

## Unified V- Model Approach of Re-Engineering to reinforce Web Application Development

Poonam Dhiman<sup>1</sup>, Nishu Singh<sup>2</sup>, Kaushal Kumar<sup>3</sup>

<sup>1</sup>(Software Engineering, Delhi Technological University, India)

<sup>2</sup>(Software Engineering, Delhi Technological University, India)

<sup>3</sup>(Software Engineering, Delhi Technological University, India)

---

**Abstract:** *The diverse and dynamic nature of elements and techniques used to develop Web Application, due to the lack of testing technique and effective programming principles which are used for implementing basic software engineering principles, and undisciplined development processes insure by the high pressure of a very short time to satisfy market request to develop Web application. This paper represent approaches of reengineering in web that how reengineering process can be carried out to evolution activities in legacy system as well we propose the V-model for re-engineering process. This paper presents the need of the technologies and approaches for building new web-services from existing web-applications. In this paper we present the processing of V-model for Reengineering in web application which is the extension of V-model used in software domain. In our approach V-model incorporates with the methodology throughout the phases of web development process to re-engineer the web system.*

**Keywords:** *Re-engineering, reverse engineering, forward engineering, V-model, application migration.*

---

### I. Introduction

The technological evolution of the last year has made the Web Service of the ideal platform for the appropriate support for their delivery and the development of Web-based applications. According to research [1] and [2] the development of a Web application is a multi-faceted Activity, involving not only technical but also organizational, managerial and even social and artistic issues. Web application development refers a set of activities which applied in order to develop a web application of high quality having awaited characteristics, and to accomplish this development efficiently and coherently. Web engineering is an important topic in these days and is gaining more attention. It is fast developing area and not existing from centuries. Web maintenance and web reengineering both falls in the scope of web engineering. The World Wide Web has ability to ubiquitously provide and gather information to the economy globalization together with the need of new marketing strategies has enormously boosted the development of Web Applications (WA). Software application is the backbone of the WWW infrastructure.

Most web applications are developed under proper schedules and in a rapidly evolving environment. The development is often ad-hoc in nature and the applications are poorly structured and poorly documented. Maintenance of such applications becomes problematic and increases the complexity of the web application grows. Creating appropriate design and architecture models is the solution to managing this complexity and supporting evolution of web applications. Researchers have identified the need to re-engineer the system already existing web applications into abstract design models.

Web application must cope with an extremely short development evolution life cycle: A high level of flexibility, maintainability, and adaptability are actually necessary to compete and survive to market inflation. Unfortunately, to accomplish tight timing schedules to deliver web services, web applications are usually directly implemented without producing any useful documentation for their maintenance and evolution, and so those requirements are never be satisfied. In order to satisfy a growing market request for Web applications and to deal with their increased technological complexity, we require specific methods and techniques able to support a disciplined and more effective development process. However, the high time pressure often forces the developers to implement the code of the application directly, without using disciplined development process, and this may have black effects on the delivered quality and documentation of the Web application. This situation same as one occurring for traditional software produced in a short time, without respecting software engineering principles and using no disciplined development process. Poor quality and poor documentation must be considered the main factors essentially abortive and expensive maintenance, unattainability of applying more structured and documentation-based approaches. Reverse engineering methods, techniques and tools have proved useful to support the post-delivery life-cycle activities of traditional software systems, such as maintenance and evolution. This paper has five sections. In this paper Section 2 presents background

information including a conceptual knowledge of the main features of a Web application, and related works on analysing existing Web applications. Section 3 describes how the re-engineering paradigm or approaches can be used to define and implement Web application re-engineering processes, while Section 4 presents the V model of re-engineering process, proposed to achieve the comprehension of existing web applications. While Section 5, finally, provides concluding remarks.

## **II. Reengineering**

Reengineering is the analysis of existing software system and modifying it to constitute into a new form. Chikofsky and Cross define reengineering as ‘the examination and alteration of a subject system to reconstitute it in a new form and subsequent implementation of that form’ [3]. According to IEEE Std. 1998 ‘A system changing activity that results in creating a new system that either retains or does not retain the individuality of the initial system’ [4].

### **2.1 Nature and Scope of Reengineering**

When maintenance cost is not feasible, we go for reengineering the software system. Reengineering makes the

software system new. Reengineering has the following three stages.

