The development of vocational high school information system using angularJS and scrum

The development of vocational high school information system using angularJS and scrum

R Nabila[1], S Oktaviana[2], and A Hidayati[3]

[1] Faculty of Computer Science, University of Indonesia. Depok - Indonesia
[2,3] Informatics Engineering Major, Jakarta State Polytechnic. Depok - Indonesia

 rusyda.nabila@ui.ac.id

Abstract. Scrum is one of the most familiar and commonly used agile models. In this paper, we report the result of system development using AngularJS along with Scrum methodology. This research has shown that the use of AngularJS framework along with Scrum was successfully reduce the development time. This paper makes contribution by exploring a match combination between a popular system development methodology and a programming language.
Keywords – AngularJS, Scrum, Web Development, Information System

1. Introduction

PT. X conducts Corporate Social Responsibility (CSR) provides the assistance in the form of coaching, training, and certifying for teachers and students in 1000 vocational high schools in Indonesia. The process of data collection is used to support the decision-making process. However, it takes 14-30 working days to collect them. It's because of the data collection manual process. Based on the explanation above, an information system is needed to store and manage data on vocational high schools' teachers and students. This information system is expected to facilitate and accelerate collecting data structurally.

The system development chosen is Scrum. Scrum provides an iterative and incremental approach by divide it into small cycles, called Sprint. One sprint has a month or two weeks duration [1]. Due to lack of development time, each sprint only lasts one week. AngularJS framework is chosen because of its advantages that could greatly help to reduce the development time [2]. This paper makes contribution by exploring a match combination between a popular system development methodology and a programming language.

2. Literature Review

AngularJS is a front-end JavaScript framework developed by Google. AngularJS can be used to build a single page application using the Model View Controller (MVC) architecture [3]. AngularJS has several advantages, i.e. providing the ability to build a Single Page Application, providing data capabilities in binding HTML, the code can be tested based on units or modules, using dependency injection, providing components that can be reused, developers can create a function with more concise code and a pure display using HTML and a controller written in JavaScript to run business processes [4].

The system development method used is Scrum methodology, which is a framework where teams can solve complex problems that are constantly changing, which at the same time produce high-value products creatively and productively. Figure 1 shows the Scrum method using an iterative and incremental approach to improve predictability and control risk [5].

In Scrum development there are several activities done, namely sprint, sprint planning, daily scrum meetings, sprint reviews, and sprint retrospective meetings. Before the sprint is implemented, the project owner will list the priorities of the desired project or called the product backlog. Sprint is the basic unit in the Scrum method. Sprint is a period of time that is limited into duration, for example two weeks, three weeks, or one month. Each sprint begins with sprint planning and ends with sprint review and sprint retrospective meeting. Sprint planning is a meeting between the team and the project owner to define the work to be completed in one sprint. In sprint planning, a sprint backlog is made in the form of a collection of work approved by the team to be completed in one sprint. Daily scrum meetings are team meetings every day to review the work results. Sprint review is a meeting to evaluate what has been completed during the sprint. In sprint review, burndown chart is usually used as an evaluation tool for team progress. The Sprint retrospective meeting is a team meeting to find ways to improve product and system development process. Figure 2 is a step in the Scrum method [6].

Black box testing is one type of trial that plays an important role in system testing. At the time of testing, the examiner has no knowledge of internal design and cannot access the source code. Testers only know the system architecture. This technique is used to ensure that all inputs needed by the system are well-received and get the correct output [7].

Figure 1. The Scrum method approach [5].   **Figure 2.** Stages in the Scrum method [6].

3. Result and Analysis

3.1. System Requirements Analysis

The front-end of vocational high school, information system is built with the AngularJS and Bootstrap framework. The system development method used is Scrum with four sprints in a month. Normally, one sprint has a month or two weeks duration [1]. Due to lack of development time, each sprint only lasts one week. The first sprint is carried out the construction of the school and data verification modules, the second sprint is carried out by the teacher module construction, the third sprint is carried out by the student module construction, and the fourth sprint was carried out by user acceptance test using black box testing. Table 1 is an estimation time of the project team to build a vocational high school information system.
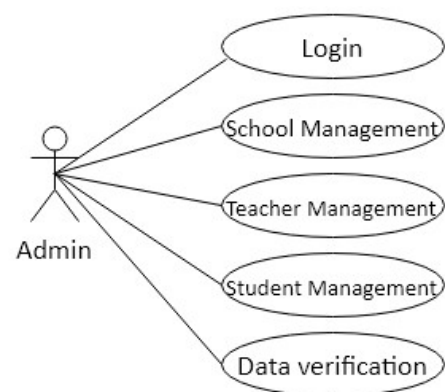
Table 1. The Project Team's Estimation Time.

