



The Rational Unified Process

White Paper

Abstract

This paper describes the uses and benefits of the Ration Unified Process

Tometa creates custom software for you

Tometa Software designs and develops robust software solutions for virtually all industries including in-house (vertical market) and retail software, some of which is on the shelves at your local software store. We focus our unique combination of creative, technical, and problem-solving skills on meeting our client's objectives. Because of our clarity of purpose, commitment to process, and broad professional skill sets, we are able to provide our clients with world-class solutions that are functionally superior and fully aligned with our client's strategic focus.

Balancing development speed, quality and cost is what we are all about. Tometa combines agile development practices with fixed pricing, so you know what the cost, end product, and delivery time table look like—up front. If we underestimate the effort, we complete the overrun on our dime. Simple as that. That's why large enterprise firms like Alcoa and NASDAQ choose Tometa.

Tometa's agile development expertise and low-overhead US location keep our prices comparable to offshore vendors – without offshore challenges. Using a fixed pricing model, we provide upfront visibility into a project's ultimate costs, end product and delivery schedule. Our clients like knowing that we have “skin in the game” – a fixed price that aligns our goals with yours, incenting us to get the job done right and fast.

Lastly, as a Microsoft Certified Gold Partner, Tometa Software, can customize its products or create custom web, client/server, and traditional applications. With programming experience in C#, C++, Visual Basic, CGI, HTML, RPG, Delphi, Java and many others; Tometa Software is uniquely positioned to meet your needs as a development firm.

[Check us out today](#)

Introduction

Enterprise Systems has become a “must have” asset for companies to manage their daily business, reduce their costs and increase their operation income. Many organizations increased their profit by double digits in a very short amount of time because they invested in systems that helped them achieve their business strategy. Others lost millions or even billions in applications that were never used. In a cost-aware economy, the last thing you want to do is invest money in something that will not generate profit. At the same time, you can’t afford to not take the risk.

How do you maximize your return on investment? How do you make sure that your money is spent on something that your business will benefit from?

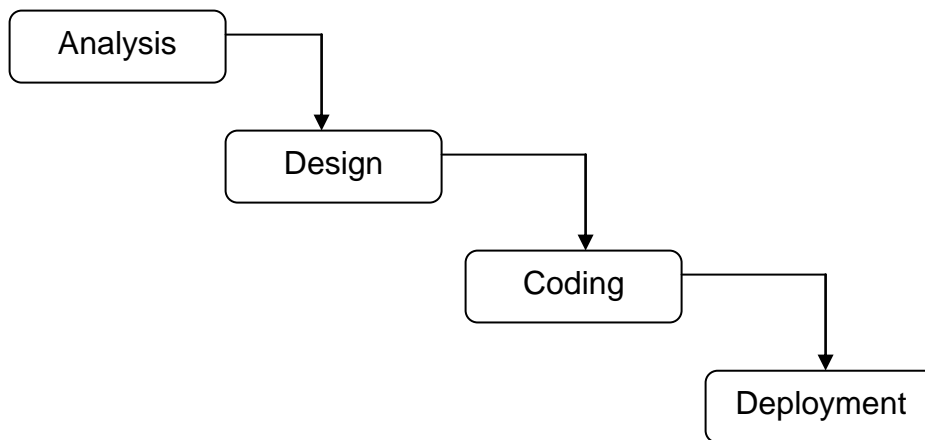
How do you manage the progress of your IT project without getting yourself bogged into details? Most importantly, how do you prevent things from going wrong?

To answer all of these questions, we will introduce the Rational Unified Process (RUP), a product of the Rational Corporation. I will shed some light on how this process could help your company build the right system for your business.

Sequential Waterfall Lifecycle

The old alternative to developing software was sequential, similar to the engineering approach used in the construction of buildings and bridges. The following diagram illustrates the different phases in this process:

Figure 1: Sequential Waterfall Lifecycle



For some time, that was the approach adopted by most managers, architects and authors. This is a very satisfactory process where requirements are well designed and not expected to change, for example, automating a well-proven manual system.

The weaknesses of this approach arise as the time scale lengthens. The complexity becomes high; many speculative decisions are made and accumulated without validation with the end-users or the management. During the design and coding phase the end-user is

almost out of the picture. The end-user usually does not come into the picture until the final product is delivered.

The risk of ending up with the wrong system is very high. There are many things that can go wrong in any phase: inaccurate or incomplete requirements, wrong architecture, lack of skills, etc...

By the nature of this approach and the complexity of IT projects, the managers base their decisions on some expectations and promises. By the time the final product is delivered, it becomes either impossible or very expensive to correct things that went wrong from the beginning. The management can either spend more money to fix the product or throw away the whole system.

A two year study reported in the MIT Sloan Management Review of successful software projects identified four factors for success: iterative development, rather than a Waterfall Lifecycle, was a first on the list.