1. Reverse engineering
2. Transformations or Transfiguration
3. Forward engineering

#### **2.1.1 Reverse engineering**

Reverse engineering is the process of analysing the system which helps in recovering its design and specification. Reverse engineering is different from re-engineering. Reverse engineering is a process of analysis to determine the relationship of the system component and create the components of the system in another form or in a higher level of abstraction. The program itself is unchanged by the reverse engineering process. The objective of reverse engineering is to obtain the design or specification of a system from its source code although the objective of re-engineering is to produce a more maintainable system. Reverse engineering is used to produce a better system and it is a part of the re-engineering process. Reverse engineering is used during the re-engineering process to recover the program design specification which engineers use to help them understand a program before re-constructing its structure. However propose of a reverse engineering process for web encompassing the following phases:

1. Static Analysis
2. Dynamic Analysis

#### **2.1.2 Static Analysis**

The number and the importance of Web applications have increasing rapidly over year by year. At the same time, the quantity and impact of security vulnerabilities in such applications have grown as well [5]. Web application security is accomplished by static analysis and runtime analysis. Web application security has been great challenge for this static analysis tool such as ASPWC is used to detect the attack and vulnerabilities based on taint analysis [6]. Static analysis does not require the execution of the application. It recovers web application architecture components and the static relations among them. HTML files, directory structure, scripting language sources as well as any other static information (e.g., database connections, use of applets/servlets) are processed. HTML pages and page sub elements (frames, forms, widgets) composing the given page are localized, classified and recorded in an intermediate representation. Central to the reverse engineering process is the mapping between web application elements and object oriented entities, according to Conallen proposals [7] [8][9].

#### **2.1.3 Dynamic analysis**

The dynamic analysis phase relies on the static analysis results. The web application is executed and dynamic interactions among the components are recorded. It is performed observing the execution of the web application, tracing to source code any event. Traced events are those observed by the user or related to components external to the web application (e.g., third party databases or WEB sites). Events are the HTML pages/frames/forms visualization, the submission of forms, the processing of data, a link traversal, or a database query, etc. All elements responsible of these actions (typically links, scripts, applets) are localized. The sequences of actions fired by an event, deriving from web application code control flow (e.g., access to a database following a user form submission) or from user actions (e.g., clicking on a link or submitting a form) are associated to sequences of messages exchanged between the objects of the web application.

## **2.2 Transformations and Transfiguration**

This phase involves the transition, alteration, modification, reformation, reconstruction, remodelling of the web application system. Web architecture is altered. It is modified, improved to cope with the new technology and new environment. It is the architecture designing stage. In web these transformation can be accomplished by changing in transaction design, adaptations of the code itself to the new computing platform, redesign of the UI to better suit the new constraints of the target platform.

## **2.3 Forward engineering**

In this stage, we move from higher level of abstraction to low level. In this stage web application integrated according to new design. It is the traditional process of moving from higher level abstraction and logical implementation to independent design to the physical implementation of a system. It follows a sequence of going from requirement through designing its implementation.

## **III. Approach to Reengineering for web**

Reengineering generally include some form of reverse engineering to attain more abstract description which is followed by some form of forward engineering or restructuring. This may include modification w.r.t new requirements not met by original system. With the advancement of technology continuous changes are introducing in web industry and hence web application need to be cope with the latest technologies and competing in market. The need to fulfil the market and additional requirement may lead to need for web reengineering where the web application system is transformed from one state to another state.

### **3.1 Reengineering of web pages**

Reengineering of web pages can be accomplished by detecting and analysing the interaction of objects and then transforming these objects for the adaption of new platform itself and generating the source code into new language. There are several presentation models that can be used to transform into another model for different context supporting flexible reverse engineering process. The detection and transformation phases of the reengineering process can be governed as

