



# Journal of Internet Banking and Commerce

*An open access Internet journal (<http://www.arraydev.com/commerce/jibc/>)*

*Journal of Internet Banking and Commerce, August 2011, vol. 16, no.2  
(<http://www.arraydev.com/commerce/jibc/>)*

## Web Development Life Cycle: A New Methodology for Developing Web Applications

---

**Aaron M. French**

**Assistant Professor, Kyungpook National University, Daegu, South Korea**

**Postal Address: 1370 Sankyuk-dong, Buk-gu, Daegu 702-701, South Korea**

**Email: [afrench@knu.ac.kr](mailto:afrench@knu.ac.kr)**

Aaron French is an Assistant Professor of MIS in the college of Business at Kyungpook National University in Daegu, South Korea. He received his Ph.D. in Business Information Systems at Mississippi State University. His research interests include eCommerce, social networking, cross-cultural studies and technology acceptance.

---

### **Abstract**

Service Science is the basis of information system and web services that ascribe to the provider/client model. This paper developments a methodology that can be used in the development of web services such as websites, web applications and eCommerce. The goal is to development a methodology that will add structure to a highly unstructured problem to assist in the development and success of web services. The new methodology proposed will be called the Web Development Life Cycle (WDLC) and adapted from existing methodologies and applied to the context of web development. This paper will outline in detail the proposed phases of the WDLC.

**Keywords: Web Development, Application Development, Systems Development, eCommerce**

© Aaron French, 2011

---

## **INTRODUCTION**

The Internet, and more specifically eCommerce, has impacted the world and giving rise to new business ventures and increased competition. This trend is impacting all countries both large and small. Developing countries contain a major growth in the use of e-commerce, containing over 80% of the world's population (Molla and Heeks, 2007). With new trends follows new problems that must be addressed.

Traditional applications have followed a series of methodologies that make application development a highly structured problem. With the advent of the Internet and use of Web applications, the structured problems that programmers once faced have become unstructured in the new web-based environment. Most web application development relies on the experience of individual developers without any rigorous or systematic approach (Gellersen, 1999). There is a high demand for a systematic approach to developing and managing web information systems. There are many different methodologies being used to suit the needs of programmers but no one standard methodology that is accepted by all the software engineering community (Escalona et al, 2002).

The purpose of this paper is to review the literature and develop a new methodology that will encompass all the needs of the organization combined with the development needs of the web environment. By examining existing methodologies currently used by developers and comparing them against traditional methodologies such as the System Development Life Cycle (SDLC), this research will evaluate the SDLC phases that may still have some validity and where the gaps in these legacy methodologies lie. Through extensive research, this research will identify development needs to create a comprehensive set of phases that can be used to create effective and efficient web applications. This new methodology aims to add structure to the highly unstructured nature of web application development.

The following section will review the literature describing the traditional application development methodologies and comparisons of methodologies used to create web applications. Following the literature review, the Web Development Life Cycle (WDLC) model will be constructed based on the requirements identified in the literature review. Each phase of the WDLC will then be described in more detail.

## **BACKGROUND LITERATURE**

Internet based applications in such domains as electronic commerce, digital libraries, and distance learning are characterized by an unprecedented mix of features that makes them radically different from previous applications of information technology. (Myers et al, 1996). The characteristics of Web applications in general are significantly different from various other kinds of applications, including the stakeholders and features of web applications (Escalona and Koch, 2004). The stakeholders participating in the development process for web applications may include systems analysts, customers, users, graphic designers, marketing coordinators, multimedia personnel, security experts and so on. Typical features of internet based applications consist of the navigational structure, user interface, and personalization. A previous study displayed an analysis of a typical web development team consisting of 31% software engineers, 31% creative design, 20% management, 9% business experts and 9% domain experts (McDonald and

Welland, 2001). The term 'backwards' is often used to describe individuals who live and perform tasks in traditional methods rather using modern techniques and moving forward with progress (Graham, 2008). While many organizations are joining the revolution of the Internet and take advantage of e-commerce, the methods for developing these solutions could be considered backwards. This research will development a methodology for developing web applications to help progress the development and forward movement of a non-structured environment towards a structured process for development. Two lines of research will be explored to fill in the gaps between traditional application and web applications: systems analysis and design and web information system methodologies. These lines of research will first be described followed by a discussion on how they can come together to create a more structured methodology for application development for internet based applications.