Because of these limitations, the Waterfall approach is increasingly being replaced by iterative processes.

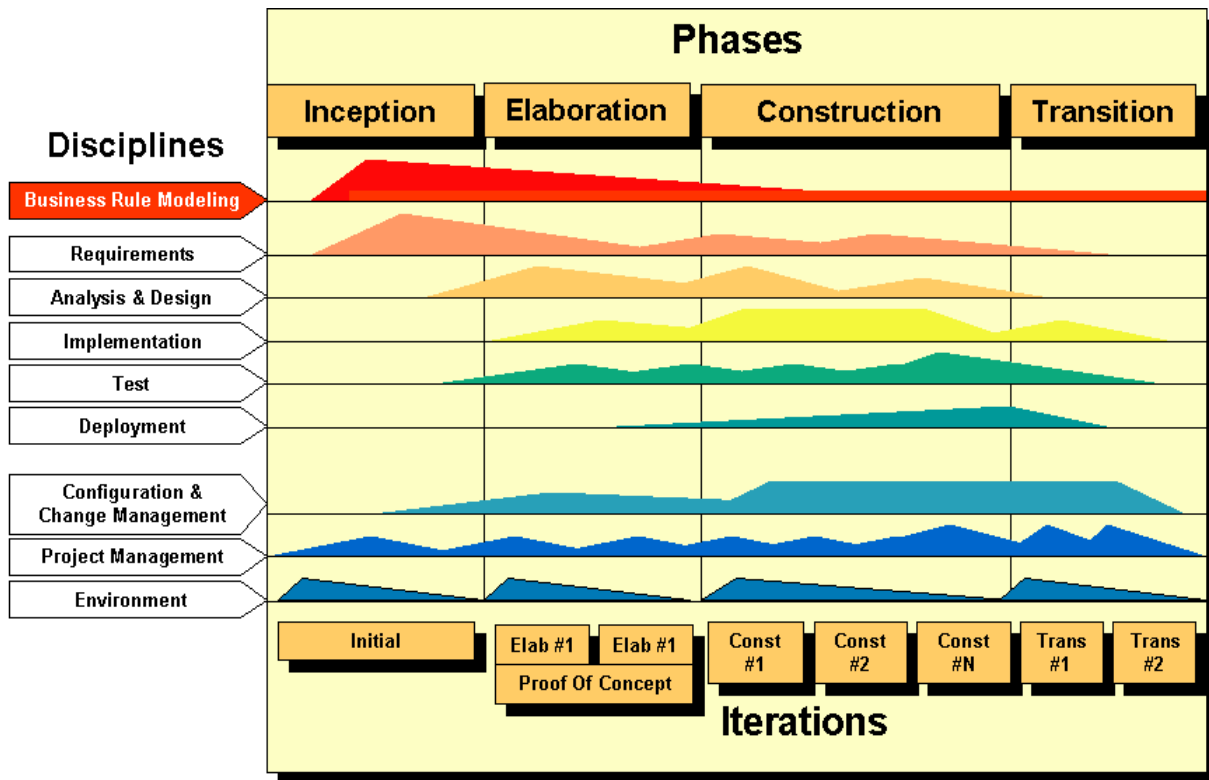
Rational Unified Process

The Rational Unified Process is an incremental process. The overall project is broken down into phases and iterations. Iterations are risk driven and oriented toward mitigating those risks. Each one should deliver executable software that is demonstrable and testable against the project requirements and use cases.

The team decides on the priorities and starts with the most important functionalities... The feedback from this can then be used to refine the requirements, and identify problems before moving to the next step.

The process is repeated, until eventually all requirements are implemented and the product is completed.

Figure 2 shows different phases of this process:



Inception

The goal of the inception phase is to achieve a consensus among all stakeholders on the objectives of the project. They may vary from organization to organization but will probably include: senior management, project managers, business representatives, architects, developers, testers, quality assurance, operations, support staff, users and others. It is important that the stakeholders define the success criteria, document and prioritize the organization needs with regards to software process.

With the vision document and success criteria, a business case should be developed.

The project managers document the economic value, the cost and the expected profit or saving that the system will generate.

Consensus by stakeholders on the economic value and market need is essential to gain support, funding, and commitment from the senior management.

The RUP recommends the following steps to develop the business case for the project:

- Describe the product and why it's needed
- Define the business this product will support
- Define the object at high level
- Develop a financial forecast including projected revenues and costs for the project
- Describe the project constraints that could potentially impact the risk and cost
- Describe the scenarios that could impact the project success

Identifying the risks is an essential task. The project team should identify high risks that could cause the team to become unable to deliver the product within the projected timeline and budget. They should be prioritized and tackled in the early phase of construction. The team should also develop a risk mitigation and contingency plan.

In order to keep the stakeholders involved in the whole process, a communication plan needs to be created at this phase. It is important for others to know: what the project team is doing, why they are doing it, the business case for it, etc.

