Open access Journal International Journal of Emerging Trends in Science and Technology
Impact Factor: 2.838 DOI: http://dx.doi.org/10.18535/ijetst/v3i03.07

Use of Continuous Integration in Single Page Rendering Web Application Development

Authors
T.Nathiya[1], Dr S. Prasanna[2]
[1]M.Phil (Research Scholar), School of Computer Science, Vels University, Chennai
[2]Assoc. Prof., Department of MCA, School of Computer Science, Vels University, Chennai
Email: *tnathiya17@gmail.com[1], Prasanna.scs@velsuniv.ac.in[2]*

Abstract
This paper is mainly focused on single page rendering web application development using continuous integration. The Continuous Integration (CI) is a software development life cycle process which mainly focuses on automation and continuous delivering backlogs. Applying CI into the web application development helps the developers have an effective application development and fulfil customer needs. Move-View-Control (MVC) design pattern is used for code separation which helps to developer easily understand code logic as well as debugging application. The development of the lifecycle is based on the continuous integration and agile technique is analysed in reflecting the quality improvement. The evolution of the continuous integration is used in single page rendering.

Keywords: *Introduction, Agile Software Development, Continuous Integration, MVC (Move-View-Control), Web Application, Testing, Integration Process.*

INTRODUCTION

There are various software development models defined and designed which used the employee during the development process. The Software Development Life Cycle (SDLC) models have been created, such as waterfall, spiral, Agile Software Development, Incremental.

Each process model following the main approach to software development that begins with customer business requirements and progresses through planning, modelling, construction, and deployment. Once software development is completed, post implementation support is required which is fulfil customer needs for change requirement (CR).

There are following six phases in every software development life cycle model:

1. Requirement gathering and analysis
2. Design
3. Implementation or coding
4. Testing
5. Deployment
6. Maintenance

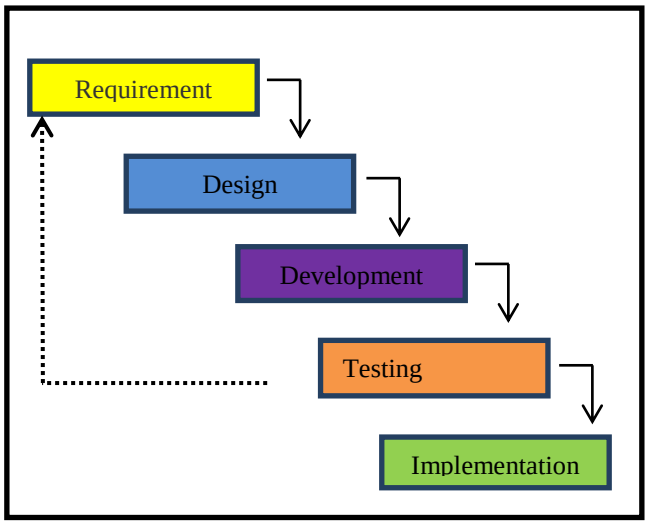

Figure 1: Waterfall Model

AGILE SOFTWARE DEVELOPMENT

Agile SDLC model is an iteration and incremental business requirement model. An agile team able to appropriately respond to changes and delivery the

backlogs. All team member works on their task and complete the requirement. Daily stand-up meeting is useful for team member day to day work and better understanding the customer requirement.

Agile model focus on every project handle differently and the existing needs to best suitable for the project requirements. Agile model is mainly divided into small time frames to deliver complete feature release.

Iteration approach is delivering software features one by one. Each built is incremental of feature, the final built holds all the features which are actual customer needs.

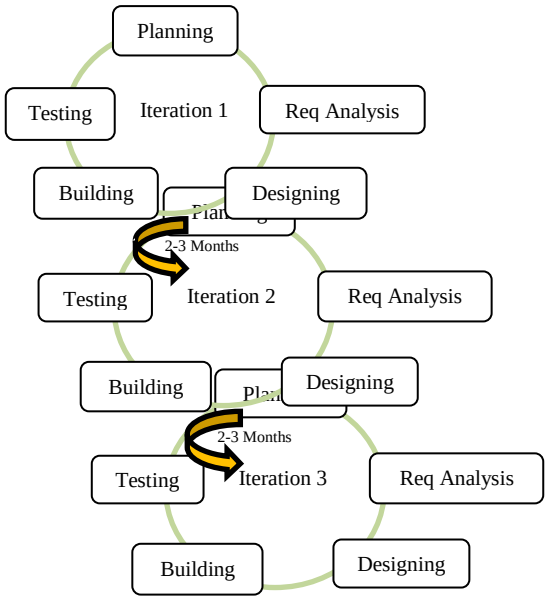

Figure 2: Incremental Model

The agile methods include Rational Unified Process(RUP), Scrum, Crystal Clear(CC), Extreme Programming(EP), Adaptive Software Development(ASD), Feature Driven Development(FDD), and Dynamic Systems Development(DSD) Methods.

