Modeling agile development of Web application e-monev using UML

I Made Suartana¹*, Pradini Puspitaningayu², Satwika Arya Pratama³, Sri Dwiyanti⁴, Maspiah⁵, Subuh Isnur Haryudo⁶

¹Department of Informatics Engineering, Universitas Negeri Surabaya, Indonesia
²Department of Electric Engineering, Universitas Negeri Surabaya, Indonesia
³Department of Nutrition Study Universitas Negeri Surabaya, Indonesia
⁴Department of Cosmetology Education, Universitas Negeri Surabaya, Indonesia
⁵Department of Cosmetology Education, Universitas Negeri Surabaya, Indonesia
⁶Department of Electric Engineering Education, Universitas Negeri Surabaya, Indonesia

Abstract. Monitoring and evaluation (Monev) are critical to determining whether the plan has been implemented and learning for future policies and planning. In education, the information obtained from monitoring and evaluation results is beneficial and will help improve existing processes sustainably. Implementation of monitoring and evaluation using printed manual documents requires time in processing the monitoring and evaluation results, giving rise to a tendency to spend much time in the implementation process. At the same time, there are essential things to do next, namely, evaluation and follow-up of the monitoring and evaluation process. An integrated solution is needed to implement monitoring, evaluation, and reporting so that the monitoring and evaluation process can be emphasized on assessment and follow-up, not monitoring. This research aims to design a prototype e-monev application using a web platform. This system is intended to be used to monitor and evaluate lectures. E-Monev was developed using an Agile and Collaborative Model Driven Development (AC-MDD). AC-MDD helps teams identify and manage interactions between business processes, domains, software, and creative design in Web engineering projects. UML was chosen in the E-Monev system design modeling. UML was expected to assist the team in communicating, designing, and validating software architectural designs. Agile Modeling with UML can be modeling the E-Monev system by adopting changes that may arise due to policy or rule changes.

1 Introduction

Monitoring and evaluation are crucial in all projects, programs, and policies related to education. Monitoring is essential because every educational system involves policy, planning, establishing targets, and implementation, and monitoring and evaluation allow progress toward those objectives to be examined. Monitoring helps to identify what has been successful and what has failed. At the same time, evaluation examines the relevance,
effectiveness, efficiency, impact, and sustainability of education policies, plans, strategies, projects, and programs. The information obtained from monitoring and evaluation results is highly beneficial in improving existing processes in education in a sustainable way [1].

A solution for monitoring and evaluation is the development of a web-based monitoring system. Web applications offer advantages in terms of easy deployment, use, and scalability for large organizations. Various monitoring systems have been created on a web basis, such as the monitoring and evaluation system for the SMARTD program. The primary goal of developing this system is to facilitate the wide range of monitoring and evaluation programs being carried out [2]. [3] Another web-based monitoring and evaluation system aims to overcome monitoring and evaluation challenges, simplify work, generate quality data, and provide successful and timely project implementation. Research [4] has also developed a model for monitoring and evaluating the performance of web-based vocational school teachers in the form of SIMoney PKG Online.

Regarding web development, the methods and processes are crucial in determining the system's success. Numerous studies have been conducted on the topic, including one that suggests using a hybrid approach with the Agile Modeling Method Engineering (AMME) and Resource Description Framework (RDF) frameworks [5]. Another study proposes integrating Human-Centered Design techniques into Agile software development to model Agile requirements. This approach involves using a metamodel to formalize the Context-Based Persona Story (CBPS) concept [6]. Additionally, a framework called Agile and Collaborative Model Driven Development (AC-MDD) has been developed for web applications to reduce errors in modeling and documentation [7]. Many web development studies have adopted the Agile model due to its iterative and flexible nature, making it suitable for software development in various cases. Web modeling ensures effective communication between the development team and consumers in addition to the development process. Design Visuals facilitate communication between groups and end consumers. The Unified Visual Language (UML) is an industry-standard modeling language that enables system modeling. Numerous software development methodologies employ UML for modeling or design. [8].

This study aims to create a model for e-monev applications utilizing a web platform. Agile development techniques are being implemented to aid users in refining or agreeing on a system design while ensuring that everyone reaches a consensus. Unified Modeling Language (UML) diagrams are being used for modeling visualization.

2 Agile Web Development

2.1 Agile Development

Agile web development is a methodology founded on the Agile Software Development Manifesto. In 2001, a team of professional developers created it. This method's primary objective is to provide flexibility and customer satisfaction through the early and continuous delivery of valuable software.

Within the agile methodology, the majority of web development stages coincide. All team members, from decision-makers to designers and content creators, should organize meetings in the early phases of web development. All team members' understanding of the project requirements at all stages reduces the need for constant emails, phone calls, and meetings. Due to the agile methodology, work can be completed much more quickly. The typical agile web development procedure consists of several "sprints." Each iteration encompasses the phases of discovery, design, development, and testing. As shown in Figure 1, the steps of the Agile methodology are scaled-down and repeated for each additional
product. The increment is then distributed across iterations and implemented. After each cycle, functional product enhancements are delivered. An agile software team must possess all the necessary skills to deliver a working product after each Sprint, a term used in Scrum for a single iteration.