Total Sprint	4 weeks		
Time	20 days		
Role	**Days**	**Hours/day**	**Total Sprint Hours**
Project Manager	20	4	80
Scrum Master	20	5	100
Programmer 1	20	8	160
Programmer 2	20	8	160
Programmer 3	20	8	160
Programmer 4	20	8	160

3.2. System Design

3.2.1. Menu Structure. Figure 3 is the menu structure of system. There is an additional menu on each menu that is verification. In the verification menu, the admin can verify data that has been entered before printing into the report.

3.2.2. Use Case Diagram. Figure 4 is the use case diagram of system. Admin logs in to enter the system. After logging in, the admin can do schools management, teacher's management, student's management, and verification of the data that has been entered. Admin can add, edit, delete, download excel, view, and search for data.
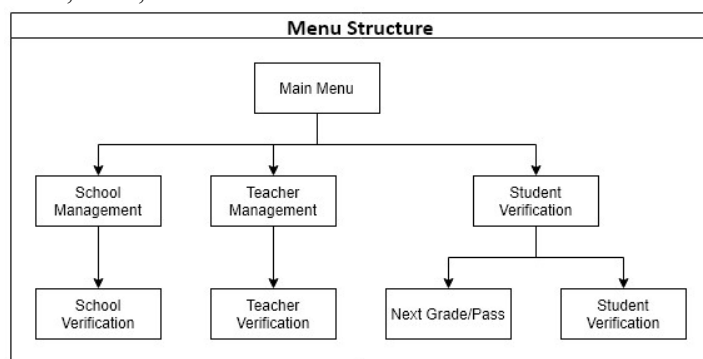
Figure 3. Menu structure of system.   **Figure 4.** Use case diagram.

3.3. Implementation

3.3.1. Management Page. Figure 5 is the code controller piece on the teacher management page. The controller functions as a process controller. The code snippet in Figure 5 contains a function to display a list of teacher data. Figure 6 is the model and view code snippet on the teacher management page. View is written in HTML tags, like <table>. The model is written in the AngularJS tag, such as {{teacher.nip}}. Figure 7 is the interface of teacher management page. There is a button to add, edit, delete, search, view, and download of teachers' data. Implementation on the school and student management page is not much different from Figure 5-7.

```
var teacher = function () {
    self.statusData_temp = self.statusData.keyName;
    self.statusGuru_temp = self.statusGuru.keyName;
    self.name_temp = self.name;

    teacherService.list_of_teacher(self.currentPage - 1, self.limit,
        self.statusData.keyName, self.statusGuru.keyName, self.name)
    .then(function (response) {
        if (response.message !== 'ERROR') {
            self.list_of_teachers = response.result;
            self.bigTotalItems = response.elements;
            self.numPages = response.pages;
            if(self.list_of_teachers.length == 0){
                self.find_nothing = true;
            }else{
                self.find_nothing = false;
            }
        }
    );
    };
};
```

Figure 5. Controller code snippet on the teacher management page.

```
<table class="table table-bordered table-striped">
    <thead>
        <tr>
            <th>No</th>
            <th>Kode SMK</th>
            <th>Nama SMK</th>
            <th>NIP</th>
            <th>Nama Guru</th>
            <th>Jabatan Fungsional</th>
            <th>Status Guru</th>
            <th>Status Data</th>
            <th>Tanggal Terakhir Update/Verifikasi</th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        <tr ng-repeat="guru in list_of_teachers">
            <td>{{getNumber($index + 1)}}</td>
            <td>{{guru.smkCode}}</td>
            <td>{{guru.smkName}}</td>
            <td><a ng-click="profileGuru(guru.id)">{{guru.nip}}</a></td>
            <td>{{guru.name}}</td>
            <td ng-if="guru.active == true">Aktif</td>
            <td ng-if="guru.active != true">Tidak Aktif</td>
            <td ng-if="guru.statusData == 'VERIFIED'">Sudah Diverifikasi</td>
            <td ng-if="guru.statusData == 'NOT_VERIFIED'">Belum Diverifikasi</td>
            <td>{{guru.modificationDate | date:"dd-MMM-yyyy"}}</td>
            <td>
                <button type="button" ng-click="idGuru_edit(guru.id)" tooltip="Ubah" ng-show="hasEditPrivilege">
                    <i class="fa fa-edit" aria-hidden="true"></i>
                </button>
                <button type="button" ng-click="idGuru_delete(guru.id,guru.name,guru.version)"
                    tooltip="Hapus" ng-show="hasDeletePrivilege">
                    <i class="fa fa-trash-o" aria-hidden="true"></i>
                </button>
            </td>
        </tr>
        <tr>
            <td colspan="15" ng-show="find_nothing" align="center">Maaf Tidak Ada Data</td>
        </tr>
    </tbody>
</table>
```

Figure 6. Model and view code snippet on the teacher management page.