The Agile defines following principles

Individuals and interactions- In agile development, self –organization and motivation are important, as are interaction like co-location and pair programming.

Working software- The Working software is considered the best means communication with customer to their requirements depending on documentation.

Customer collaboration- As the requirements cannot be gathered completely in the beginning of the project, customer interaction is very important to get the product.

Responding to change- Agile development is focused on quick response to change and development.

Scrum

Scrum the name derived from an activity. Scrum is an agile process model that was developed by Jeff(1990) .

Agile software methodology is managing product development and satisfies customer needs. It defines product development where a development team works as a unit to reach the output. challenges are traditional, sequential approach product development, and team to self-organize their meeting. Daily meeting is organizing by team member and assemble in physical co-location or online meeting with team members, as-well-as daily face-to-face communication all the team members and disciplines to the project.

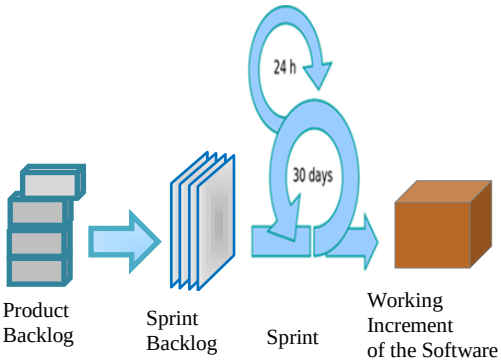

Figure 3: Scrum Model

There are three roles in scrum management.

1. Scrum Master
2. Project Owner
3. Team

The project owner is responsible for transforming functional and non-functional requirements. After

the requirements are delivered, these requirements together provide product backlog. Project owner to arrange the list to the priority, creating the starting point for development. The priority is changing the development process. Reflecting changes in the requirements. Scrum development activities within process the following framework activity, work task occur within a process pattern called a sprit. Each sprit is take 30 days, starting within a planning meeting.

The project owner and a Team meet to place together a Sprint backlog .During The First part of the meeting, the owner presents the highest priority tasks from a product backlog to the team. In Second part, the scrum is planned and developers have 30 days to meet the required plan. During this 15-minute-session, each member provides information what they done since the previous daily scrum and they are planned to do until the next daily scrum. During this daily meeting, the work of the whole team is synchronized. In the end of every sprint, there is a review meeting. As a result of the scrum are presented to the product owner and then to other stakeholder

Continuous Integration

Overview

We already mentioned two values of agile development are responding to change and working software. The continuous integration as a software development practice where members of a team integrate their work frequently, usually each person integrations per day.Automated build tools are very useful for delivering release quickly as possible and easily to rectifying errors. Many teams find that this approach leads to significantly reduced integration problem and allows a develop software.

History

The Continuous Integration, there were several times to development processes.

Nightly built- A team process that compiles and execute code base every night.

Continuous built- A continuous build appears. After one built had finished, another one take place based on the newest version in the version control system.

Key practices

Continuous Integration defined some basic practices:

- Maintain a single source repository
- Automate the build
- Make your build self-testing
- Every one commits to the every day
- Every one commit should build the mainline on the integration machine
- Keep the build fast
- Test in a clone of the production environment
- Make it easy for everyone to get the latest executable
- Automate development
- Fix broken build immediately
- All tests and inspections must pass
- Run private builds
- Avoid getting broken code

Key Values of Continuous Integration

Summarize the value of continuous integration as following:

- Low delivery ricks
- Reduce repetitive manual processes
- Deployable software
- Batter project visibility
- More product confidence

Design Pattern (MVC)

Overview

Model-View-Controller (MVC) is a Software design pattern for developing software and better organizing code separation. MVC has three interconnected parts, so as to separate internal representation of information from the ways that information is presented to or accepted from the user.

MVC architecture has become extremely popular for designing web application and easy to test modules.

History

MVC was one of the early development of the GUI and one of the first approach and implement software construction The MVC pattern has slightly different from HMVC, MVA, MVP, MVVM, and other that adapted MVC to different contexts.

