# REVIEW ON THE EFFECTIVENESS OF AGILE UNIFIED PROCESS IN SOFTWARE DEVELOPMENT WITH VAGUE SYSTEM REQUIREMENTS

Lisana
Universitas Surabaya (UBAYA), Raya Kalirungkut, Surabaya, Indonesia
E-Mail: lisana@ubaya.ac.id

## ABSTRACT

Agile Unified Process (AUP) has been known as a suitable methodology for small-to-medium software development projects. This methodology focuses on the rapid iterations, small and frequent releases, capable of handling changing requirements from user, and involving user in the software development process. However, little is known that AUP can be effectively used for vague and incomplete system requirements. This study reveals how AUP is the most suitable software development methodology for such system requirements. An owner of a local gold jewelry store wishes to develop a new computerized system, but he does not know the system requirements for the software to be built. The purpose of the software is to help the owner in monitoring and controlling the main business processes in the jewelry store. In the beginning of the software development, it is very hard for the owner to mention what he really needs. However, since the owner is actively involved during the software development, he can slowly refine the system requirements with the help from the development team. AUP Phases can be accurately followed and proved to be very useful and suitable for the vague system requirements. Rapid iterations, small and frequent releases of the software modules lead to the completion of the software project on time. The resulting software is successfully tested with live transactions.

**Keywords:** agile unified process (AUP), software development methodology, RUP.

## INTRODUCTION

Software development methodology is a framework that is used to structure, plan, and control the process of developing a computer-based information system. This is the most important element in the software development. It depicts the necessary phases in software development from preliminary development, analysis, to post-development software testing and evaluation.

Software development methodologies like Waterfall and Rational Unified Process (RUP) are called traditional software development methodologies and can be classified into the heavyweight methodologies [1]. These two methodologies are based on sequential series of steps from requirements specifications, design, implementation, testing and deployment. Traditional software development methodologies should define and document a stable set of user's requirements at the beginning of a project. The success of a project depends on knowing all requirements before the development begin. It means that changes during the development process can cause problems [2].

Current software development is expanding and becoming more complex. Somehow, users can not provide all requirements in sufficient details for implementation to occur at the beginning of a project. Users may also have some problems in deciding the features to be included in the project. Changing requirements from the user is making it even more difficult. In the other side, many software companies tend to produce valuable software in shortest time period with minimal costs, and within unstable, changing environments. Traditional software development methodology cannot handle this anymore.

Thus, new software development methodologies are introduced, as agile methodologies, mainly to solve those problems. Agile methodologies are based on the idea of incremental and iterative development, in which the phases within the development life cycle are revisited over and over again [2]. Those iterations improve software by using customer feedback to converge on solutions. Agile methodologies prefer software development over documentation. Their philosophy is to deliver many working versions of the software in short iterations, then update the software according to customers' feedback. This will overcome the problems mentioned earlier, by welcoming changes, satisfying user requirements, faster development, and at the end, users will get the software they really need.

## AGILE UNIFIED PROCESS METHODOLOGY

One of Agile methodologies is Agile Unified Process (AUP). This methodology is a combination between the Rational Unified Process (RUP) and Agile Method (AM) [3]. AUP is also a simplified version of the Rational Unified Process (RUP) [4]. The AUP's Model discipline is a combination of the RUP's Business Modeling, Requirements, and Analysis and Design disciplines. Moreover, the RUP's Configuration and Change Management discipline is changed to the AUP's Configuration Management discipline. Figure-1 shows the lifecycle of the AUP.

AUP consists of four major phases and seven disciplines [4]. The AUP's phases are:

**1. Inception**

The objective is to identify the initial scope of the project, a potential architecture, and to obtain initial project funding and stakeholder acceptance.

**2. Elaboration**

The objective is to define the architecture of the system.

**3. Construction**

The objective is to construct the system that meets the stakeholders' needs.

**4. Transition.**

The objective is to validate and integrate the system with the using environment

All AUP's disciplines are performed in an iterative manner, defining the activities which development team members perform to build, validate, and deliver working software which meets the needs of users. The AUP's disciplines are:

**1. Model**

The objective is to understand the bussines organization, to define the problem and what the user needs, to identify the best solution.