### Systems Analysis and Design

During the late 1970's and early 1980's, a systematic approach to developing applications was developed called the Systems Development Life Cycle (SDLC) (Avison and Fitzgerald, 2003). The SDLC is a phased approach consisting multiple phases that are designed to improve the management of information systems development. Other methodologies that have been established for application development include Joint Application Development (JAD), Rapid Application Development (RAD), and Prototyping. Prototyping is an adjustment to the SDLC used to speed up development and delivery of applications (Kendall and Kendall, 2010; Turban et al, 2004).

The SDLC will be used as a benchmark for application development due to it being a complete methodology containing all of the steps required for development of applications. Due to the nature of web application development environment, time schedules for development are relatively short, often being less than one or two months for development (Pressman, 2000). These web applications require a high degree of granularity that is often not completely suited for the SDLC. Prototyping generally is more suited to these types of problems due to its ability to speed up the steps of the SDLC. Therefore, this research will focus its attention on the SDLC and prototyping methodologies. The majority of the literature covering systems analysis and design focus on the analysis of information or knowledge used in designing a system or application. These methodologies are formal attempts used to address the complexity of system development through standard, predictable approaches (Vessey and Glass, 1998). Developing web applications is a highly complex problem consisting of many dimensions that must be addressed for web application development to be successful (McDonald and Welland, 2001):

Short development life-cycle times.

1. Delivery of customized solutions.
2. Multidisciplinary development teams.
3. Small development teams working in parallel on similar tasks.
4. Analysis and Evaluation.
5. Requirements and Testing.
6. Maintenance.

The SDLC covers most of the needs for successful web development minus one critical

factor, the need for short development time. The SDLC has added structure to complex problems that exists in application development. The basic concept behind the SDLC method is that there is a well-defined process by which an application is conceived, developed, and implemented. (Kettinger and Teng, 1997). The speed at which applications need to be developed and the additional factors such as navigation and the graphical component adds complexity that is beyond the use of SDLC by itself.

Prototyping is generally used for more unstructured processes and for rapid development of applications. There are several methods of prototyping that can be used for application development. The method focus on by the current research is known as the "Throw Away" prototype, which is used primarily to clarify and satisfy information requirements and then discarded. (Kettinger and Teng, 1997). One technique used in the throw away method of prototyping is to create a visual display or image of what the web pages will look like using a graphical tool such as Adobe Photoshop or Macromedia Fireworks. This will allow the developers to quickly create a visual prototype for the users to see what the website will look like and make any necessary changes. This allows the users to work in iterations decreasing the time of development.

### Web Application Development

Internet applications encompass four general types of web-based systems: Intranets, web-presence sites, electronic commerce, and extranets (Isakowitz et al, 1998). These systems all have different functions and goals but are similar in regards to the technology used, namely the protocol known as Transmission Control Protocol (TCP). TCP is the most commonly used transfer protocol of the Internet (Altman et al, 2005). The Information used by these web applications is stored in many places and rapidly changes containing both structured and unstructured formats (Fraternali and Paolini, 2000). Other differences between web applications and software applications is the content, including text, graphics, images, audio, and video, which are integrated with procedural processing (Pressman, 2000).

The speed at which information is desired has also increased resulting in the need for quicker delivery times for web based applications. Numerous methodologies have been developed and used in order to respond to the changes brought about in the web development environment. McDonald and Welland (2001) describe several differences between web-based application development and traditional software development, such as:

1. Short Development Life Cycle times (Typically 3 months or less)
2. Custom made systems integrating software and data
3. Multidisciplinary development teams

Research has shown that adopting a minimalistic approach to design results in increased effectiveness (Rosen and Puriton, 2004). Strategic use of graphics can draw users to the site and sustain their interest. However, excessive use of graphics may overwhelm the users causing them to exit the website. This calls for careful planning in the layout and graphical display of the website during the requirements and planning phases of development.

Many development methodologies have been proposed to address the complexity of web application development. Each methodology has its own series of phases that may or not be the same as other methodologies. Escalona and Koch (2002) developed a taxonomy describing different methodologies and the phases that each methodology contains compared against the system development life cycle. Most methodologies used for web development focus on designing and coding the application. Hypermedia Flexible Processing Model (HFPM) strategy is the only methodology that includes all of the phases of the SDLC. However, HFPM does not state the techniques to be applied in each of the phases, it only gives the process. No single methodology listed offers a complete suitable framework for development of web applications (Escalona and Koch, 2002).