The end of the inception phase is the first major project milestone: the Lifecycle Objective Milestone. The estimated expenditures are replaced by actual figures.

The project may be cancelled or re-thought if it fails to pass this milestone.

Elaboration phase

The goal of the elaboration phase is to further analyze the business domain and create the architecture that will serve as a base for future work. The project team must eliminate the risks that may have big impact on the system. Usually, the team develops a prototype to prove the concept in one or many iterations.

The complexity and length of this phase depends on many factors. The most important are:

- Size of the project: The size of the project and the complexity of this phase have a linear relationship. The bigger the project, the longer the elaboration phase will take.
- Novelty of the project: If the system is new, a new architecture needs to be created. As manager, you need to make sure that the team is composed of people who understand the business side and the technical side. Sometimes training and developing special skills may be required before the team can be ready to take such responsibility. Of course, all these cost money and time for the company and need to be included in the project planning.
- Existence of a pre-built system or framework that can be used as a base to develop the architecture: If such system exists, it can reduce the complexity of this phase.
- Technology to be deployed: If the technology is new to the team, there's a learning curve and skill development that will take place before the team can move to the construction phase.
- Complexity of the system: This is defined by the nature of the functions that needed to be implemented in the system. The more complex the functions, the more time is needed to understand them and build the right architecture that will support them.

In this phase, the following artifacts are generated:

- Revised Risk list
- More detailed use cases
- High level Architecture of the system
- Prototypes
- A development plan for the overall project showing iterations and the evaluation of each iteration

The end of this phase is the second most important milestone: The Lifecycle Architecture. The team reviews the proposed architecture, and the risk test result.

If the architecture is solid enough and the major risks have been addressed, the stakeholders may give the green light to move to the construction phase. On the other hand,

if the risks are not addressed, the estimate is higher than expected, or the stakeholders believe that the current architecture can't achieve their vision, the project may be aborted or re-considered completely.

Construction phase

Once the architecture has been approved and the risks have been eliminated, the remaining features of the system are developed and tested.

Here are the key points to keep in mind during this phase:

- 1- Starting with a pilot project: Pick a low priority project that's not critical and have a low visibility. The goal here is to prove that the software process works for your organization. There are many people who will go with the process, others will find it difficult and other will resist the change. You will need to identify all these problems and make the judgment of whether or not the new process can be implemented in your organization before initiating the real project.
- 2- Opting for paralleling constructions or having team spread on many locations introduces another level of complexity that the managers need to deal with. A clear project plan with measured milestones and robust architecture will help everyone in the team to know what needs to be done and when it is due without duplicating the efforts.
- 3- Adopting rigorous methods like Extreme Programming will help your team deliver high quality software.

The end of the construction phase is the third major milestone: Initial Operational Capability Milestone. At this stage, the team is ready to deliver a portion of the system, the user manual and the description of the release. If the project is stable, the stakeholders may decide to roll it out to production. The managers update their estimated expenditures by the actual ones. If the figures are not acceptable to the stakeholders, the project managers may be asked to adjust the future iterations to reduce the cost, speed up the process, or improve the quality of the system.

Transition phase

This is the phase where the product is deployed across the organization. Keep in mind that in the Rational Unified Process, you don't have to wait until the whole product is completed before rolling it out to production. Instead a minimum set of functions to make the system usable is defined, implemented and then deployed as soon as possible to get the feedback of the users. Those feedbacks are incorporated in the next iterations. This is actually the most important rule in the RUP process. I have seen many companies discard this rule and as a result, pay a high price in the end: a system that does not work and is very costly to fix afterwards. Rolling out the system in small pieces allows the management to have an accurate estimate about the status of the project, be proactive, and make the right decision at the right time.

On the other hand, having the end-users test each delivery and incorporate their feedback in the next releases guarantees the outcome of the final product and eliminates any surprises of having the wrong system or a system that doesn't support the business vision of the company.

Conclusion

Divide and conquer is a strategy that works in lot of disciplines especially in developing IT projects. Dividing the big systems into small, manageable sub-systems that can be tackled iteratively and delivered incrementally will maximize the likelihood of the success of the whole project, make it simpler to manage and adjust during the development process. RUP process uses this strategy to help the company develop products that go along with their vision and lead to high return on investment.

References

- [1] Philippe Krutchen, the Rational Unified Process, An introduction, Second Edition, Addison-Wesley, 2000
- [2] Software Project Management: A Unified Framework by Walker Royce, Addison-Wesley, 1998.
- [3] Software Cost Estimation with Cocomo II by Barry et al. (Prentice Hall, 2000)
- [4] The Mythical Man Month: Essays on Software Engineering, 2 Edition, by Frederick P. Brooks, Addison-Wesley, 2000.

This document is for informational purposes only. Tometa Software, Inc. MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2004 Tometa Software, Inc. All rights reserved.