- Percolating the objects, tags and elements of the web pages that include selection of any HTML item, with given properties that require to keep all control mechanism and discarding the unwanted tags and elements from web pages.
- Transformation in the layout options and relationships that include alignment, balance (horizontal or vertical balancing), centric which depend upon the position of the objects on the page.
- Content updating of the Webpages according to the requirement changes, market evolution, usage and owner of website.
- Clustering of Webpages allow for the information obtained by analysis (static and dynamic) of reengineering process can be used to produce a graph whose nodes represent the set of Web application objects, and whose links specify the interaction between these objects. In [10], this kind of graph is called WAG; Web Application connection Graph. WAG, analysis may support the comprehension of the application. However, since this graph may be large (in terms of the number of nodes and edges) even in the case of small size Web applications, in order to simplify the analysis of large WAG graphs, some kind of automatic clustering [11] can be used to decompose this graph into smaller cohesive parts. In the third step of the reverse engineering process, the automatic clustering approach proposed in [10] is applied, in order to group software items of a WAG into meaningful (i.e. highly cohesive) and independent (i.e. loosely coupled) clusters. This clustering approach evaluates the degree of coupling between entities of the application (such as server pages, client pages and client modules) that are interconnected by Submit, Build, Link, Load in Frame, Redirect, and Include relationships.

Among several clusters obtain by clustering algorithm we choose the most optimal cluster by evaluating the degree of intra-connectivity and degree of inter-connectivity (minimizes intra- connectivity and maximizes the intra-connectivity). The 'optimal' configuration is considered the most suitable for including clusters implementing functions at higher levels of abstraction than that of the cluster's single items. Validation of the clusters, based on a Concept Assignment Process [12] has to be carried out.

- Grouping of objects that are close to each other because they are semantically related this process is called association and ungroup objects that are isolated without any connection when they are unrelated which is called dissociation.

### **3.2 Transaction Reengineering**

In a transaction-oriented Web site, the user executes a series of activities in order to carry out a specific task. One of the reasons for the success of e-commerce business today is the transactional behaviour that the Web offers. Business processes are realized by means of transactions, which in this context can be

interpreted as high-level workflows corresponding to user tasks (e.g. purchasing an airplane ticket). The formalism underlying the process is a revised version of the UWA Transaction Design Model [13], which is the portion of the UWA framework that focuses specifically on the design of Web application transactions.

The UWA design framework provides a complete design methodology for ubiquitous Web applications that are multi-channel, multi-user, and context-aware. The UWA design framework organizes the process of designing a Web application into four main activities [14]. (1) Requirements elicitation [15]. (2) Hypermedia and operation design [16]. (3) Transaction design [17] and (4) customization design [18]. Using the UWA methodology, the transaction design process produces two conceptual models: the Organization Model and the Execution Model. The Organization Model describes a transaction from a static point of view. It uses a particular UML class diagram [19] in which the Activities involved in the transaction are represented by class stereotypes, which are arranged to form a tree. The Activity represented by the root of the tree corresponds to the entire transaction; component activities and sub-Activities are intermediate nodes and leaves of the tree that represent sub- transactions and elementary activities, respectively. The Execution Model of a transaction defines the possible execution flow among its component activities and sub-activities. It is a customized version of the UML Activity Diagram [6]. The sequence of activities is described by UML Finite State Machines, in which activities and sub-activities are represented by states (ovals), and execution flow between them is represented by state transition (arcs).

**3.4 Application Migration Reengineering**

Migrating applications to the newer technologies can give business a leading edge by removing inefficient workflow and processes while preserving original objectives, model and investment. We can help enterprises in migration of the legacy systems from old technologies to present day platforms. Reengineering must keep into consideration the strategically designed to overcome the cross platform compatibility challenges.

Due to upcoming advance technology and growing business states, there is need for the migration of legacy software systems to new technologies and environments. There are different kind of legacy system re- engineering services that includes language and database migration, platform-to-platform porting and system redevelopment.

A web application must follows the enterprises standard and rules implemented in a legacy application, while transforming those to new business and architecture requirements, to produce a flexible, tested or validated modified system. Re-engineering and Migration Benefits are the saving time and effort, Enhancements in operational efficiency, Benefits of the latest technologies and platforms

Web application migration can include following services: Legacy application and reusable component analysis  
 New technology and platform inspection  
 Platform, language, database and architecture migration  
 Design, development and integration  
 Version rendering Functionality enhancement Application and process organising

Data Migration	VBA, VB.NET SQL Server 6.5 / 2000 to SQL Server 2005/2008 MS SQL Server to ORACLE
Architecture Migration	Client Server to N-TIER Legacy to Web Services Client Server to SOA ( Service Oriented Architecture) Legacy to Web Enablement