## **WEB DEVELOPMENT LIFE CYCLE**

The purpose of this paper is to propose a new methodology encompassing all the phases of the SDLC in a suitable way that can be used to develop web based applications quickly in a structured manner. The Web Development Life Cycle (WDLC) is a new methodology being proposed specifically for the development of web applications. This methodology is grounded in previous methodologies found in the literature to create structured process for the highly unstructured problem of web application development. The WDLC is a hybrid of two former methodologies known as the Systems Development Life Cycle and Prototyping. The WDLC uses components from each methodology, combining them into a new approach that will decrease the time of development, add structure to an unstructured problem, and keep the users involved throughout the entire development life cycle.

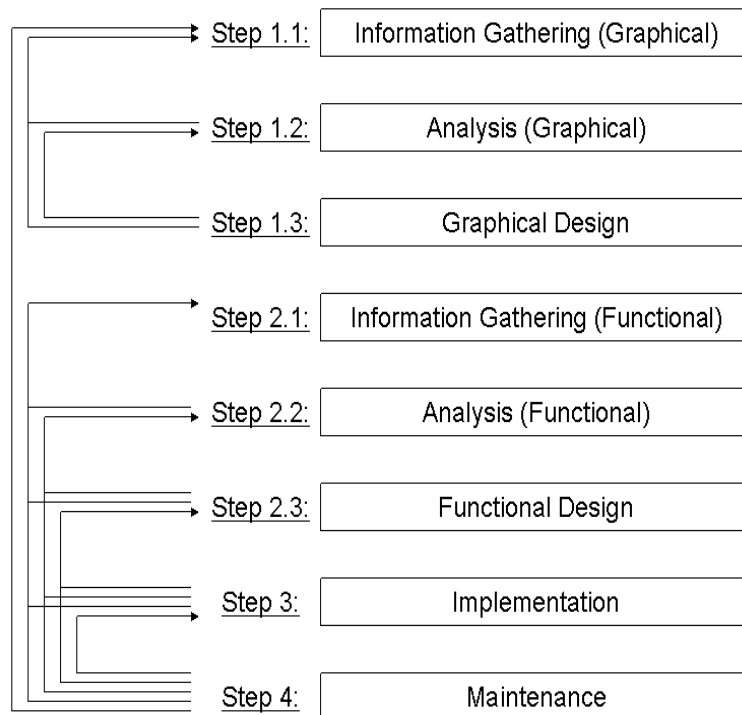
Rosen and Purinton (2004) describe three factors are important to users when using web based applications: coherence, complexity, and legibility. Coherence involves the ease at which the user can navigate through the site and includes the understandability of the content contained within the website. Complexity revolves around the design of the website including the variety of images that satisfies the users desired to explore the environment. Legibility is rooted in the impression made by the website to the user. It is the perception that the website is memorable and distinctive. Internet application design addresses both the database/application development and the graphical representation to the users (Gellersen, 1999). The visual element of a website encompasses the website's aesthetics whereas the database element affects functionality within a website. Both the aesthetic design and functionality of the website are important but each are distinct from the other. The graphical design of a web application is often performed by a graphic designer or artist whereas the database design would be performed by a database administrator or programmer. Therefore, the WDLC will contain two iterative steps, which include the graphical development and the functional development.

Scalability, modifiability, and robustness are important factors that must be addressed in any system (Ross et al, 2008). Due to the dynamic nature of Internet applications and their need to be viewable 24 hours a day, 7 days a week, these factors have become even more prevalent. The ability to rapidly change content, requirements, and the architecture of Internet applications is important when designing for scalability. (Fraternali and Paolini, 2000). Lack of scalability is a problem that has been seen in many large scale IS projects in developing countries (Avgerou, 2008). One phase that

few Internet application development methodologies fail to address is the requirements phase (Escalona and Koch, 2002). Spending a little more time on the front end on information gathering and analysis will allow for faster implementation of applications on the web site in the future. Systems that are well designed and thoroughly implemented often display improved software quality and increased reusability (Pree, 1995). Web-based applications tend to evolve very quickly with frequent updates and redesigns. These factors often lead to decreased maintainability and can be very troublesome throughout the life of the application (Gellersen, 1999). The WDLC addresses this issue by dividing the development of Internet applications into two distinct iterations of the development process:

1. Graphical Development
  - a. Information Gathering
  - b. Analysis
  - c. Design
2. Functional Development
  - a. Information Gathering
  - b. Analysis
  - c. Design