Components

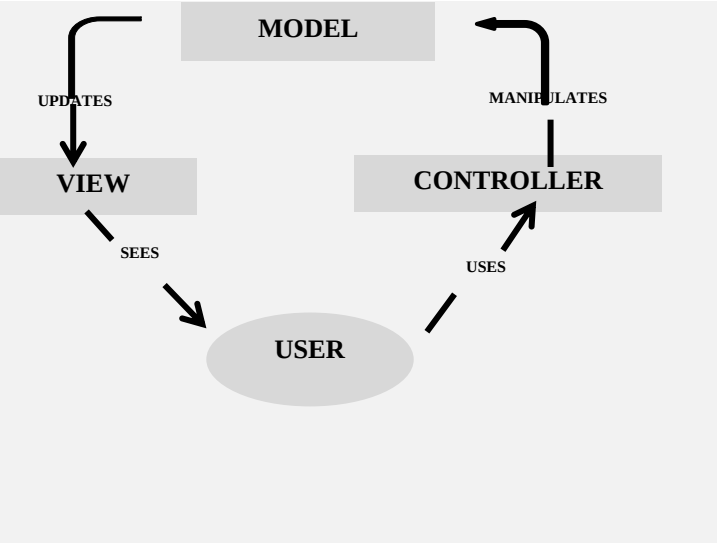

Figure 4: MVC Model

The Model – It is manages the data, data format and rules of the application:

The View – It is output representation of data to user, such as a chart or a diagram, multiple views of the same information are possible, such as a table, chart for management.

The Controller- It is getting input and mapping values to model and input validation and business logic done here.

Interactions

In the application into three kinds of components, the model, view, controller are interacting with them.

A controller can send commands and get values and map the values to model. Model carrying the values and bind into view.

A **model** is used for stores data from controller and pass into the view.

A **view** part is providing GUI for output presentation to the end user. It is displaying model values to user.

A **controller** generates values to model and send into view.

Use of Web Application

Currently web application is widely using technology. It is easy to access from anywhere. This is providing better interaction with user. Web application is available in various fields like commercial, non-commercial, social media etc. We can easily achieve client server concept.

In this MVC approach, Browser is sending http request to server. This request to routing to controller. Controller is getting input form view and implement business logic. Once it all business logic completed then it is preparing model and send into view. Client side script is main role in web application such as Angular JS, Ember JS, Java Script MVC and Backbone have been created and allow MVC components to execute on the client. We can achieve unit test by using Qunit.

Integration Process

Continuous Integration process starts with a local development. Development are committing their work to Version Control repository. Based on the changes in the version control repository, the Continuous integration server starts the build of the application. The integration build of the application includes:

- Source code and compilation
- Testing
- Code inspections and Deploy

Source Code Compilation

Source code compilation important factor of development and it is depends on the type of the used language. Currently there are many tools available in market which is maintain all the build activities.

Testing

Automated testing should be resolve most of the issue and it will give minimum level assurance to application. There are several type of testing using on several level. Developer can identifies code coverage level.

- Unit test – to test the reliability of the code and code coverage.

- Component test - to verifies features of modules
- System test – to exercise the complete software system
- Function test – to test functions and user interface of the software system

Automated Functional Testing Tool

During the development of a web application, Functional tests is necessary to ensure the quality of software.

- Lower the probability of user interface or functionality error in production environment
- Faster revelation of an error
- One time setup and multiple time ensure quality of product.

Future of Continuous Integration

The CI process brings higher quality of the application development. Integration the application continuously, the developers can easily respond to the change of requirement, easy to find an error. One of the key the CI best practice is that everyone should commit to the mainline every day, it is necessary to avoid committing a broken code. However, in the standard, the code is tested during the build, which means that the broken code is already committed in the version control repository. Team city used by us as a continuous integration server brings a way of keeping the work on project going and also improves the quality of the continuous integration processes, Pre-tested commit.

Conclusion

The best practice of continuous integration is using the agile development process. The development team could use scrum model because it is very easy to gathering information and developing the web application. MVC design pattern is applied to this technique which reduces the drawback. The developer can continuously testing each process. Through this approach time will be saved and cost will be minimized without compromising quality of the product.

Bibliography

1. HANBURY VENTURES LTD. Available from: http://www.prozeny.cz
2. BECK Kent, Mainfesto for agile software Development(2001). Available from: http://agilemainfesto.org
3. SCHWABER Ken. Agile project management with scrum.
4. R.Owen. Scaling Continuous Integration [online]. Available from: http://www.springerlink.com/index
5. Steve MATHYAS, Andrew GLOVER. Continuous integration: improving software quality and reducing risk.
6. JETBRAINS. Issue Tracker and Agile Project Management Tool[online].2009. Available form: http://jetbrains.com/youtrack/
7. JETBRAINS, Continuous integration for Everybody- Teamcity[online],2006. Available form: http://www.jetbrains.com/teamcity/
8. Mercurial SCM. Mercurial SCM[online],2009. Available form: http://mercurial.selenic.com/
9. SDLC- overview [online] Available form: http://www.tutorialspoint.com/sdlc/
10. Model-view-controller- Wikipedia[online] Available form: http://en.wikipedia.org/wiki/
11. Design Pattern MVC Pattern[online] Available form: http://www.tutorialspoint.com/design_pattern/mvc_pattern.htm
12. Scrum Method[online] Available form: https://en.wikipedia.org/wiki/Scrum_(software_development)