**2. Implementation**

The objective is to transform models into executable code and to perform a basic level of testing, particularly unit testing.

**3. Test**

The objective is to find defects, to validate the system works as designed and meet the user's requirements.

**4. Deployment**

The objective is to integrate the system into the using organization

**5. Configuration Management**

The objective is to manage access to project artifacts. This includes tracking artifact versions over time and then controlling and managing changes to them.

**6. Project Management**

The objective is to direct the activities that takes place on the project. This includes managing risks, directing people (assigning tasks, tracking progress, etc.), and coordinating with people and systems outside the scope of the project to be sure that it is delivered on time and within budget.

**7. Environment**

The objective is to supports the development process with processes, guidance, and tools.

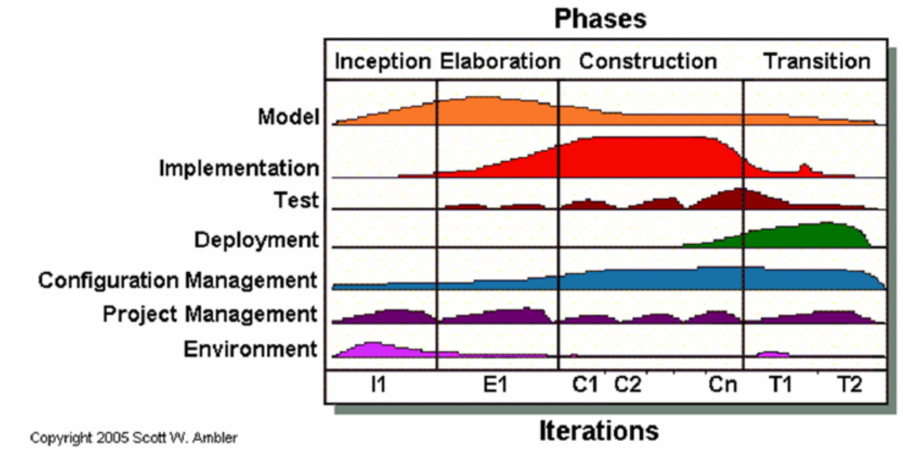


**Figure-1.** The lifecycle of the Agile Unified Process, source Ambler, 2005.

Ambler created AUP based on six principles [4]. First, most people are not willing to read detailed documentation, but they will need guidance and training from time to time. Second, the software project should be described simply in a few pages, not thousands. Third, AUP conforms to the value and principles of the Agile Alliance. Fourth, the software project must focus on high-value features. Fifth, project teams can use any tools freely which are best suited for the given job. The last one, the AUP product is easily tailored via any common HTML editing tool.

## LITERATURE REVIEW

One of the largest banks in Greece had successfully implemented AUP as its development methodology in a small to medium-scale project called the Integrated Desktop (ID) [5]. The purpose of the ID project was to host private-banking applications that can be accessed via a single sign-on. AUP was adopted to produce quick-win user results. The development team finally concluded that AUP provided a flexible and reasonably agile methodology. Moreover, the team also found out that to succeed in applying AUP, the organisation's culture and management must be receptive to both RUP and Agile Methods.

INT, a professional services and consulting firm located at Houston - Texas, always adopt AUP methodology for all software projects received from their clients. They had proved that implementing AUP can effectively reduce cost and risk [6]. The frequent released software will increase feedback from client so that the potential risks could be mitigated as early as possible. Moreover, the iterative development cycle provided high project visibility and control and also allowing customer to give some feedback quickly to changing system requirements.

Both AUP's implementations in software projects above have shown that AUP is suitable for small to medium IT project by utilizing its iterative software development cycle and frequent product releases. However, there is little evidence that AUP can cope with vaque and incomplete initial system requirements. There is a need to prove that AUP with its frequent product releases can reduce costs in a software development with frequent changes in system requirements.

## THE IMPLEMENTATION OF AGILE UNIFIED PROCESS

In this paper, Agile Unified Process (AUP) methodology was implemented in the development of a management information system at a gold jewelry store. The two main reasons why AUP was adopted. First, the new system was categorized into a small to medium-scale project. Second, the user (owner), with no IT knowledge, was unable to state clearly at the beginning about what they needs. There were some vaque requirements of the new system. Thus there would be high probability that the owner would keep changing the requirements during the project lifecyle.