This follows the same concept of the SDLC model using a waterfall approach. However, by incorporating prototyping methodology, the WDLC become iterative process involving the users through the development process. By dividing the graphical development and the functional development into different stages, each of these set of phases can be worked on sequentially or simultaneously. For instance, if the website has already been designed and developed and a new application needs to be added, the designers and developers could skip to the functional development set of phases in the WDLC and add the application to the existing website. By following proper measures in the graphical development and planning for scalability, future updates or redesigns will require minimal efforts. Redesigning an existing website would only require the graphical development phases of the WDLC to be performed since the functionality of the website would remain unchanged. This creates congruence among the different parts of a web site and its applications. Since the design phase of the WDLC uses prototyping, after it has been tested and works you can implement it as the working system. This creates a four phase approach to web application development with two of these phases containing 3 steps each. The complete WDLC can be seen in Figure 1:

**Figure 1: Web Development Life Cycle**

These four phases (8 steps) are borrowed from both the SDLC and prototyping and will now be explained in more detail.

#### Information Gathering (Graphical)

The first phase of the WDLC is the information gathering step for designing the website. The design of a website is extremely important because if it is not appealing to the customer and easy to navigate, then they will be less likely to return and purchase the products or use the services offered. The analyst is required to gather information that will assist the graphic designers in creating an effective layout for the website and determine the various pages will be included. How information is to be arranged and the navigation through this information should also be discussed in the information gathering phase of the WDLC.

#### Analysis (Graphical)

The next step involves analyzing the information acquired and documenting the needs of the website design. This documentation includes the color schemes to be used along with logos and other graphics that will be incorporated on the website. The analyst will also be required to outline the site map entailing how the users will navigate through the website. Incorporating a navigation elements, search mechanisms, and site map within the website will increase the user's ability to find the information they desire (Malak et al, 2010). This is also important for scalability because it will allow the programmers to create templates and implement cascading style sheets (CSS) into the site allowing for easy maintenance and updates in the future.

The graphic designers use the acquired information to create an image of what the website should look like for the developers to use. This will be a graphical representation, or prototype, of the website that the programmers will use to develop the template. The graphic designers typically work closely with users (marketing, merchandising, etc.) in charge of leading the efforts for development of the website. Images of the website pages will also be included in the documentation.

### Graphical Design

The graphical design phase of the WDLC is where the developers use the documentation provided by the analysts and graphic designers and create a prototype of the website. The only functionality available at this phase of the WDLC is the ability to navigate through the website. This is where the programmers create the templates and navigation for the website. Once the prototype has been finished and tested, the working model will then be used as the actual system.

### Information Gathering (Functional)

Phase 2 involves the functional development of the website. During the information gathering phase for the functionality development, the systems analyst meets with the users to gathering requirements for the functionality of the website. They identify the purpose of the website, what functionality is required, and the different components for each section of the website.

### Analysis (Functional)

The functional analysis phase of the WDLC is where the analyst creates Entity Relationship Diagrams (ERD) and Data Flow Diagrams (DFD) needed for the functionality of the website. They will then break down the different components of the website into smaller fragments. For example, if developing an e-commerce website, the developer might create different components such as a shopping cart, products page, contact page, information page, and a frequently asked questions page. Then each one of these components will be designed and developed as if they were individual programs. Once the website's functionality has been designed and the different components have been developed, the developer will start creating the physical prototype.

### Functional Design

The functional design phase of the WDLC is where the developers start creating the prototype for each of the components on the website. This is where the functionality of the website is developed. The web developers work with the users to identify the components of the website that are required for the implementation. After identifying the critical components, they begin developing the prototype for those components. The users should be actively involved with the developers as each component is created. Many times the user knows what they want but do not understand what can be done in web development. Therefore, the web developer should guide the users and involve them during the development process.



As functionality is added to the website, the users should test the components and give feedback to the developer. After all the necessary changes have been made to the component, the web developer will start on the next component and repeat this process. Each component can be implemented upon completion and successful testing. This creates an iterative process of the functional phases in the WDLC and increases the speed at which the website is developed by working in modules.

### Implementation

This phase in the WDLC is similar to normal implementation in the SDLC. The prototype is typically developed on a test server or development server. This allows the users to work with the developers until the prototype is complete. Once the component being developed is completed, the database and web page files will be moved to the production server for implementation.

### Maintenance

Maintenance is an ongoing step of the WDLC that never ends. Maintenance may include modifications to existing programs, updating the style sheet to give the website a different look, or anything that might need to be done to the website once it has been implemented.

The WDLC is an iterative process, particularly during functional phases of development. Once the website has been designed there is little need to perform the graphical phases of the WDLC until a re-design of the website is needed. For each new application or component added to the website after it has been designed, the systems analyst can start on phase two of the WDLC. For a website redesign, only phase one will need to be completed because a redesign will change the layout and look of a website but not the functionality.