Table 1: web migration services

**3.5 Graphic design re-engineering**

Re-engineering transforms a final user interface into a logical representation that can be changed to direct forward engineering to allow a user interface from one computing platform to another one which is having minimum effort. Re-engineering is used to adapt a UI to another format. To change user interface it is not mandatory to start developing it from scratch. Some transcoding tools [20][21][22][23] automatically transform a UI code from the original platform to a target platform. Portability and transcoding exhibits some limitation as they do not need to consider constraints imposed by the target platform such as: operating system, programming language, screen resolution, interaction capabilities To overcome these shortcomings UI reverse engineering process can be combined with UI forward engineering process to produce not only more usable UIs in a logical way, but also to benefit from the reverse engineering to port a UI to any other target platform.

The Cameleon Reference Framework [24] locates UI development steps for context-sensitive interactive applications. A context is defined an element of the environments set considered for the interactive system, element of the platforms set considered for the interactive system and an element of the

users set for the interactive system. A simplified version (Fig. 1) structures development for two contexts of use, here for two platforms: the one on the left represents the source and the one on the right represents the target. The development process can be decomposed into four steps:

Task and concepts describe the various tasks to be performed and the application-oriented notions according to tasks to be performed. Logical UI is a basic symbol and notation for manipulation of the application concepts and routine in a way that does not dependent upon the perspective interacts present on the targets. The elements used in the logical UI are abstractions of existing product. Physical UI represents a logical UI into real Interaction Objects to define product layout and interface navigation scheme. This interface is now composed of existing UI product. Final UI produced at the last step of the objectification supported by a multi-target development environment and represented as source code.

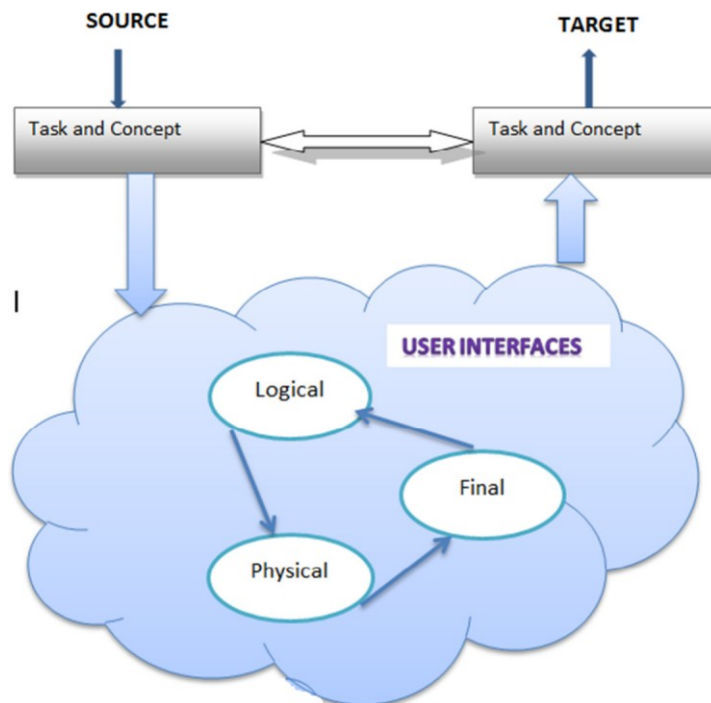


Figure1: UI development steps

#### IV. V model for web reengineering

The Re-engineered product goes through a complete web development life cycle and therefore it becomes mandatory for it to pass through complete testing cycle. The legacy system or product is transformed in new form by various means. The below figure illustrates the V-model for the Re-engineering process. A V-model as described below is proposed for designing the testing strategies for this category. Similar to the traditional V- model, left side of the Re-engineering V-model describes the stages of the design and coding and right side defines the corresponding stages of validation process. It has following phases:

##### a) Requirement gathering for new web application

The first step involves the collection of the new requirements. This will list out the key points why reengineering is required for the software under consideration. Client get start discussing with the web development team about the newly generated requirement due to market evolution, technology changes and for the product improvement for better performance. System is re-engineered in order to incorporate the new business requirements which involve functional and non-functional requirement. In this phase developer may make check list that deals with the various reason of re-engineering is required.