The main objective of the new system was to help the owner of a gold jewelry store to control and monitor his main business processes, such as selling items to customer, purchasing items from supplier, and returning items from customer. All processes were done manually (paper-based). This caused slow in performance and several problems. i.e.

- The process of Adding New Items took a lot of time and inefficient. The owner had to weight the new items one by one, wrote the details of each new item in a book, and manually created the item-tag for each item.
- The process of creating the sales invoice was time-consuming. The sales staff had to write the item details in the sales invoice. Moreover, they often made mistakes in calculating the sales invoice total to be paid by the customer
- The process of summarizing all sales transactions took a lot of time and sometimes inaccurate. At the end of the day, the owner had to summarize all sales transactions occured during that day. This process took a lot of time from tracking those records one by one. Not to mention if the owner found a mistake in the calculation.
- The process of calculation total profit or loss from all sales transactions during period of time took a lot of time and inaccurate.
- The process of calculation total purchase item from supplier during period of time took a lot of time and inaccurate.
- The process of calculation total sales handled by each staff during period of time took a lot of time and inaccurate.

### Inception Phase

During this phase, the team had to set up a meeting with the owner to define the scope of the new system. The meetings were held twice.

The initial requirements were gathered and it was not necessary to state and define all the requirements at that time. As the results, the team defined the scope of the new system as follows:

- There are three type of users that can access the new system: owner, manager, and staff.
- There is a feature that can record and maintain all suppliers and staffs data.
- Record data of each new items bought from supplier and automatically generate a barcode that can be printed as the item's tag.
- Handle the process of item sales. The new system should be able to get the data from barcode scanner.
- There are some features of creating some reports that can be used to control and monitor the current system such as: profit and loss report, total purchase report, total sales report, and report of all items in the store (assets) grouped by type of jewelry (such as necklace, ring, earing etc).

From those demanding requirements, the team created the initial use case diagram. Figure-2 shows the initial use case diagram for the new system. Next step, the team tried to estimate the cost, plan the schedule and manage the risks.

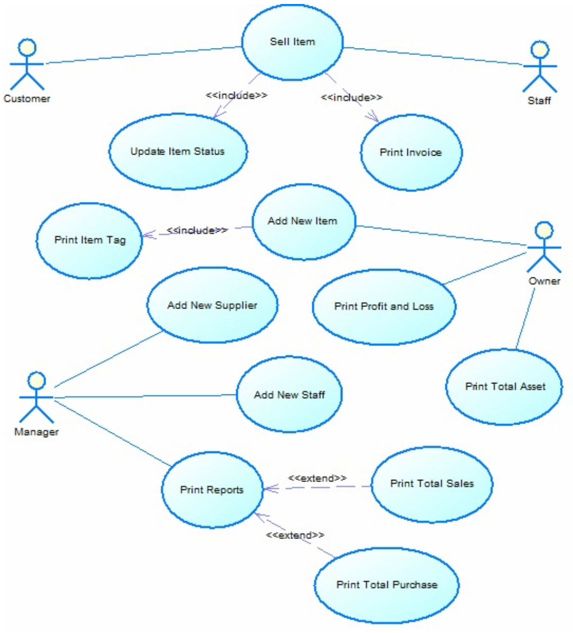


**Figure-2.** Initial use case diagram.

During this phase, the team found some difficulties as follows:

- It was quite hard to set up a meeting with the owner. This was because the owner is a successful businessman and he travels a lot to do some business.
- The team especially the manager should be able to maintain the owner commitment throughout all phases of software development.
- The team should prepare as early as possible the current human resources so they can adapt to the new innovative system.

### Elaboration Phase

The objective of this phase was to make sure that the new system could fullfil all the user's requirements. In this phase, the team tried to assign estimates and created a development plan. The team's system analyst gathered all users' requirements. All features were examined and rough time estimation was generated. The owner then used the estimate to assign a priority to each features. It is not necessary to assign a priority to all features at that time. The owner also collaborated with the team to create a good development plan which describes how the deliverables will be.