## **CONCLUSION**

The WDLC is a theoretical model based on the author's experience in web development culminated with a theoretical foundation in the literature using existing methodologies. The short history and lack of structure within web application development has led to many websites being unsuccessful over long periods of time. Adding structure to the development process will hopefully provide organizations with a tool that can be used to create a web presence, whether it is an information website, electronic commerce, or an intranet system used for reporting capabilities. Regardless of the reason for development of a website, success is the ultimate goal and the WDLC is designed to give developers a structured methodology to help ensure that success.

## **LIMITATIONS**

One limitation with the current research is lack of empirical evidence to prove the success of the WDLC. The WDLC is a theoretical model based on the literature with the Information Systems community guided by the author's experience. Future research should evaluate the success of this model within a practical environment. The difficulty in testing this sort of model is obtaining a large enough sample size to conduct statistical

analysis to show significance. Many companies have small development department with few web developers in order to collect data from. To overcome these limitations, future research might employ these methods within a web development group and perform a case study along with qualitative analysis to gain the perspective of business professionals who have agreed to adopt the methodology.

## REFERENCES

- Altman, E., Barakat, C., & Avrachenkov, K. (2000) A Stochastic Model of TCP/IP With Stationary Random Losses. In Proceedings of ACM SIGCOMM.
- Avgerou, C. (2008) Information Systems in Developing Countries: A Critical Research Review. *Journal of Information Technology*, 23 (3), 133-146.
- Avison, D. & Fitzgerald, G. (2003) Where now for development methodologies? *Communications of the ACM*. 46 (1).
- Escalona, M.J., Mejías, M., & Torres, J. (2002) Methodologies to develop Web Information Systems and Comparative Analysis. *Informatik/Informatique*. núm. de I/I, 3 (2).
- Escalona, M.J., & Koch, N. (2004) Requirements Engineering for Web Applications: A Comparative Study. *Journal of Web Engineering*, 2 (3), 193-212.
- Fraternali, P., & Paolini, P. (2000) Model-Driven Development of Web Applications: The AutoWeb System. *ACM Transactions on Information Systems*, 18 (4), 323–382.
- Gellersen, H., & Gaedke, M. (1999) Object Oriented Web Application Development. *IEEE Internet Computing*, 3 (1), 60-68.
- Graham, M. (2008) Warped Geographies of Development: The Internet and Theories of Economic Development. *Geography Compass*, 2 (3), 771-789.
- Isakowitz, T., Bieber, M., & Vitali, F. (1998) Web Information Systems. *Communications of the ACM*, 41 (7), 78-80.
- Kendall, K. E., & Kendall, J. E., (2010) *Systems Analysis and Design* (8th ed.). Upper Saddle River, New Jersey: Prentice Hall.
- Kettinger, W., & Teng, J. (1997) Business Process Change: A Study of Methodologies, Techniques, and Tools. *MIS Quarterly*, 21 (1), 55-98.
- Malak, G., Sahraoui, H., Badri, L., & Badri, M. (2010) Modeling Web Quality Using a Probabilistic Approach: An Empirical Validation. *ACM Transactions on the Web*, 4 (3).
- McDonald A. & Welland R. (2001) *Web Engineering in Practice*. Proceedings of the Tenth International World Wide Web Conference (WWW10).
- Molla, A., & Heeks, R. (2007) Exploring E-Commerce Benefits for Businesses in a Developing Country. *The Information Society*, 23 (2), 95-108.
- Myers, B., Hollan, J., & Cruz, I. (1996) Strategic Directions in Human-Computer Interaction. *ACM Computing Surveys*, 28 (4), 794–809.
- Pree, W. (1995) *Design Patterns for Object-Oriented Software Development*. Addison-Wesley. Reading, Mass.
- Pressman, R. S. (2000) What a Tangled Web We Weave. *IEEE Software*, 17 (1), 18-21.
- Rosen, D., & Purinton, E. (2004) Website Design: Viewing the Web as a Cognitive Landscape. *Journal of Business Research*, 57 (7), 787-794.
- Ross, A., Rhodes, D., & Hastings, D. (2008) Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value. *Systems Engineering*, 11 (3).

Turban, E., Aronson, J., & Liang, T. (2004) *Decision Support Systems and Intelligent Systems*. Prentice Hall, Seventh Edition.

Vessey, I., & Glass, R. (1998) Strong vs. Weak Approaches to Systems Development. *Communications of the ACM*, 41 (4).