INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND MOBILE APPLICATIONS - IJCSMA

Agile Unified Process

Charles Edeki
Ph.D, American Intercontinental University, Department of Information Technology,
160 Parkside Ave, 9M, Brooklyn NY, 11226

cedeki@faculty.aiuonline.edu

Abstract

The Agile Unified Process (AUP) is based on a cut-down version of the Rational Unified Process (RUP). AUP uses an agile approach which focuses on both the larger life-cycle and the iterations within each step to deliver incremental releases over time.

1. Overview of the Rational Unified Process

The Rational Unified Process (RUP) is a software engineering approach whose goal is to produce high-quality software that meets or exceeds the expectation of its users. RUP achieved this by providing a disciplined approach to the management of tasks and responsibilities within a software-development organization (Rational Unified Process, 2002).

The Rational Unified Process is a descendent of the Rational Objectory Process which was the product of a 1995 merger between Rational Software Corporation and Objectory AB. Objectory provided the concept of use cases, and Rational provided the iterative process model (Kruchten, 2004).

Objectory was developed by Dr. Ivar Jacobson, which centered on use-cases and object-oriented design methods. The Rational approach, which was developed by Rational staff, included Philippe Kruchten, Grady Booch, and Walker Royce, provided the iterative development approach which focused on software architecture (Jacobson, 1998). The combination of these two products created a process that covered the entire software-development life-cycle (Kruchten, 2004).

2. Agile Unified Process

The agile unified process is a hybrid modeling approach created by Scott Ambler when he combined the Rational Unified Process (RUP) to agile methods (AM) (Christou, Ponis & Palaiologou, 2010). Scott Ambler works for the IBM Methods group as the practice leader for agile development (IBM, n.d.).

By combining RUP to AM, Ambler created a solid process framework that can be applied to all sorts of software projects, large or small. Agile methods provided values, principles, and practices to AUP. The agile manifesto shows what these values and principles are.

The manifesto describes four value statements for agile development. These values include individuals and their actions, delivering working software, customer collaboration, and responding to change (Sutherland & et al., 2001).

The principles described in the manifesto include satisfying the customer through early and continuous software deliverables, welcoming change, developers and business collaborating throughout the project, building projects through motivated individuals, using the most effective means of conveying information like face to face conversations, attention to technical excellence, simplicity, using self-organizing teams, regular reflection and adjustments (Sutherland & et al., 2001).

When Ambler created the AUP, he centered the design around the following principles:

- Most people won't read detailed documentation. However, they will need guidance and training now and then.
- The project should be described simply in a few pages.
- The AUP conforms to the values and principles described by the Agile Alliance.
- The project must focus on delivering essential value rather than unnecessary features.
- Developers must be free to use tools best suited to the task at hand, rather than to comply with an edict.
- AUP is easily tailored via common HTML editing tools.

Source: (Ambler, 2005).

3. Applying the Agile Unified Process

The AUP is an iterative-incremental process consisting of workflows and phases. A direct comparison between RUP and AUP shows that AUP combines business modeling, requirements, and analysis and design into a single workflow called Model (The agile unified process, n.d.). Workflows progress through two weekly iterations which transition the process across the different phases (The agile unified process, n.d.).

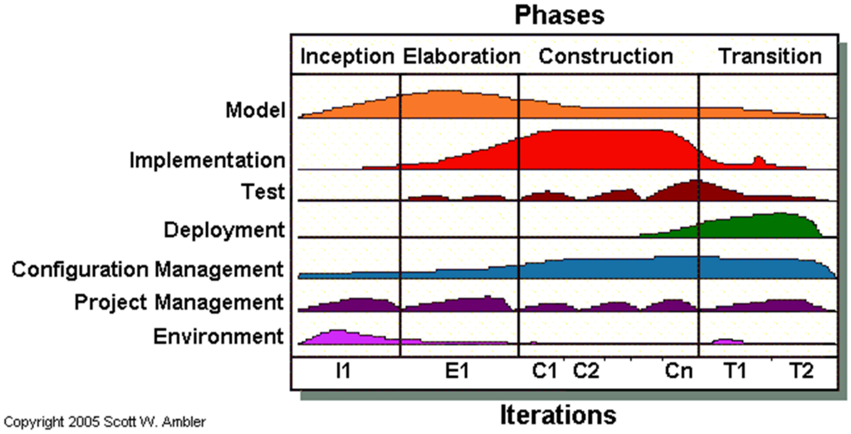

Figure 1: Agile Unified Process Phases, source (Ambler, 2005)

