eldai, O, Raviraja S, et al, 'Towards a New Methodology for Developing Web-based Systems', World Academy of Science and Technology, 46, 2008.
22. Jawadekar,W, Software Engineering: principles and practice, McGraw- Hill, New York, computer engineering series, 2004.
23. Fraternali, P,'Tools and Approaches for Developing Data-Intensive Web Applications: A Survey', ACM Computing Surveys http://sys192.cs.washington.edu/Related/p227-fraternali.pdf.