Fig. 1. Agile Process development [9].

2.2 UML-Based Agile Web Development

UML-based Agile Web Development uses UML extensions for modeling activities in Agile Web Development. A navigation model, a component communication model, a conceptual model, and an architectural model comprise the UML model generated for web applications. Consequently, developers can benefit from model-driven and test-based development. The cyclic procedure has two phases. Figure 2: Analysis. A requirements analysis and conceptual model (as a class diagram) are generated. It also defined an architecture (as a component diagram). The construction of. This phase includes the Build and Sophistication subphases. During a build, the developer iteratively selects a subset of requirements and constructs a storyboard that depicts how the requirements are realized via user interaction. Sophistication necessitates meticulous planning and execution. Existing subsystems are integrated with newly implemented components, and integration/regression testing is performed. This cycle continues until all requirements are satisfied.
3 Methodology

Agile development methodology is starting to be applied in web-based projects on a large scale by software developers. The use of a method involves more than just related processes or documentation. Based on this, it needs reasonable consideration in identifying the methodology, company needs, size and quality of the project, and especially the characteristics of Agile development before being implemented in system development. The characteristics of the project, the challenges that will be presented during the process, the suitability of each stage, and the requirements of the end consumers are among the factors to consider when selecting the most appropriate methodology. [10].

This study utilized Agile processes, models, and tools. The Agile approach was chosen due to the highly dynamic nature of the system being developed, so the method is expected to accommodate changing needs quickly. This study focused on using the UML modeling language for system modeling. Modeling also focuses on the navigation and communication aspects of web-behavior. In addition, this model will serve as a reference for system developers. The Agile and Collaborative Model Driven Development (AC-MDD) [11] framework model was chosen for modeling frameworks. The AC-MDD framework process is described in Figure 3.

In the first activity, "Web Application Business Agile Modeling," we must define technical requirements and user stories that will be refined, modeled, developed, and tested during the Sprint. Its definition should consider the technical capabilities and burden of the development team. Depending on the complexity of the Web application and the number of hours required by the development team, there may be multiple Sprints. After selecting a set of technical requirements and user stories, systems analysts and stakeholders refine their respective models for the Web Application Development subactivity.

In the subsequent step, without stakeholder participation, the systems analyst must collaborate closely with the developer to complete the Sprint-required sub-activity of Detailed Agile Modeling of Features. Suppose additional refinements are required after this sub-activity. In that case, you can revert to the previous sub-activity, Requirements Modeling and Agile User Stories, to reduce the likelihood of future issues in later phases.
The sub-activity Detailed Agile Modeling of Features employs the Just in Time (JIT) concept. Given the application's characteristics, the primary output of this activity is the principal class model and Computer Human Interfaces (CHI).

The following sub-activity is the Agile Modeling of Web Application Functional Tests performed by testers and developers. To verify system behavior, they must link to the agile test modeling user story in this activity. They should concentrate on functional testing, perusing experiences, success scenarios, and use case alternatives. If enhancements are required, you can return to the Agile Modeling of Features Subactivity after this activity.

![Diagram](image)

**Fig. 3.** Modelling process.

Figure 3 shows the modeling step using the AC-MDD framework. The framework focuses on increasing application development productivity, conforming to requirements, and generating source code based on an agile model. AC-MDD framework reduces the unnecessary use of resources during the web application's modeling and documentation.
phase, which entails collaborative work among stakeholders. Each step, 1 to 4, is described in table 1.

Table 1. Modeling process summary step.

<table>
<thead>
<tr>
<th>Step</th>
<th>Step goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Assist the development team in comprehending the scope of the web application by identifying the problem and the application's interactions with other elements. E-Monev Profile: Context Diagram Elements.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Identifying and documenting the stakeholders' macro-requirements. E-Monev Profile: Requirement, Components.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Define the requirement scope for the element sprint, Elements of E-Monev, and Development Sprint Scope Profile.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Use Case and Business rule modeling. E-Monev Profile: Use Case Diagram Components.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Model in an agile manner the user stories to detail the requirement identified in the Sprint, complementing use case information. Elements of Web-AML profile: Domain classes.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Create an agile model of web application domain classes, focusing on Sprint's use case and user stories and element of Web-AML profile: Domain classes.</td>
</tr>
<tr>
<td>Step 7</td>
<td>Minimize the flaws found in the agile modes on the business level that may directly impact the development of Sprint's functionalities. Elements of Web-AML profile: Adjustment in the agile model of the development sprint.</td>
</tr>
<tr>
<td>Step 8</td>
<td>Create an agile model of the web application functionalities to be developed, focusing on the current development sprint, to represent the computer-human interface(CHI).</td>
</tr>
<tr>
<td>Step 9</td>
<td>Create agile functional test.</td>
</tr>
</tbody>
</table>

The modeling phase's objective is to contextualize better a Web application's primary functional and non-functional requirements. Agile business modeling is implemented in Sprint 0 (early stage). Participating in this initial phase are systems analysts and stakeholders, who, through collaborative decision-making, seek to mitigate waste in requirements gathering. Unified Modeling Language (UML) is the modeling notation used.