Model Workflow

The initial step in the AUP is the model workflow which begins with the Inception phase. Here project scope, risks, costs and schedule, and project feasibility are defined, including the preparation for the project environment (Ambler, 2002).

A major milestone in the Inception phase is the defined requirements without regards to its final technical implementation (Taylor, Medvidovic & Dashofy, 2010). According to Pappus, a fourth-century Greek mathematician, "in analysis, we start from what is required. We take it for granted, and we draw consequences from it, until we reach a point that we can use the starting point in synthesis." Nevertheless, knowing what is required to solve a problem needs a comprehensive understanding of the requirements (Taylor, Medvidovic & Dashofy, 2010).

Ambler (2005) explains that the requirement's phase includes identifying the stakeholders, understanding the user's problem, establishing a basis of estimation, and defining the user interface for the system. Although these activities occur during the Inception and Elaboration phases, they can continue through other phases to improve the unfolding design.

The deliverable from the requirement's phase is the business Use Case Model. This model consists of UML Use Case diagrams and essential use cases. The intent of these models is to describe the functions of the business in a technology-independent way (Ambler, 2002).

During the construction phase, user stories are implemented and iteratively reworked to reflect the expanding understanding of the problem domain as the project progresses (Ambler, 2002).

The final phase of the Model workflow is the Transition phase. Here the software is delivered to the user. However, even during this phase of the project, iterative improvements like modeling, implementation, testing, and management activities can still be applied (Ambler, 2002).

Implementation Workflow

A major part of the Implementation workflow occurs during the Construction phase where the model is transformed into executable code and tested thoroughly by unit tests (Ambler, 2005).

Test Workflow

During the Test Workflow phase, an objective evaluation of the developed application is performed to ensure that it conform to the quality requirements. These phases include unit-testing, usability testing, and user-acceptance testing (Ambler, 2002).

Deployment Workflow

The purpose of the Deployment Workflow is to ensure that the newly developed application is successfully deployed (Ambler, 2002).

Configuration Workflow

The Configuration Workflow ensures that the artifacts of the system are tracked and properly versioned (Ambler, 2002). For instance, if a YAML file is changed over time, that there is a proper configuration management process in place to prevent the wrong version from being used.

Project Management Workflow

The Project Management Workflow directs the activities that occur during the project. These activities include risk management, project management, and coordinating external resources to ensure that the project is delivered within scope, cost, and time (Ambler, 2005).

Environment Workflow

The Environment Workflow makes certain that the project occur in an environment suitable enough to ensure the success of the project. For instance, those proper processes are in place, that there are standards and guidelines available, and that the teams have the proper facilities, hardware, and software to complete the project (Ambler, 2005).

4. Commercial Example of Agile Unified Process

A large bank in Greece adopted AUP as its development methodology to implement an important Service Oriented Architecture (SOA) project. The project's objective was to provide a way to host private banking applications that can be accessed via a single sign-on (Christou, Ponis & Palaiologou, 2010).

The development team followed AUP's four phased approach throughout the software development life-cycle (Christou, Ponis & Palaiologou, 2010).

During the inception phase, use-cases were discovered and the project manager made project cost and schedule estimates for the project's later phases. Tasks scheduled soonest were detailed more accurately while future tasks were understood to be best effort estimates (Christou, Ponis & Palaiologou, 2010). Moreover, in addition to the time estimates, the project manager also collaborated with development, architecture, and operations to create the cost estimates which were submitted to the steering committee for approval (Christou, Ponis & Palaiologou, 2010).