- Better performance
- Code restructuring
- New platform support
- Data migration
- New business requirement



-Requirement change

This phase require interviews with the client , mails and supporting documents given by the client, discussion notes, online chat, telephonic conversation, model sites/application etc.

Requirement analysis is carried out with new objective for re-engineering, cost involved in changes, supporting document and the approval.

Moreover the analysis should cover all the aspects on how the web application is going to join the existing system. The analysis should be done with in short time span having descriptive information. The analysis should be cost effective. To achieve this, the analyst should concern the designers, developers and testers to come up with an optimised work plan.

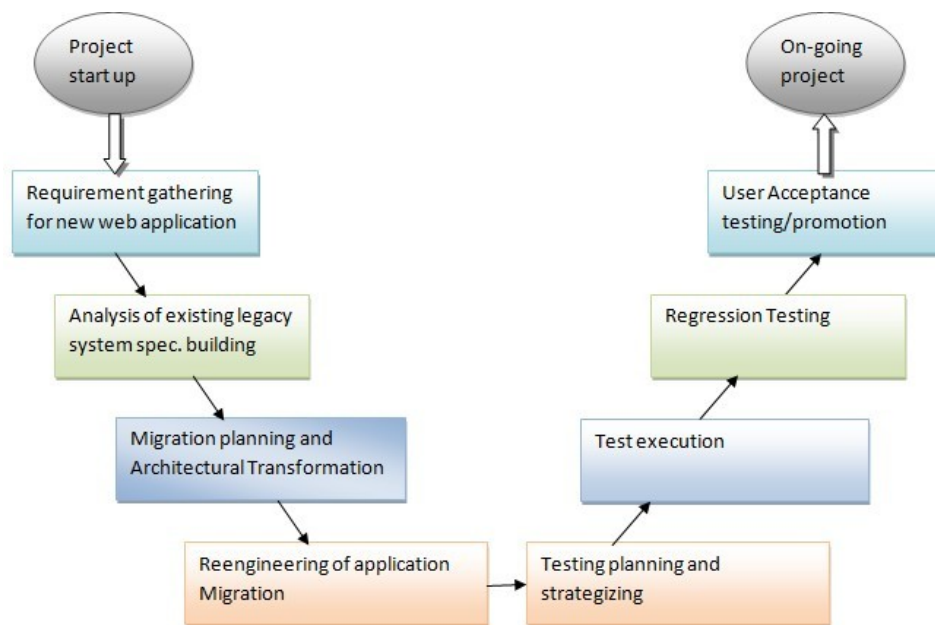


Figure 2: Web re-engineering V Model

b) Analysis of existing legacy system/specification building

The second stage is the study of the legacy system functionality and underlying design and come out with the difference with new functions. The nature of re-engineering is to improve or transform existing system so it can be understood, controlled & reused as new system. Web re-engineering is vital to restore & reuse the things inherent in the existing system, put the cost of system maintenance to the lowest in the control & establish a basis for the development of system in future.

Web application broadly classify into two forms static application and dynamic application. static application implemented in HTML, & dynamic application provide client server interaction and consist of DHTML pages, JAVA server pages, java servlet, PHP, CGI, ODBC, JDBC etc. So in this phase we carefully analyse both perspective of web application (static & dynamic) and generate WAG graph for reusable components and objects for re-engineering. This phase may consist of three major steps

-Identification of reused Components

-Encapsulation of Identified Components to new system

-Analyse interfaces of the recovered components and define specification

Identification of legacy component aims to identify web components from legacy systems. The identified components should conform to specific user requirements that should relate to new functionality, access and manipulate data, are free of side effects, and comply with specific pre and post conditions. We encapsulate the recovered legacy system components to collections of object classes that encapsulate specific functionality of the legacy system. As an object encapsulates into legacy system flexible systems can be easily designed and implemented using the re-engineering paradigm. Furthermore, we analyse interfaces of the recovered components and define a specification to represent their interfaces. The service specification provides standard, enriched, and well-understood information about the interface and functionality of the offered services.

c) Migration planning and architectural transformation

This phase addresses migration planning that is how to move from the start to the Target Architectures by finalizing a detailed Implementation and Migration Plan. The objectives of migration planning Phase are to:

- Finalize the Architecture Roadmap and the supporting Implementation and Migration Plan.
- Ensure that the Implementation and Migration Plan is coordinated with the business approach to managing and implementing change in the business's overall change.
- Ensure that Transition Architectures is understood by key stakeholders
- Estimate Resource Requirements, Project Timings, cost estimation for introducing change

#### d) Re-engineering of application migration

When existing systems become redundant, business switch from legacy systems to modern and new systems built on the latest technology / platforms. This switch is usually time consuming and expensive. A cost effective alternative to such scenarios is to reengineer, migrate or port the legacy systems into the latest technology / platforms.

