INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 01, January 2023, pp : 156-161

e-ISSN : 2583-1062

Impact Factor : 2.625

DEVELOPMENT OF WEB APPLICATION USING MODEL VIEW CONTROLLER (MVC) ARCHITECTURE

Patel Anwar Ismail[*1]

[*1]Research Scholar MSc IT Part 2, Priti Academy Degree College, Shahad. Thane, Maharashtra, India.

DOI: https://www.doi.org/10.58257/IJPREMS30585

ABSTRACT

The web applications have evolved into more advanced systems and their complexity has grown in industry significantly. The MVC implementations are heavily dependent on a specific set of technologies.

I propose an implementation of MVC based web application for stores or outlets in more abstract form, in which – I think – we will increase the realization of the advanced web applications form, thus lowering the engineering complexities and difficulties of web applications.

I believe that this implementation is more applicable in a wider range of environments and technologies, and will uphold the architectural properties like performance and modifiability. Based on this implementation we introduce an MVC based architectural style for web applications.

This web application will have a partial impact on industrial production and marketing advance ability in society for users as well as for industry.

In future, we expect to improve this further towards supporting Rich Internet Applications involving data analysis and developing machine learning models for growth of industry and to match user criteria.

1. INTRODUCTIONS

A Web application development is the creation of application programs that reside on remote servers and are delivered to the user's device over the Internet. A web application does not need to be downloaded and is instead accessed through a network. An end user can access a web application through a web browser Microsoft Edge, Mozilla Firefox and many more.

To improve the user experience and help developers build faster and more powerful web applications. The web environment today uses HTML and CSS to present data to users. These technologies are called "front-end" or "client-side" technologies. On the other hand, "back-end" or server-side" technologies refer to data storage and processing technologies.

In conducting the research, I have used our experience in building web applications and I have tried to identify both strengths and weaknesses of these systems while providing our own view on how these practices could be improved. Studied frameworks and systems include: MS SQL, ASP.NET MVC, HTML and CSS.

2. SYSTEM DETAILS AND APPLICATION STRUCTURE

Hardware Requirement

This application requires minimum system requirements for using ASP.NET MVC platform are as follows:

Operating System

Windows 10

Windows 8, 8.1

Windows 7

Windows vista

Windows server 2008 and later

Android 4.1 & later

iOS 9 & later

Hardware Environment

Processor: x86 or x64

RAM: 512 MB (minimum), 1 GB (recommended)

Hard disc: up to 3 GB of free space may be required

Development Environment

Microsoft Visual Studio 2010/2012/2013/2015/2017/2019/2022

.NET Framework 3.5/4.0/4.5/4.5.1/4.6

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 2.625

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 01, January 2023, pp : 156-161

MVC Version: MVC3/MVC4/MVC5

Microsoft SQL Server 2005 or higher

Web browsers with JavaScript enabled

Planning and Process/Specifics Requirements for Web Application Development Process

- Roadmap: Defining web application, Purpose, Goals and Direction.
- Defining Application Scope
- Performance Analysis
- Technology selection, technical specification, Web Application structure, Timelines

ASP DOT NET MVC

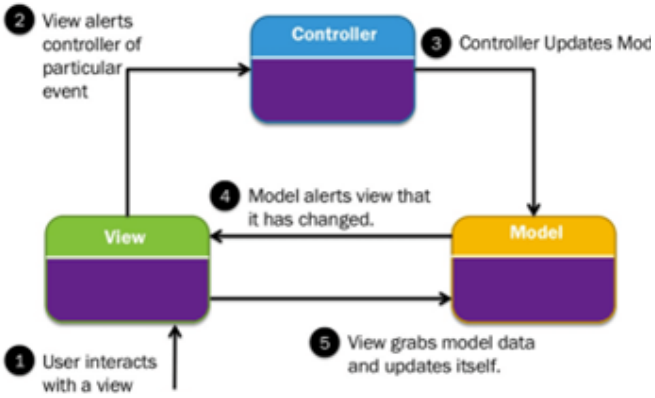

Fig 1: Process flow of ASP DOT NET MVC module

Model-View-Controller (MVC) framework is an architecture that separates an application into three main logical components Model, View, and Controller. Each architectural component is built to handle specific development aspects of an application.

