DEVELOPMENT OF SOFTWARE FOR PROJECT MANAGEMENT BASED ON PMBoK

Zh. Madimova¹, L. Atymtayeva²

Kazakh-British Technical University, Department of Computer Engineering

¹Tole bi 59, 050000, Almaty, Kazakhstan
²E-mail: zhanar.madimova@gmail.com
²E-mail: I.atymtayeva@gmail.com

Nowadays effective project management leads to success of the project as a whole. Therefore it is very important to know how to manage them in most effective and accurate way. Numerous guides and applications are considered to solve this problem. However variety of software for managing documents electronically and based on specific guide is pure. The purpose of this article is to introduce the developed web application based on PMBoK guide, which simplify project documentation management. The article provides general information about PMBoK guide, methods used in web application development and shows its class diagrams.

Keywords: project management, PMBoK, MVC

1. Introduction

Today project management in information technologies sphere is the most rapidly developed knowledge area. Therefore wide range of standards, books and guides are published. It is very difficult for beginner project managers to get success in this sphere due to variety of terms and rules. As a rule, it is followed by huge proportion of unfinished projects towards success ones. The solution of this problem is development of conceptually new software for project management based on specific methodology.

First, let me briefly determine termins of project management. According to the PMBoK, a widespread guide among project managers, project is a temporary endeavour undertaken to create a unique product or a service. The fundamental base of every project is the “triangle of the project”. The main purpose of project managers is to do everything to hold the project inside this triangle. The sides of triangle are price, structure of works and time. The project, which was inside of this triangle during the whole its lifecycle, considered as successful project [1, 2].

Unfortunately, it is very difficult to watch after software projects. Therefore it is reasonable to separate them into stages and to provide every step with documentation, because the documentation is one of the most important indicators of the project.

However, project management have experienced project separation into the stages during its history. Thus, these works have been done for us. Our aim is to choose the best one. After examining different guides, the authors have decided that PMBoK guide is the most suitable on this case, because in this guide processes are divided into knowledge areas and groups of processes, which can be successfully applied as a base of web application.

2. Project Overview

The main purpose of the tool is development of unified web application which helps to manage projects by simplifying the organization of documents in effective way. The programming language is PHP. Also to provide the extension and integration with other projects the MVC pattern will be used. So its’ view and business logic will be separated.

As it was mentioned before, document management providing by application will be based on knowledge areas and process groups in the PMBoK guide. That is, processes will be located in the cells of these concepts intersection. User will allow uploading necessary document into this cell. If user do not know what he should include into the document he will able to download a sample or, in special cases, fill provided forms.
Generally, tool will provide the following functions:

- Authentication;
- Creation of new project;
- Uploading/downloading documents;
- Creation of some specific documents by filling forms;
- Report generation from forms.

3. The MVC Pattern Implementation

As it was mentioned before the MVC pattern would be used to build the web application. Let us to briefly introduce this concept.

The MVC (Model-View-Controller) architecture is a way of decomposing an application into three parts: the model, the view and the controller. It was originally applied in the graphical user interaction model of input, processing and output.

A model represents an application’s data and contains the logic for accessing and manipulating that data. Any data that is part of the persistent state of the application should reside in the model objects. The services that a model exposes must be generic enough to support a variety of clients. By glancing at the model's public method list, it should be easy to understand how to control the model's behaviour. A model groups related data and operations for providing a specific service; these group of operations wrap and abstract the functionality of the business process being modelled. A model’s interface exposes methods for accessing and updating the state of the model and for executing complex processes encapsulated inside the model. Model services are accessed by the controller for either querying or effecting a change in the model state. The model notifies the view when a state change occurs in the model.

The view is responsible for rendering the state of the model. The presentation semantics are encapsulated within the view; therefore model data can be adapted for several different kinds of clients. The view modifies itself when a change in the model is communicated to the view. A view forwards user input to the controller.

The controller is responsible for intercepting and translating user input into actions to be performed by the model. The controller is responsible for selecting the next view based on user input and the outcome of model operations.

The figure below shows the implementation of this concept in the main part of application.
It can be clearly seen from the Figure 2, that the View part of the application only displays data, neither operate with them. It is the part, which interacts with users. All data entered in the view will be passed to Controller class. Controllers connect Model and View. Models are core of the system; its domain logic.

For example, let’s see how the list of particular user projects is displayed in our application by using this concept. Firstly, we should define in the model class the method, which gets projects titles of users, by their id.

```php
public function getUserId($user_id)
{
    $select = $this->select()->where('user_id=' . $user_id)->order('id DESC');
    $row = $this->fetchAll($select);
    return $row;
}
```

Then we should implement this method in controller and push results into the view.

```php
public function indexAction()
{
    $projects = new Application_Model_DbTable_Projects();
    $auth = Zend_Auth::getInstance();
    $user_id = $auth->getIdentity()->id;
    $this->view->projects = $projects->getByUserId($user_id);
}
```

And finally display results in view.

```
My projects
-------------
Sample2
-------------
Sample1
-------------
```

Figure 2. The MVC concept in the applications' class diagram

Figure 3. The displayed view
Benefits of the MVC concepts are the following:
1. Code reuse – Separation of concern principle will provide code reusability as the design will have proper domain model and business logic in its logical unit.
2. Adaptable design – A good design is close for changes and open for additions. Isolated code in presenter/controller, domain model, view and data access provide a freedom of choosing a number of views and data sources.
3. Layering – MVC/MVP forces to separate data access login for the other layers and various other patterns would be opted to implement data access layer.
4. Test driven approach – Isolated implementation allows testing each component separately. Especially in MVP pattern that uses interface for a view, it is a true test driven approach.

4. Conclusions

The Republic of Kazakhstan is one of the most rapidly developed countries in information technology. However, most of companies, government structures use software, which has been developed in the foreign countries. This application will be one of the first domestic applications for project management. Despite its simplicity, it covers all complex PMBoK concepts. The application helps to manage projects documentation as well, so as helps to understand PMBoK.

In conclusion, software for project management is universal instrument of effective project management, The MVC pattern, which is used to develop this application, provides possibility to extension of tool. In future it can be upgraded by adding such functionalities as risk analysis, planning, cost estimation, etc. The user of the application takes such benefits, as a hard disk memory economy and an accurate document management.

References


Received on the 21st of July 2012

48