Then, the team developed a working prototype which demontrated the architecture and user interface functionality for each features. The prototype was showed to the owner to get the feedback, comments, remarks, and suggestions on the user interface and business workflow and system functionality. This was used to produce the necessary results for second iteration. Next, the team made some changes. The team also conducted a user-acceptance-testing during which they received a second of minor correction requests. During this phase, the owner still could add some new requirements, modified or removed the existing requirements. The team together with the owner worked in developing the final version of documentation.

After looking at the demonstration of the working propotype, the owner requested two changes. First, the owner gave the formula for profit calculation. Second, the owner told the team to change format of the item code. Furthermore, after discussing with the team manager, the owner added some new features as follows:

- Record all prospective customer data.
- Integrate the new digital scale hardware with the proses of record data of each new items so the staff does not need to type the weight of the item manually.
- Create a report of sales transaction based on customer.
- Create a report of sales transaction based on staff.

The team then directly revised the existing initial use case. Figure-3 showed the modified use case diagram after second iteration.

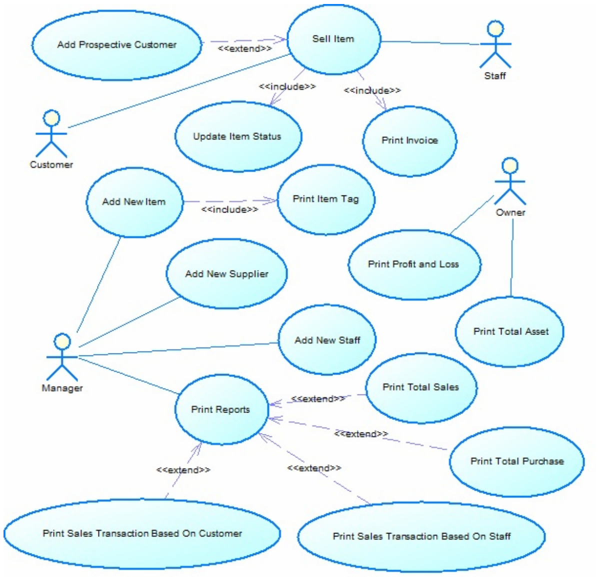


**Figure-3.** Modified use case diagram.

### Construction Phase

Construction was the largest phase. In this phase, the team developed the new system in small increments called iterations. An interation consists a series of steps performed over a short. The steps included picking up the features to be implemented, refining bugs, designing the features, implementing the design (including testing and creating documentation), deploying an executable release of the software to obtain owner's feedback. The successful completion of each iterations ended with a user acceptance test.

During this phase, the team created the data model, a class diagram, as seen in Figure-4. The team had three four-week iterations, implementing between four to five use-cases at a time. The application was developed using Microsoft Visual Basic and MySQL.