The three parts of the MVC architecture pattern can be described as follows:

- Model: Manages data and business logic.
- View: Handles layout and display.
- Controller: Routes commands to the model and view parts.

ASP DOT NET MVC Features

- MVC structure enhances the test-driven development and testability of the application
- It utilizes the component-based design of the application by logically dividing it into Model, View, and Controller components.
- It supports all the existing vast ASP.NET functionalities

Web Application Development

(Executed by Comentum's Development Team)

The application's Design Interface is turned over to Comentum's Development Team who take the following steps to develop the project:

- Create the Web Application Architecture and Framework
- Design the Database Structure
- Develop / Customize the Web Application Module, Libraries and Classes
- Complete the Development and Implement all Functionalities - Version 1.0

Specifics Requirements for Web Project Development Process

There are many studies pertaining to the actual Web applications development processes. Some of these studies show some general and basic requirements on a development process, while others revealed the underlying problems in the Web applications development that cannot be addressed by current process models. In the next subsections, the study will discuss the most important requirements of the Web applications development process. These requirements are the main reason why the traditional software development process model must be customized to suit the needs of the Web application development, or else they may not acceptable.

Handling Short Development Cycles

The fact found in several empirical studies is that the development time of Web applications is very short, usually not more than six months, and the average is less than three months. As the short development cycle is so common for Web applications, it is presumed to be the first requirement in the Web application development process.

Handling Changing Requirements

The Web project requirements are always subject to change. Throughout and even after the development of the application, the technical and organizational constraints are constantly modified. This could be due to the uncertain requirements at the beginning of the development or modification of requirements after the system is completed. For this reason, the Web applications are frequently referred as "moving targets". As a direct result of this inconsistent requirement, there is a need for a strong integration between the customer and the Web project development team.

Parallel Development Process

The strong competition has pressured the competitors to shorten the development cycle. With this sort of time constraint, only parallel development projects can meet the requirements. This means, all the methodological activities from design, implementation to quality assurance phases must be carried out concurrently. In most cases, a number of small development team working on similar duties in parallel, and this requires planning on staff deployment. As such, high communication level is required in the Web project development.

Reuse and Integration

The enormous time pressure in Web application development has driven developers to reuse as many Components as possible. This often involves the interoperability and integration of diverse components which were either developed internally or purchased from third parties. Therefore, usually the development process of one Web application is not done in isolation from other Web development applications within the organization itself. Commonly, a reusable component is developed for a project in coordination with other projects that will use this component. Furthermore, there are always advantages in developing a common architecture for more than one Web applications.

In line with the growing integration of Web applications with the customers' business processes, the need to integrate Web applications with the existing applications, or other Web applications under development has also increased. As a result, the development process has to be coordinated with the desired results and the approaches used to achieve them.

Web project Development Process

The wide choice of different process models reflects the large range of different software projects. The well-known software development processes can be grouped into two categories:

Lightweight Processes: It is better known as agile processes. They are suitable for smaller projects with smaller development teams.

Heavyweight Processes: Heavyweight processes are particularly used for large teams with high demands on the quality.

The terms "light" and "heavy" refer to the degree of process formalization, for instance the number of documents and models created in the project.

Web projects are significantly different from the traditional software projects in several aspects which influence the suitability of the conventional software development process models. In the next subsections, the study will check the suitability of the software development process models to meet the specific requirements for the Web project development process. We will look at the Rational Unified Process (RUP) as a representative of the heavyweight, phase-oriented, and iterative process models and Extreme Programming (XP) as an example of lightweight, agile process model.

Rational Unified Process (RUP)

This section describes RUP as a representative of the heavyweight, incremental, iterative, and phase oriented processes. The key concept of RUP is to describe all the activities throughout the development life cycle, including requirements elicitation, analysis, design, implementation, and testing phase. Since the RUP activities can be overlapped and carried out in parallel, it is found to be different from the classical waterfall process.