Application re-engineering: In this we re-architect the product using new platforms and technologies such Web 2.0.

Technology Migration: This includes enterprises migrate applications to corporate standards and migration of products from older legacy technologies to newer technologies to ensure integration with other tools.

Application Server Migration: To take care of all cross-platform compatibility challenges, while the Client stays focused on product innovation.

Database Migration: This include migration of non-relational databases to industry-standard relational databases such as DB2, MS-SQL Server, Oracle, MySQL, thus increasing business agility.

Migration of Middleware technologies: It helps to migrate from legacy systems to new industry standard systems using implementation of middleware technologies (Server-side and Client-side) web services and others.

Code restructuring: To improve the coding paradigms it provides easy way of working. Code restructuring paid more attention in adding new features and re-factoring. It deals with continuously refactoring of web application which gives more flexible and maintainable code—Joshua Kerievsky, Refactoring to Patterns [2].

#### e) Test planning & strategizing

The stage will include the test planning & test cases preparation if required as per the new requirements and strategizing the test execution for functional and non-functional areas. Test strategizing play an important role in carrying out the entire test execution program and involvement of high business risk, huge investments and mission critical systems make it important to strategizing the test phase. The best way is to identify the risky areas and the failure rate and then develop a test strategy

#### f) Test execution

This stage carry out the functional test execution as per plan defined in the previous stage. Test execution carry out performance testing, if the major design changes are there or the new requirements are related to improvement of performance. It test all the links in web pages, database connection, forms used in the web pages for submitting or getting information from user, Cookie testing, Test for navigation, Content checking, Interface Testing include Web server and application server interface, Application server and Database server interface. Web testing includes functional testing, usability testing, interface testing, compatibility testing, performance and security testing. Performance testing is an used to determine the responsiveness, throughput, reliability, and scalability of a system under a given workload. Web application should sustain to heavy load. Web performance testing should include Web Load Testing and Web Stress Testing.

#### g) Regression testing

Regression means retesting the effect of change in other parts of the web application. Test cases are executed again and again in order to check whether previous functionality of application work appropriately and changes made have not introduced any new bugs or error. This test can be performed on a new reconstructed system when new functionality added to it. A regression testing plan covers the updated functionalities. Many automated tool for regression testing is available for web application.

#### e) User Acceptance testing

The purpose of user acceptance testing is to make sure that your application meets the user's expectations. It ensures that the application is ready to deploy services and change has been done effectively.

The activities for user acceptance testing ensure browser compatibility, make sure that mandatory fields are given data in forms, check for time outs and field widths, and make sure that proper control is used to input data.