4 Case Study

4.1 Case study

This paper utilizes a test Case study of designing a web application for monitoring and evaluating learning (E-Monev). E-Monev is an application for monitoring university lecture processes and monitoring conducted on the preparation of teaching lecturers, Implementation of Learning, Evaluation of Learning, and Program Learning Outcome (PLO) Measurement.

Developing a web-based monitoring system has become a monitoring and evaluation solution. Large organizations benefit from the simplicity of deployment, usability, and scalability of Web applications. The development of a web-based monitoring and evaluation system aims to overcome monitoring and evaluation challenges, simplify work, generate high-quality data, and ensure the successful and timely implementation of projects.
4.2 The main result

As depicted in Figure 4, the primary function of the monitoring and evaluation process begins with the Administrator registering system users and contributing data that serves as the auditor's point of evaluation. Additionally, the Administrator determines which auditors will evaluate each existing Locus. As an auditor, you must enter the system using user credentials. After entering the system, the auditor can determine and input data for the Locus, which falls under the assessment's purview. The data from the auditor's evaluation findings will be used to generate a report on the monitoring results. The audit results report becomes relevant information for the manual evaluation that stakeholders will conduct.

Sprint 01 only examines the appearance of its profile page (dashboard page) as an Ordinary User. As Administrator, the user can perform all E-Monev management and fundamental CRUD (Create, Read, Update, or Delete) database operations (Sprint 02). This article provides fragmentary examples of design documents and does not describe each User Story in Figure 4 in exhaustive detail. Table 2 illustrates the proposed User Story for evaluation and monitoring data input in Step 5 of the AC-MDD Method.

![Fig. 4. Use Case Diagram.](https://example.com/image)

Table 2. User Story of Evaluation Data Input.

<table>
<thead>
<tr>
<th>Actor</th>
<th>Auditor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td>Input the data from the assessment of the learning process at the Locus assessed during the evaluation.</td>
</tr>
<tr>
<td><strong>Reason</strong></td>
<td>Identifying and documenting the stakeholders' macro-requirements. E-Monev Profile: Requirement, Components.</td>
</tr>
</tbody>
</table>
User Story

<table>
<thead>
<tr>
<th>Actor</th>
<th>Auditor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Important</td>
<td>Portability and Responsive Web Interface</td>
</tr>
<tr>
<td>Less Important</td>
<td>Ordering Locus, according to the user choice.</td>
</tr>
<tr>
<td>Priority</td>
<td>5 High</td>
</tr>
<tr>
<td>Desirable Time</td>
<td>1 Month</td>
</tr>
<tr>
<td>Estimated Effort</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Sprint</td>
<td>Sprint 02</td>
</tr>
<tr>
<td>Team</td>
<td>Made Suartana</td>
</tr>
<tr>
<td>Comments</td>
<td>This user story can only be developed after developing the user story of Input data evaluations.</td>
</tr>
</tbody>
</table>

After implementing the AC-MDD Method, Sprint 01 functional test agile modeling produces test cases. Figure 6 depicts a test case intended to verify two questions: Test Login With Correct User - An existing user logs in to the web application successfully. Login Test With User Not Found - A user not registered in the database cannot access restricted pages.

Step 10 of the WebAC-MDD Method was executed after the agile model of the functionality defined in Sprints 01, and 02 was completed to generate the source code for this model. Based on the agile modeling depicted in Figure 5, Step 10 of the AC-MDD method generates source code automatically to implement web application functionality. Figures 7 and 8 illustrate the CHI representing the CRUD of the auditing procedure.

Fig. 5. Agile Model Input Evaluation and Modelling Data
It should be emphasized that an agile approach, which generates source code from an agile model, is proposed to increase productivity in Web Application development. The objective is to reduce resource waste during collaborative web application modeling and
documentation phases. Since the agile model used in this study concentrates on adding value to the creation of source code through modeling, it provides a substantial productivity increase in application development web. To ensure that web application development can be conducted more systematically and applications can be adapted to user needs.

5 Conclusion

This paper uses an Agile framework for web application development that aims to increase productivity by using Agile modeling. The agile framework reduces wasted resources involving stakeholders’ collaborative work at the web application modeling and documentation stage. Several UML profile visual constructs are implemented by implementing the AC-MDD Framework to attain this objective. This new visual construct, the AC-MDD Method implemented in a case study, highlights the step-by-step transformation process of web application development with the final result at the modeling stage. The result of the product is an agile model using the AC-MDD Framework. This project aims to provide solutions for monitoring and evaluating learning in tertiary institutions.

Acknowledgments

This project is funded by the Faculty of Engineering at Universitas Negeri Surabaya. Additionally, we would like to thank the Dean of the Faculty for providing laboratory space for this investigation. Thanks to DAPT-EQUITY Program, Lembaga Pengelola Dana Pendidikan (LPDP), Ministry of Finance, Indonesia for supporting this publication.

References

2. A. Budiarto, et al., Indonesian Association for Pattern Recognition International Conference (2018)