Figure 7. Teacher management page interface.

```
<table class="table table-bordered table-striped">
    <thead>
        <tr>
            <th ng-show="hasVerifyPrivilege">Verifikasi<br><input type="checkbox"  ng-click="checkAll()" ng-model="selectedAll"></th>
            <th>No</th>
            <th>Nama Guru</th>
            <th>NIP</th>
            <th>Jabatan Fungsional</th>
            <th>Status Guru</th>
            <th>Status Data</th>
        </tr>
    </thead>
    <tbody>
        <tr ng-repeat="guru in list_of_teachers">
            <td ng-show="hasVerifyPrivilege"><input type="checkbox" ng-model="guru.selected" ng-change="isLabelChecked()"></td>
            <td>{{$index + 1}}</td>
            <td>{{guru.name}}</td>
            <td><a ng-click="getDetail(guru.id)">{{guru.nip}}</a></td>
            <td>{{guru.position}}</td>
            <td ng-if="guru.active == true">Aktif</td>
            <td ng-if="guru.active != true">Tidak Aktif</td>
            <td ng-if="guru.statusData == VERIFIED">Sudah Diverifikasi</td>
            <td ng-if="guru.statusData != VERIFIED">Belum Diverifikasi</td>
        </tr>
        <tr>
            <td ng-hide="find_nothing">Maaf Tidak Ada Data</td>
        </tr>
    </tbody>
</table>
```

Figure 8. Model and view code snippet on the teacher verification page.

```
self.verify = function(){
    input();
    self.isDisabled = true;
    teacherService.verify_teacher(request).then(
        function (response) {
            if (response.message !== 'ERROR') {
                var hasil = response.result;
                var modalVerify = $modal.open({
                    templateUrl: '/views/modal/verifyTeacherModal.html',
                    controller: 'verifyTeacherModalCtrl',
                    size: 'md',
                    backdrop: 'static',
                    resolve: {
                        modalParam: function () {
                            return {hasil: hasil};
                        }}});
            }else{
                self.isDisabled = false;
                $modal.open({
                    templateUrl: '/views/modal/Modal.html',
                    controller: 'ModalCtrl',
                    size: 'sm',
                    resolve: {
                        modalParam: function () {
                            return {
                                title: 'Gagal',
                                message: response.result,
                                path: ''
                            };}}});
            request.listToVerify = [];
        }});};
```

Figure 9. Controller code snippet on the teacher verification page.

Figure 10. Teacher verification page interface.

3.3.2. Verification Page. Figure 8 is the model and view code snippet on the teacher verification page. The model is written on HTML tag, while the view is written in the AngularJS tag. Figure 9 is a piece of code controller on the teacher verification page. Figure 9 shows the function to verify teacher data. Figure 10 is the display of the teacher verification page. There is a button to search and verify teachers' data. Implementation on the school and student verification page is not much different from Figure 8-10.

3.3.3. Burndown Chart. Burndown chart (Figure 11) is used to evaluate team performance. Table 2 is a breakdown of time in the construction of system. Burndown chart is formed based on data in Table 2. The Burndown chart will compare the remaining effort and ideal trend values. Remaining effort is the amount of time left, while the ideal trend is the ideal amount of time left for the team to complete the project. Figure 11 shows that in the first and second sprint, the performance has more decreases than increases in the third sprint and finishes on time in the fourth sprint.
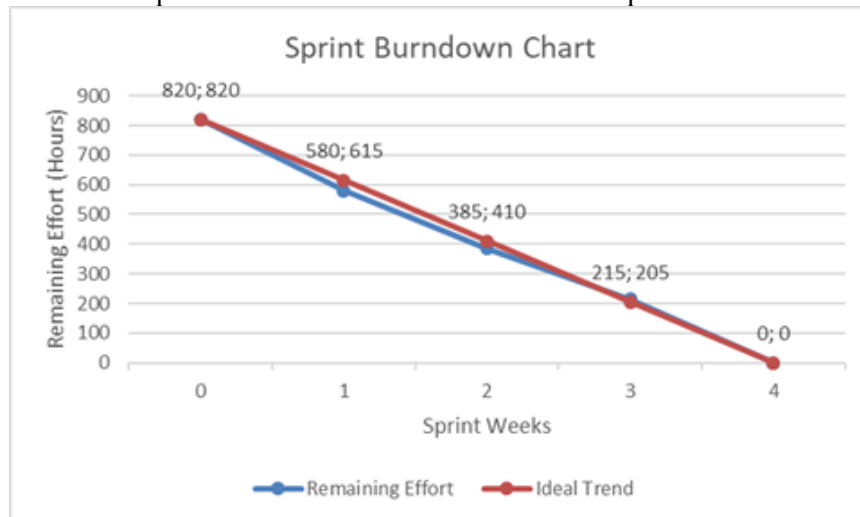