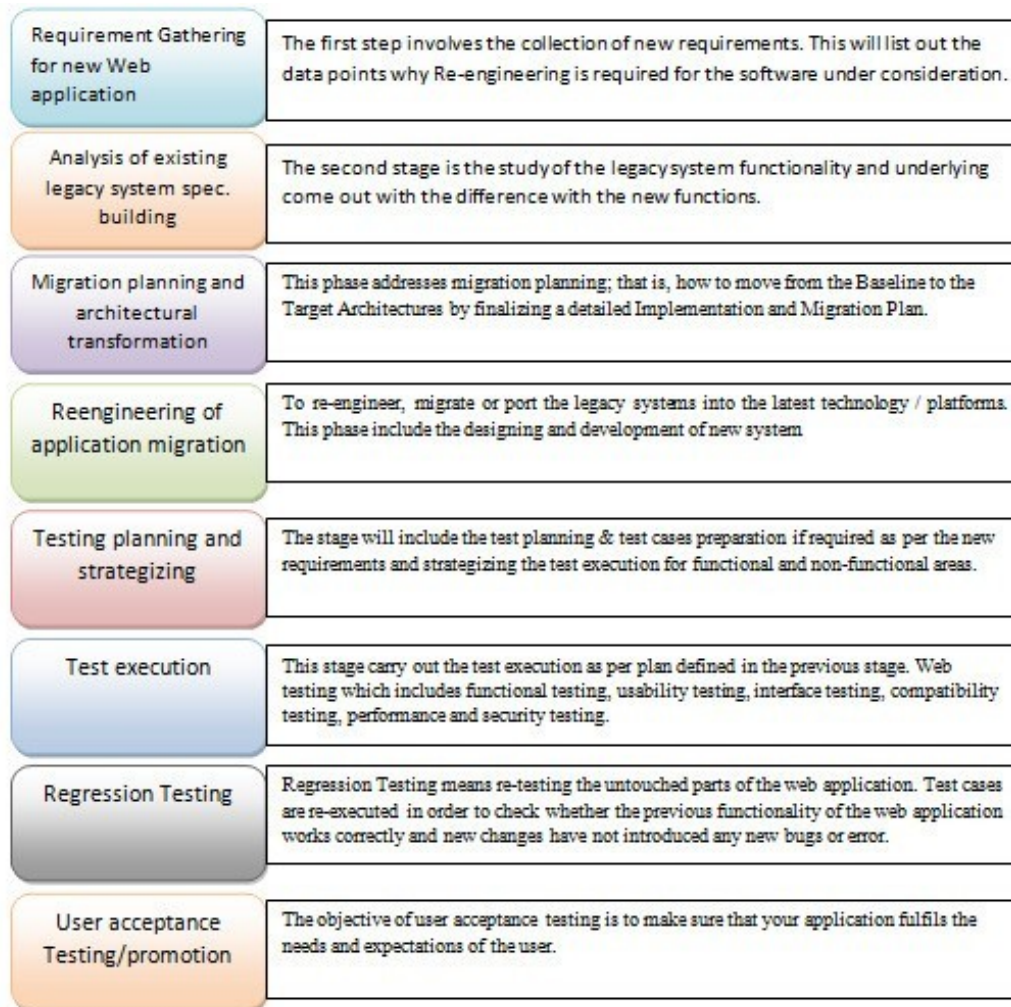


Figure 3: Description of V-Model stages

## V. Conclusion

Web application development process is very complex and it faces a lot of challenging requirements. It focuses more on planning, web architecture, system design, testing, evolution and continuous /frequent update and maintenance of the system as per requirement. The problem becomes more complex when we require maintaining it by adding new functionality, adapting it to the new platform or improving its performance. Hence, therefore there exists the re-engineering of web application with the objective being expedite maintenance process. Web reengineering is used to develop an associated and communal web application that is related to existing legacy component and also partially replaces them. In this paper, we present the web re-engineering approaches defining the different re-engineering processes that facilitate legacy evolution in web environment. The different re-engineering processes like re-engineering of web pages, transaction re-engineering, application migration reengineering, graphic design reengineering gives the direction to reconstruct, refactor and reengineer the legacy web system. In this paper we proposed the V-model for web re-engineering. V-model for re-engineering provides effective and easy way of reconstruction and re-testing web application, it also provides flexibility and reusability that takes into consideration all the user and business standard. The proposed V-model of re-engineering process is proposed for designing the testing protocol for web application development. By introducing V- model into web reengineering process, it increases the maintainability and the effectiveness of the website because V-model led to better validation and verification and produce in accordance to web development life cycle. The structure of V- model enables testing process starts from unit testing to acceptance testing. Re-engineering V-model will save the time for reconstructing or refactoring the web due to strong testing and validation.