There are four separate phases within each of the activities in a development project, and each of these phases is organized in a number of separate iterations. Fig.2 shows the mentioned phases and they are inception, elaboration, construction, and transition. While they occur respectively, there may iterate until the project is complete.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 2.625

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 01, January 2023, pp : 156-161

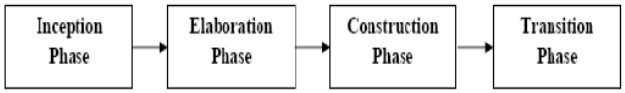

Fig.2: The four phases of the Rational Unified Process

RUP can be considered as an iterative process and the iterative characteristic within a phase is subject to the objectives of that phase. For example, in the early iteration it can develop a prototype of the project under development. The prototype is used to identify the requirements of customers and users. On the other hand, several iterations in the elaboration phase are scheduled to usually implement fractions of the architecture; however, without a complete job.

To investigate whether the RUP can achieve the desired goal in Web application development projects, the study have to assess the four phases based on Kruchten's findings:

- **Inception phase:** During the inception phase, the developers define the scope of the project and business case of the system. The goal of this phase is to develop a common vision of the final product by cooperating with the customer and future users. The definition of the first phase is tricky for Web project development since the requirements of the target groups are unknown at the beginning of the project. Moreover, these requirements change continuously, causing the vision of the Web application to evolve continuously, even when the Web application may already be in use. Research to fix the vision beforehand takes a long time and it is costly. There is also a hazard of the vision becoming outdated by the time the product is completed.
- **Elaboration phase:** In this phase, the developers analyze the requirement of the project in detail and define its architectural foundation. The goal of this phase is to exclude the highest project hazards to the widest possible extent to formulate a fixed price by the end of this phase. This includes the selection of an optimal and expandable architecture as well as the familiarization of the staff with the technologies to be used.
- Since the first product version has to be built in an extremely short time frame during the development cycle, the main concern is to calculate a fixed price to a well-defined product. Since the customers cannot be expected to be loyal to a single Website, with its competitors being just a mouse click away, it is difficult to predict the economic success of a Web application than that of a traditional software product.
- **Construction phase:** In this phase, the developers emphasize on completing the analysis, performing the majority of the design and implementing the system. This means that the product is built by implementing all the components and integrating them into one product. The main challenge here is the question of whether there could be one point of time where all the components are completed since they are being handled in parallel by a team of people with different capabilities.
- **Transition phase:** The developers, in this phase, deliver the system to the users by incorporating the product with the user environment. This phase can be very straightforward if it is possible to simply replace an existing application with a new one. However, unlike the traditional software, the distribution to users happens through the Web's architecture. Besides, there is usually no user training required.

Extreme Programming (XP)

XP represents the agile iterative processes. Alternatively, the study could use other agile processes such as Scrum, Adaptive Software Development (ASD), Feature Driven Development, or Crystal Clear in our research. XP was selected merely because it is very common.

Agile processes are created on the iterative development basis. They use feedback rather than planning as their major control procedure. The feedback is driven by normal tests and advanced versions of the software. XP projects have four core values:

- **Communication:** The goal is to provide all developers with a common vision of the system that matches the clients.
- **Simplicity:** Encourages beginning with the easiest solution and building it again to improve it further. XP project concentrates on coding and designing that are required today instead of those of tomorrow.
- **Feedback:** Guides the project throughout the way. Within XP, feedback is required and used at various phases of the development. The main objective of this core value is to prevent problems at the earliest possible point during the development process.
- **Courage:** means that developers, managers, and customers should be brave enough to try new approaches and ideas.

Pair programming is the best characteristics of XP model and it is based on the belief that two individuals could see more than a single person, and being side by-side, they can expand their ideas better than an individual who does it alone. The quality increase achieved by this approach substitutes the extra cost.

Moreover, this approach ensures that the communication principle is anchored in the process and its knowledge is distributed throughout the team. To investigate whether an XP model can achieve its desired objective in Web application development projects, the study will discuss how XP model can meet the defined requirements for Web project development process.

