# A Discussion on the Unified Process Software Development Methodology

Elizabeth Di Bene
*University of Maryland University College*

**Abstract**
No abstract provided.

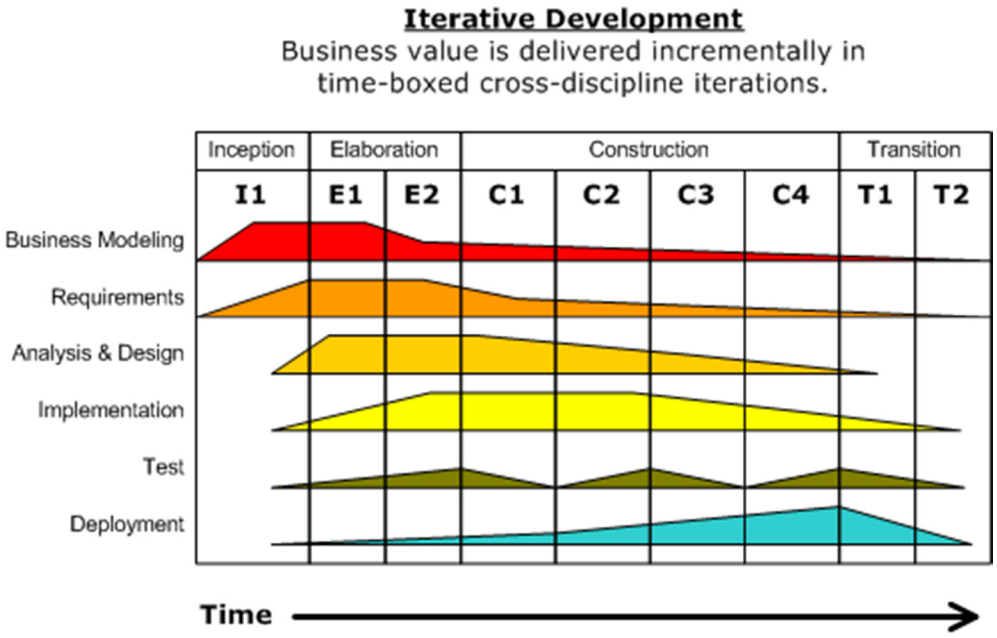
## Introduction

A software development methodology refers to the framework that is used to structure, plan, and control the process of developing software. The Agile Methodology (AM) in software development is a collection of software methods that are iterative, evolutionary and adaptive. The AM is not one specific development method but a collection of methodologies sharing Agile principles focused on collaboration, face to face communication, self-organizing teams, continuous integration, incremental design and rapid and flexible responses to changes. This short paper will explore one AM, the Unified Process (UP) and discuss its composition and application.

## Unified Process

The UP is centered in architecture, driven by use-cases, and characterized by iterative and incremental development practices that leverage the unified modeling language (UML). It was first described by Ivar Jacobson, Grady Booch and James Rumbaugh in a 1999 book titled *The Unified Software Development Process.* The Rational Unified Process (RUP) is a refinement of the UP and is a trademark software engineering process from IBM that uses the practices of UP. RUP provides a disciplined approach to assigning tasks and responsibilities within a development organization (IBM, 2001).

The RUP defines nine disciplines: Business Modeling, Requirements, Analysis and Design, Implementation, Test, Deployment, Configuration and Change Management, Project Management, and Environment. The UP allows a project to refine the implementation of RUP according to project specific requirements. Agile refinement of UP such as the OpenUP, part of the Eclipse Process Framework simplifies RUP by reducing the number of disciplines (Unified Process, 2014).




(Unified Process, 2014)

The UP is customizable for specific organizations through four phases: Inception, Elaboration, Construction and Transition. The Elaboration, Construction and Transition phases are divided into a series of time boxed iterations and the Inception phase may be divided into iterations for a large project (Unified Process, 2014).

The UP is use-case driven in which use-cases are used to define functional requirements and what will happen through the iterations. The iterations take a set of use-cases from requirements through implementation, test and deployment (Unified Process, 2014). The UP is architecture centric supporting multiple architecture models and views which further enforces the aspect that UP is tailorable for organizations.

Two significant benefits of the UP include the fact that is tailorable. Another benefit is that UP is risk focused. Risks are addressed in order of criticality, early in the project life cycle. The most critical risks are addressed early in the project life cycle because the deliverables of each iteration must be selected in order to ensure that the greatest risks are addressed first (Unified Process, 2014). The tailoring of UP can be complex which can require the use of a large number of resources, i.e., human, financial. This can be a disadvantage because tailoring is also difficult because it must account for many factors in order to avoid the appearance of inconsistencies due to the reduction of activities (Geambaşu, 2011).

The four phases of the UP divide a project's development lifecycle. The Inception phase is the smallest and should be short. An Inception phase that was long could indicate excessive up-front planning which is contrary to the foundation of UP. UP is iterative and incremental characterized by time-boxed iterations. The Inception phase ends by the Lifecycle Objective Milestone. In the Inception phase, the project develops an approximate vision of the system, makes the business case, defines the scope, and produces a rough estimate for cost and schedule (Unified Process, 2014).

Next is the Elaboration phase in which the deliverable is a plan including cost and schedule estimates for the next phase, the Constructions phase. The Elaboration phase addresses known risk factors and captures a majority of system requirements (Ambler, 1999). The

Elaboration phase creates use-case diagrams, conceptual diagrams and architectural diagrams, laying out the foundation for the system.

The Construction phase is the largest phase which focuses on iterations which result in executable releases of the software (Unified Process, 2014). Full text use-cases become the start of each new iteration while system features are implemented in short time boxed iterations built on the foundation defined in the Elaboration phase.