**References:**

- [1] P. Fraternali, Tools and Approaches for Developing Data-Intensive web Applications: a Survey, ACM Computing Surveys, 1999. [2] S. Selmi, N. Kraiem, and H. Ben Ghezala, Toward a comprehension view of web engineering, 2005.
- [3] E. Chikofsky and J.H.Cross, Reverse Engineering and Design Recovery: A Taxonomy, IEEE Software Engineering journal, (Jan. 1990), pp 13-17.
- [4] IEEE Std 1219-1998, In IEEE Standards Software Engineering, 1999 Edition, Volume Two, Process Standards, IEEE Press.
- [5] Jovanovic, N. ; Secure Syst. Lab., Tech. Univ. of Vienna ; Kruegel, C. ; Kirda, E., a static analysis tool for detecting Web application vulnerabilities, Security and Privacy, 2006 IEEE.
- [6] X. Zhang ; Comput. & Inf. Eng. Coll., Hohai Univ., Nanjing, China ; Z. Wang, e-Business and Information System Security (EBISS), 2010 2nd International Conference on 22-23 may 2010.
- [7] J. Conallen, Building Web Applications with UML, Addison- Wesley Publishing Company, Reading, MA, 1999.
- [8] J. Conallen, Modelling web application architectures with uml, Communications of the Association for Computing Machinery, 42(10), October 1999.
- [9] J. Conallen, Modelling web application with uml, White paper, Conallen Inc. <http://www.conallen.com/whitepapers/webapps/Modellingwebapplication.htm>, March 1999.
- [10] D. Lucca GA, Fasolino AR, De Carlini U, Pace F, Tramontana P. Comprehending, Web applications by a clustering based approach, Proceedings 10th Workshop on Program Comprehension. IEEE Computer Society Press: Los Alamitos CA,2002; 261–270.
- [11] N. Anquetil, T. Lethbridge, Experiments with clustering as a software remodularisation method, Proceedings 6th Working Conference on Reverse Engineering. IEEE Computer Society Press: Los Alamitos CA, 1999; 235–255.
- [12] T. Biggerstaff, B. Mitbander, D. Webster, Program understanding and the concept assignment problem. Communications of the ACM 1993; 37(5):72–83.
- [13] D. Distanto, Reengineering Legacy Applications and Web Transactions: An extended version of the UWA Transaction Design Model, Ph.D. Dissertation, University of Lecce, Italy. June 2004.
- [14] UWA (Ubiquitous Web Applications) Project, “Deliverable D3 Requirements Investigation for Bank121 pilot application”, <http://www.uwaproject.org>, 2001.
- [15] UWA (Ubiquitous Web Applications) Project, Deliverable D6: Requirements Elicitation: Model, Notation and Tool Architecture, 2001. [www.uwaproject.org](http://www.uwaproject.org).
- [16] UWA (Ubiquitous Web Applications) Project, Deliverable D7: Hypermedia and Operation design: model and tool architecture, [www.uwaproject.org](http://www.uwaproject.org), 2001.
- [17] UWA (Ubiquitous Web Applications) Project, Deliverable D8: Transaction design, [www.uwaproject.org](http://www.uwaproject.org), 2001. [18] UWA (Ubiquitous Web Applications) Project, Deliverable D9: Customization Design Model, Notation and Tool Architecture, [www.uwaproject.org](http://www.uwaproject.org), 2001.
- [19] G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modelling Language User Guide, (Rational Corporation Software), Addison-Wesley.
- [20] M. Moore, Representation Issues for Reengineering Interactive Systems, ACM Computing Surveys Special issue: position statements on strategic directions in computing research, Vol. 28, No. 4, Dec 1996, article # 199, ACM Press, New York, NY, USA.
- [21] M. Moore and S. Rugaber, Using Knowledge Representation to Understand Interactive Systems, in Proc. of the Fifth International Workshop on Program Comprehension IWPC'97 (Dearborn, 28-30 May 1997), IEEE Computer Society Press, Los Alamitos, 1997
- [22] G. Mori, F. Paternò, C. Santoro, Tool support for designing nomadic applications, Proc. of the 2003 international conference on Intelligent user interfaces, Jan 2003, (Miami, USA), ACM Press, New York, USA, pp141-148
- [23] L. Paganelli, F. Paterno, Automatic reconstruction of the underlying interaction design of web applications, in Proc. Of the 14th international conference on Software engineering and knowledge engineering, (July 2002, Ischia, Italy), ACM Press, New York, USA, pp 439 – 445.
- [24] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, J. Vanderdonckt, Plasticity of User Interfaces: A Revised Reference Framework, in Proc. of 1<sup>st</sup> International Workshop on Task Model and Diagrams for user interface design Tamodia'2002 (Bucharest, 18-19 Jul 2002), INFOREC Publishing House Bucharest, Romania,2002.