Keeping in step with UAP tasks during the Inception phase, the development team also defined the risks facing the project. The plan was that earlier iterations would addressed the highest-priority risks first (Christou, Ponis & Palaiologou, 2010).

During the elaboration phase, the analysts gathered requirements while the development team created a mock-up showing the user interface as screenshots (Christou, Ponis & Palaiologou, 2010). The team held workshops to solicit comments, remarks, and suggestions for improvements based on the presented mock-up (Christou, Ponis & Palaiologou, 2010).

Phases prior to the construction phase identified the requirements which were used to create the architectural baseline for the new system. This became the bases for the construction phase. The construction phase was applied as two time-boxed iterations lasting 24 and 46 days respectively. Each iteration concluded with a deliverable that was presented for user acceptance testing (Christou, Ponis & Palaiologou, 2010). The construction phase was eventually concluded with a system demonstration to the steering committee (Christou, Ponis & Palaiologou, 2010).

During the Transition phase, the team focused on moving the system into production. Initial feedback after deployment resulted in two further iterations of improvements (Christou, Ponis & Palaiologou, 2010).

Before the final roll-out, extensive beta testing was performed to satisfy the bank to release the system to the rest of the user base (Christou, Ponis & Palaiologou, 2010).

The bank's development team concluded that AUP provided a flexible and reasonably agile methodology. However, they also found that the organization's culture and management needs to be receptive to both agile and the Unified methodology in order to succeed (Christou, Ponis & Palaiologou, 2010).

5. Conclusion

The Agile Unified process (AUP), which is based on agile methods and the Rational Unified Process, provides an iterative-incremental approach to software development. AUP consists of seven workflows, each of which has four phases.

The AUP workflow consists of Mode, Implementation, Test, Deployment, Configuration Management, Project Management, and Environment. The AUP phases consist of Inception, Elaboration, Construction, and Transition.

The initial project step is the Inception phase that begins with the Model workflow where the project scope, risks, costs and schedule, and feasibilities are defined.

The Construction phase is typically the largest phase and is mostly concerned with producing the executable code.

The Test workflow ensures that the developed application conform to quality requirements.

The Deployment workflow ensures that the developed system is properly deployed, while the Configuration workflow makes sure that the system's artifacts are properly tracked and versioned. The Project Management Workflow directs the activities that occur during the software-development life cycle, while Environment Workflow ensures the project team has all that it needs to be successful.

A commercial project that used AUP for its development methodology concluded that AUP provides a flexible and reasonably agile methodology. However, they also found that the organization's culture and management needs to be receptive to Agile methods in order to successful use AUP.

References

1. Ambler, S. (2002). Agile modeling: Effective practices for extreme programming and the unified process. New York: John Wiley & Sons, Inc.
2. Ambler, S. (2005). The agile unified process (aup). Retrieved from http://www.ambysoft.com/unifiedprocess/agileUP.html
3. Christou, I. T., Ponis, S. T., & Palaiologou, E. (2010, April 7). Using the agile unified process in banking. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5232801
4. IBM. (n.d.). Though leaders. Retrieved from http://www-01.ibm.com/software/rational/bios/ambler.html
5. Jacobson, S. (1998, October). *The rational objectory process - a uml-based software engineering process*. Eurospi98, Gothenburg, Sweden. Retrieved from http://www.iscn.com/select_newspaper/object/rational.html
6. Kruchten, P. (2004). *The rational unified process: An introduction*. Boston: Pearson Education, Inc.
7. Rational Unified Process. (2002, May). *Rational unified process*. Retrieved from http://sce.uhcl.edu/helm/rationalunifiedprocess/process/ovu_proc.htm
8. Sutherland, J., et al., J. (2001, February 2). Manifesto for agile software development. Retrieved from http://agilemanifesto.org/

9. Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2010). Software architecture foundations, theory, and practice. Danvers: John Wiley & Sons, Inc.
10. The agile unified process. (n.d.). The agile unified process. Retrieved from http://www.cs.sjsu.edu/~pearce/modules/lectures/se/aup.htm