Figure 11. Burndown chart of vocational information system projects.

Table 2. Time Breakdown.

Task	Person in Charge	ET	S1	S2	S3	S4
Business Process Design	Project Manager	20	20	0	0	0
Interface Design	Scrum Master	20	20	0	0	0
Database Design	Scrum Master	20	20	0	0	0
Front-End Coding	Programmer 3	160	30	40	50	40
	Programmer 4	160	20	40	45	55
Back-End Coding	Programmer 1	160	55	55	20	30
	Programmer 2	160	55	40	35	30
Testing	Project Manager	60	10	10	10	30
	Scrum Master	60	10	10	10	30
Remaining Effort		**820**	**580**	**385**	**215**	**0**
Ideal Trend		**820**	**615**	**410**	**205**	**0**
Notes	ET : Estimation Time (in Hour) S1 : Sprint Week 1					

3.4. System Testing
The system testing is done using black box testing. All results from the test are successful. Table 3 is the result of the teacher management page test. The results on other page are not much different from Table 3.

Table 3. Teacher management page testing results.

No	System Component Tested	Expected Results	Test Results
1	Access the teacher management page	The teacher management page appears	Success
2	Creating a new teacher data	A message appears "Teacher successfully added"	Success
3	Editing teacher data	The message "Teacher data successfully saved" appears	Success
4	Deleting teacher data	A message appears "Teacher has been deleted"	Success
5	Searching for teacher data based on search filters	The teacher data appears	Success
6	View details of SMK data	Enter the SMK Profile page	Success
7	Verifying teacher data	A message appears "Data successfully verified"	Success
8	Download excel teacher data	Data is stored in Excel and automatically downloaded	Success

4. Conclusion

Based on the research that has been done, it can be concluded that the use of the AngularJS framework along with Scrum can speed up the system development time. Programmers don't need to build code architecture from scratch. The MVC architecture applied by AngularJS can facilitate and speed up the process of testing and repairing the system. This system implements Single Page Application, which is an advantage of AngularJS, so the system does not need to do a lot of reloading pages that can spend more bandwidth.

Figure 11 show that the team successfully completed the project smoothly. This is shown from the comparison between remaining effort and ideal trend that is not much different.

References

[1] Anwer, Faiza, et al 2017 Comparative analysis of two popular agile process models: extreme programming and scrum *Int. J. of Computer Science and Telecommunications* **8.2** pp 1-7.

[2] Chansuwath, Wutthichai, and Twittie S 2016 A model-driven development of web applications using AngularJS framework Computer and Information Science (ICIS) *2016 IEEE/ACIS 15th Int. Conf. on. IEEE.*

[3] Haviv A Q 2014 *MEAN Web Development* (Birmingham: Packt Publishing).

[4] Waghade V and Chaudhari B V 2016 Study of angularJS with other frameworks *Int. J. of Research in Computer & Information* **1** no 2 pp 151-4.

[5] Sutherland J and Schwaber K "The Scrum Guides," Scrum.Org dan ScrumInc. , Juli 2016. [Online]. Available: http://www.scrumguides.org/docs/scrumguide. [Accessed 5 Desember 2016].

[6] Permana P A G 2015 Scrum method implementation in a software development project management *Int. J. of Advanced Computer Science and Applications* **6** no 9 pp 198-204.

[7] Jan S R, Shah S T U, Johar Z U, Shah Y and Khan F 2016 An innovative approach to investigate various software testing techniques and strategies *Int. J. of Scientific Research in Science, Engineering and Technology* **2** no 2 pp 682-9.