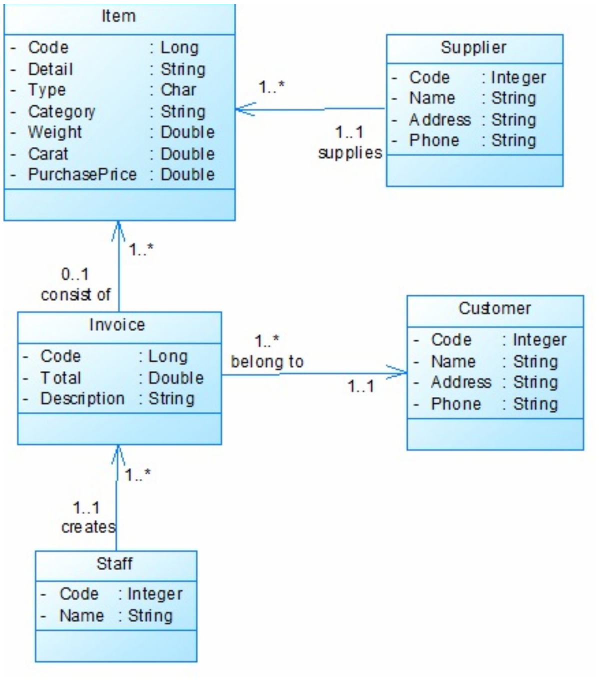


**Figure-4.** Class diagram.

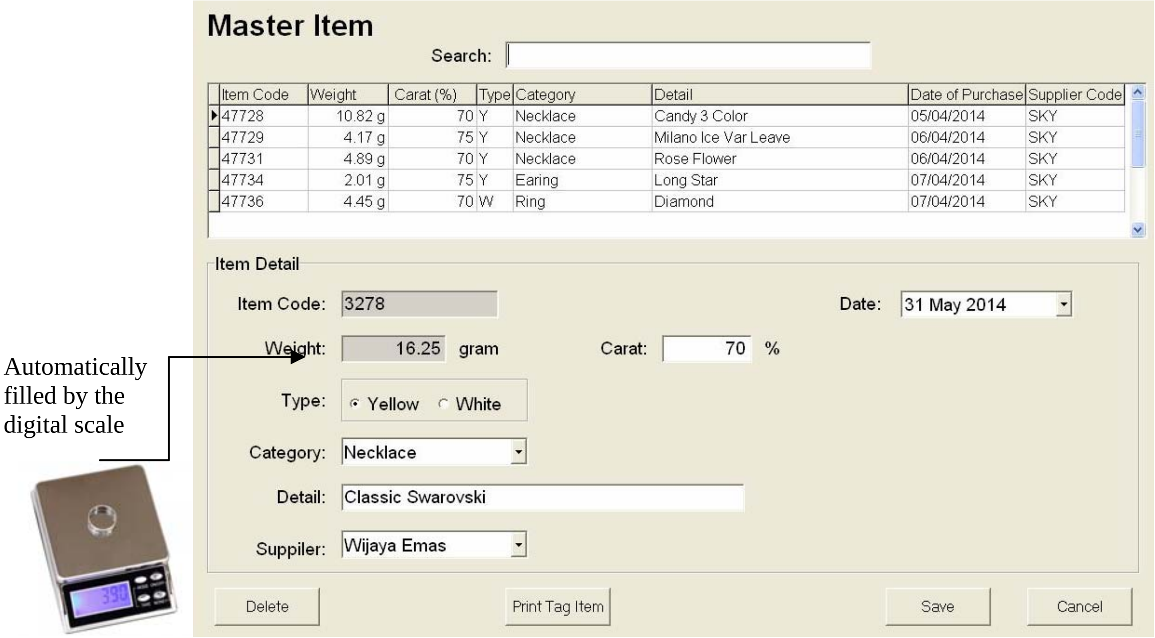
In the first iteration (first executable), the team implemented 40 percent of all features in the new system, including the most difficult ones. The interation continued with the second. By adding more features, the new system will become expandable and getting perfect. This phase finished in 60 working days. Figure-5 shows a screen of add new item as a part of the result of the new system.

**Transition Phase**

Transition phase includes system testing, user testing, system rework and system deployment. Although some testing had been done during the earlier phases (elaboration and construction phases), the real users (owner, manager, and staffs) still had to check whether the new system can run properly in the real environment. In this phase, the team gradually deployed the new system. The owner was encouraged to conduct the acceptance test to make sure that the new system had met all the users' requirements. Otherwise the owner could send some feedback so the team could make further refinements.

In this phase, firstly, the team had to set-up the laptop with all supported devices, such as barcode printer, barcode scanner, and digital scale hardware. After the hardware had been properly set up, the team then deployed the new system and conducted testing to make sure that all features can work perfectly. The team also conducted some training to the manager and staffs. This phase ended when the real users (owner, manager and staffs) were satisfied with the new system. Furthermore, all supported documents, including user manual, were also handed on to the owner.

The transition phase was relatively standard, with the only problem of converting the manual transaction data into the new database.