The Transition phase is the last phase of UP. The system is deployed to target users; feedback is received on iteration releases which can then be rolled into subsequent iterations. User training also occurs in the Transition phase when the system is ready.

The Basic Unified Process (BUP) is variation of RUP. BUP preserves the essential characteristics of RUP, namely iterative development, use-cases during development, risk management and architecture centric approach (Balduino).

BUP is organized into disciplines that are requirements, architecture, development, test, project management and change management. These disciplines allow a project to select only the desired content when creating a project.

BUP is organized in two different dimensions which are **method and process**. The method dimension defines method elements such as roles, tasks, artifacts and guidance. The process dimension defines how method elements are applied.

**The Method dimension**

The roles of BUP are the Project Manager, analyst, architect, developer, tester and "any role". The "any role" represents anyone in the team who can perform tasks such as submitting and performing changes, participating in meetings, etc.

- The Project Manager plans and manages the BUP effort.
- The Analyst gathers requirements and documentation as necessary. In some Agile practices, it may be more suitable for the Analyst to also be the customer representative.
- The Architect makes software architecture decisions and additional technical decisions.
- The Developer creates a solution or part of a solution by conducting the design practices. The developer conducts implementation, unit tests and integration of components.
- The Tester conducts the appropriate testing.

Some of the tasks defined in RUP have been transformed into guidelines for reference to simpler tasks. Other RUP tasks were transformed into steps and included inside other major tasks being performed by the same role at the same point in time. The roles and associated activity in a BUP project are at the discretion of the project.

BUP uses a reduced number of artifacts compared to RUP. Only six of the artifacts have (informal) templates with the remaining artifacts containing guidelines which explain how to informally represent them (Balduino). BUP allows the guidelines to be captured in existing artifacts which allows projects to select the appropriate level of effort for artifacts.

**The Process dimension**

The Process dimension is where the method elements are applied in a behavioral sense, where, deriving the same set of requirements, different lifecycles can be used for a variety of different projects. BUP creates reusable method content separate from its application through

processes. These methods that are organized into reusable pieces are called capability patterns which allow a consistent development approach to common problems. Patterns are activities organizing tasks (from the method content), grouping them in a sequence that makes sense for the particular area where that pattern is applied (Balduino).

Method content describes how specific development goals are achieved through step-by-step explanations. Processes then take these method elements and specify them into types of projects by semi-ordering them in sequences. Patterns may be small and focused where they are considered the basic building blocks to create larger patterns or identify delivery processes. A requirement specified as a use-case is assigned to developers and work progress is tracked based on goals. Context is specified when a requirement is assigned for development. It is the developer's responsibility is to create a design and implementation and to write and run a unit test. Patterns may occur as many times as there are requirements to be developed in a given iteration.

To create larger patterns, basic building blocks are combined. The combination of building blocks can be used to assemble template patterns and address objectives within the pattern.

The delivery process in BUP occurs through four phases through the combining of the iteration template patterns as many times as necessary. Projects with different needs may need to assemble iteration template patterns differently. Either each individual iteration template pattern or the whole delivery process can be used to instantiate a project plan, or parts of a project plan because the capability patterns and delivery process contain Work Breakdown Structures (WBS) with tasks organized under activities (Balduino).

BUP is designed for small teams and small projects with a focus on small roles. BUP is a flexible approach in that it allows organizations to tailor roles, tasks and artifacts. BUP would not be suitable for teams or projects larger than 3-6 people and 3-6 months of development efforts.

## Conclusion

Projects can extend BUP to add variations on how to capture design decisions. UP allows Agile practices to refine and streamline workflows, artifacts and disciplines. Regardless of the variation of UP, each variation is a lightweight version of IBM's trademark RUP. Each system development framework has its own strengths and weaknesses because not one methodology is suitable for all projects. Selecting the proper framework involves evaluating the technical, organizational, project and team factors. To be successful, most projects will need to have a mix of different methodologies. A software engineer and software project lead must be flexible and recognize that different technologies require different techniques; categories, methods, projects, teams, technologies are all different and can vary. Regardless of which methodology is employed for a software project, the software engineer must prioritize and define goals vice artifacts. It is best to have a repertoire of software frameworks so that the software engineer can tailor a mix of any of them to a software project.

## References

Ambler, S. W. (1999). *The Unified Process Elaboration Phase : Best Practices for Implementing the UP*. Lawrence, Kan: CRC Press LLC.

Balduino, R. (n.d.). *Basic Unified Process: a Process for Small and Agile Projects.* Retrieved 10 16, 2014, from IBM Rational and Eclipse: https://www.eclipse.org/proposals/beacon/Basic%20Unified%20Process.pdf

Geambaşu, C. J. (2011). Influence Factors for the Choice of a Software Development Methodology. *Accounting & Management Information Systems / Contabilitate Si Informatica De Gestiune, 10*(4), 479-494. Retrieved from http://ezproxy.umuc.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=70104877&site=eds-live&scope=site

IBM. (2001, 11). *Rational Unified Process Best Practices for Software Development Teams.* Retrieved 10 16, 2014, from IBM.com: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Pelosi, D. (2014). *WK05, UP*. Retrieved from SWEN 603 9080 Modern Software Methodologies Course Content, UMUC: https://learn.umuc.edu/d2/le/content34176/Home

(2014). Unified Process. In *Modern Software Methodologies for Software Engineering "MSM" SWEN 603 WikiBook* (pp. 77-81). PDF Wikibook. Retrieved from https://learn.umuc.edu/d2l/le/content/34176/Home