- **Handling short development cycles:** XP and other agile process models meet this requirement completely since the highly successive releases are one of the characteristics of XP projects. Iterations also allow to structure the short development cycles.
- **Handling changing requirements:** The simplicity is one of the core values of XP, which means that any requirement for tomorrow's need is rejected. Instead, the close integration with the customers, combined with a rapid delivery of results, allows development and continuous requirement adaption.
- **Parallel development of different releases:** XP does not fundamentally exclude the parallel development of different releases, because no XP project uses a cautious way of working.
- **Reuse and integration:** The integration of the existing components requires a methodological support rather than support by the process itself. However, it should be reminded that this approach might be a challenge to achieve, because XP processes are mostly selected to solve a specific single problem only. In such scenario, it might be better to select processes designed especially for the development of reusable software than XP in particular or agile models in general.

3. CONCLUSIONS

In short, he study has illustrated that there is no single process studied in this paper, which is able to meet all the requirements of the Web applications development process. Since the requirements of Web application are different from those in traditional software, the software development process models should adopt and adapt the best method to meet the needs of the Web application development. Therefore, it is extremely important that Web application developers understand the special Web application requirements that may influence the Web applications development process.

4. REFERENCES

[1] Lee, S. C., and Shirani, A. I. A Component Based Methodology for Web Application Development. Journal of Systems and Software, 71(1−2), 2004, pp.177−187.

[2] Murugesan, S., and Deshpande, Y. Meeting the Challenges of Web Application Development: The Web Engineering Approach. Proceedings of the 24th International Conference on Software Engineering. USA: ACM. 2002, pp.687−688.

[3] Ginige, A. Web Engineering: Managing the Complexity of Web Systems Development. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering. USA: ACM. 2002, pp. 721−729.

[4] Mayr, H. Web Project Management. In: G.Kappel, B. Proll, S. Reich, and W. Retschitzegger (eds.). Web Engineering: The Discipline of Systematic Development of Web Applications. USA: John Wiley & Sons, 2006, pp. 171–195.

[5] Shklar, L., and Rosen, R. Web Application Architecture: Principles, Protocols and Practices. USA: John Wiley & Sons, 2003.

[6] McDonald, A., Welland, R. Web Engineering in Practice. Proceedings of the 4th Workshop on Web Engineering. Hong Kong, 2001, pp.21-31.

[7] McDonald, A., Welland, R. A Survey of Web Engineering in Practice. Scotland: University of Glasgow. Technical Report, R-2001-79.

[8] Lowe, D. Characterisation of Web Projects. Proceedings of the 8th Australian World Wide Web Conference (AusWeb 2002). Sunshine Coast. Australia, 2002,pp. 58-68.

[9] Brambilla, M., Ceri, S., Fraternali, P., and Manolescu, I. Process Modeling in Web Applications. Journal of ACM Transactions on Software Engineering and Methodology (TOSEM). 15(4), 2006, pp.360-409.

[10] Ramesh, B., Pries-Heje, J., and Baskerville, R. Internet Software Engineering: A Different Class of Processes. Journal of Annals of Software Engineering. 14(1–4), 2002, pp. 169–195.

[11] Rossi, G., Pastor, O., Schwabe, D., and Olsina, L. (2008). Web Engineering: Modelling and Implementing Web Applications. UK: Springer-Verlag, 2008.

[12] Kruchten, P. The Rational Unified Process: An Introduction. USA: Addison Wesley, 2003.

[13] Hunt, J. The Unified Process for Practitioners: Object Oriented Design, UML and Java. USA: Springer-Verlag, 2000.

[14] Salo, O., and Abrahamsson, P. An Iterative Improvement Process for Agile Development. Journal of Software Process. 12(1), 2007, pp. 81-100.

[15] Kessler, R., and Dykman, N. Integrating Traditional and Agile Processes in the Classroom. Proceedings of the 38th Technical Symposium on Computer Science Education (SIGCSE 2007). USA: ACM, 2007, pp. 312-316.

[16] Trujillo, S., Batory, D.,and Diaz, O. Feature Oriented Model Driven Development: A Case Study for Portlets. Proceedings of the 29th international conference on Software Engineering. USA: ACM, 2007, pp. 44-53.

[17] Mattsson, M. K. Problems in Agile Trenches. Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. Germany: ACM. 2008, pp. 111-119.