Master Item

Search:

| Item Code | Weight | Carat (%) | Type | Category | Detail | Date of Purchase | Supplier Code |
|---|---|---|---|---|---|---|---|
| 47728 | 10.82 g | 70 | Y | Necklace | Candy 3 Color | 05/04/2014 | SKY |
| 47729 | 4.17 g | 75 | Y | Necklace | Milano Ice Var Leave | 06/04/2014 | SKY |
| 47731 | 4.89 g | 70 | Y | Necklace | Rose Flower | 06/04/2014 | SKY |
| 47734 | 2.01 g | 75 | Y | Earing | Long Star | 07/04/2014 | SKY |
| 47736 | 4.45 g | 70 | W | Ring | Diamond | 07/04/2014 | SKY |

Item Detail

Item Code: 3278

Date: 31 May 2014

Weight: 16.25 gram

Carat: 70 %

Type: ● Yellow ○ White

Category: Necklace

Detail: Classic Swarovski

Suppiler: Wijaya Emas

Automatically filled by the digital scale

**Figure-5.** Screen of add new item.

## THE HIGHLIGHTED RESULTS

For the team, this project was the first experience in developing a system using Agile Unified Process (AUP) methodology. In the earlier phase of applying the AUP methodology, the team was a little bit confused because they work without complete documentation like they were used to in the waterfall methodology. However, a few days later they could adapt and felt comfortable with.

The management information system at the gold jewelry store was considered to be a small to medium-scale project with initial twelve use cases. However, during the elaboration phase, the owner changed the some requirements and added four new features. Based on that feedback the team could easily modify the existing use case diagram by adding three new use cases and directly built the first version of the executable software.

In the construction phase, the team produced three software iterations. At the first iteration, team implemented five use cases. This first executable software could be considered as the most difficult ones including Add New Item Process and Selling Item Process. After showing the first software, the owner had discussion with the team and gave some feedback. The team then directly made some revision without any difficulty and developed the second version. During this phase, the team still could handle the requirement change effectively.

In the beginning, owner did not know exactly the real system requirements, but, with subsequent iteration, the owner slowly understood the software requirements to be developed. The finished software

The result had proved that AUP methodology is a simple, easy to understand, and easy to implement approach in developing a management information system at a gold jewelry store. The keys of success are the effective team work, frequent communication between team and the owner, and deep participation of the users (owner, manager and staff). These factors can accelerate creating the good model and finally high quality software can be produced. To mitigate risk of failure, software was deployed gradually. Moreover, by doing both system testing and user testing can detect error rates earlier and improve the quality of system design for better maintainability. This will reduce the cost of detecting and correcting errors. AUP methodology provides the owner with high system visibility and control, allows the owner to give constant feedback to make sure that the implementation of all features are inline with the owner's objectives, and allow the owner to change the initial requirements.

These results emphasize the advantages and strength of AUP as the methodology of choice for small-to-medium IT Projects, even with vague and incomplete system requirements.

## CONCLUSIONS

The result showed that Agile Unified Process (AUP) methodology was effective in developing software for gold jewelry store, which the initial requirements were vague and incomplete. AUP break the system into small iterations in which requirements, design, development, and testing occur continuously. Owner was actively involved from the beginning of the software development. Thus, the team could produce very accurate software and all features can represent the owner's objectives. At the end, the users (owner, manager and all staffs) are happy and very satisfied with the newly implemented software.

## REFERENCES

[1] Khan A. I., Qurashi R. J., Khan U. A. 2011. A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies. International Journal of Computer Science Issues (IJCSI). Vol. 8, Issue 4, No 2. pp 441-450.

[2] Leau Y. B., Loo W. K., Tham W. Y. and Tan S. F. 2012. Software Development Life Cycle AGILE vs Traditional Approaches. International Conference on Information and Network Technology (ICINT). IPCSIT. 37: 162-167.

[3] C. Edeki. 2013. Agile Unified Process. International Journal of Computer Science and Mobile Applications. 1(3): 13-17.

[4] S. Ambler. 2005. The Agile Unified Process (AUP), Retrieved from http://www.ambysoft.com/unifiedprocess/agileUP.html.

[5] Christou I. T., Ponis, S. T., and Palaiologou, E. 2010. Using the Agile Unified Process in Banking. IEEE Computer Society. pp. 72-79.

[6] INT Professional Services. Using Agile Unified Process to Reduce Cost and Risk. Retrieved from http://int.com/pdf/INT_Agile_Unified_Process.pdf.