

# Software Development Processes

Will Lin\*

Zhejiang Normal University  
Software Project Management

## Abstract

Software Process is the set of activities, methods, and practices that are used in the production and evolution of software. It has several models, which are some specific embodiments of a software process architecture. This article aims to explicate the conceptions of some important software development processes and make full use of their characteristics to manage software project.

**Keywords:** software engineering, project management, software development processes

## 1 Introduction

In software engineering, a software development methodology (also known as a system development methodology, software development life cycle, software development process, software process) is a splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.[for Medicare & Medicaid Services (CMS) Office of Information Service 2008]

Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology. Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization. For example, there are many specific software development processes that fit the spiral life-cycle model.

## 2 Approaches

Several software development approaches have been used since the origin of information technology, in two main categories. Typically an approach or a combination of approaches is chosen by management or a development team.

"Traditional" methodologies such as waterfall that have distinct phases are sometimes known as software development life cycle (SDLC) methodologies, though this term could also be used more generally to refer to any methodology. A "life cycle" approach with distinct phases is in contrast to Agile approaches which define a process of iteration, but where design, construction, and deployment of different pieces can occur simultaneously.

**Waterfall Model:** It is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases, typically(e.g., see Figure 1)

\*e-mail:1073318082@qq.com

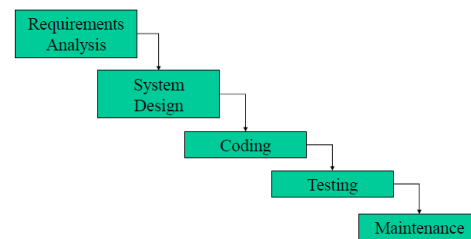
Copyright 2016. The material in this article is copyrighted by the respected authors. The article is based on work to support Software Project Management.

**Software Project Management (2016/29)**

Author Name: Will Lin  
University: Zhejiang Normal University  
Title: Software Development Processes  
Supervisor: Dr. Kenwright

- Requirements analysis resulting in a software requirements specification
- System design
- Coding
- Testing
- Maintenance

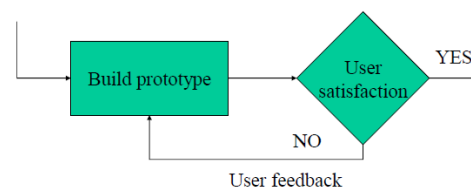
### Waterfall Model



**Figure 1:** The activities of the software development process represented in the waterfall model.

**Prototyping Model:** Software prototyping, is the development approach of activities during software development, the creation of prototypes, i.e., incomplete versions of the software program being developed.(e.g., see Figure 2)

### Prototyping Model



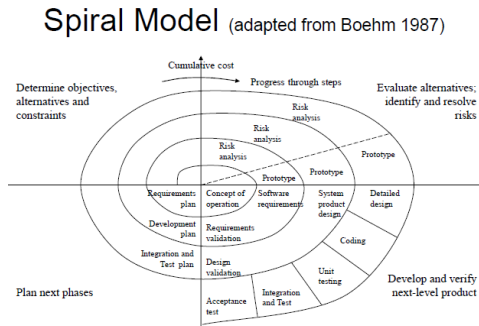
**Figure 2:** The activities of the software development process represented in the prototyping model.

The basic principles are:

- Not a standalone, complete development methodology, but rather an approach to handle selected parts of a larger, more traditional development methodology (i.e. incremental, spiral, or rapid application development (RAD)).
- Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
- User is involved throughout the development process, which increases the likelihood of user acceptance of the final implementation.
- Small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves to meet the users requirements.
- While most prototypes are developed with the expectation that they will be discarded, it is possible in some cases to evolve from prototype to working system.

- A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problems.

**Spiral Model:** Spiral development combines some key aspect of the waterfall model and rapid prototyping methodologies, in an effort to combine advantages of top-down and bottom-up concepts. It provided emphasis in a key area many felt had been neglected by other methodologies: deliberate iterative risk analysis, particularly suited to large-scale complex systems.(e.g., see Figure 3)



**Figure 3:** The activities of the software development process represented in the spiral model.

The basic principles are:

- Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to evaluate risks and weigh consideration of project continuation throughout the life cycle.
- "Each cycle involves a progression through the same sequence of steps, for each part of the product and for each of its levels of elaboration, from an overall concept-of-operation document down to the coding of each individual program." [Boehm 1986]
- Each trip around the spiral traverses four basic quadrants: (1) determine objectives, alternatives, and constraints of the iteration; (2) evaluate alternatives; Identify and resolve risks; (3) develop and verify deliverables from the iteration; and (4) plan the next iteration. [Richard H. Thayer 1986]
- Begin each cycle with an identification of stakeholders and their "win conditions", and end each cycle with review and commitment. [Boehm 2000]

**Incremental Model:** Various methods are acceptable for combining linear and iterative systems development methodologies, with the primary objective of each being to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process. (e.g., see Figure 4)

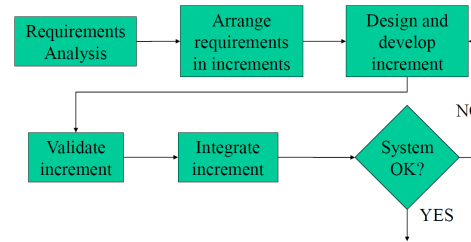
The basic principles are: [for Medicare & Medicaid Services (CMS) Office of Information Service 2008]

- A series of mini-Waterfalls are performed, where all phases of the Waterfall are completed for a small part of a system, before proceeding to the next increment, or
- Overall requirements are defined before proceeding to evolutionary, mini-Waterfall development of individual increments of a system, or
- The initial software concept, requirements analysis, and design of architecture and system core are defined via Waterfall, followed by iterative Prototyping, which culminates in installing the final prototype, a working system.

## 2.1 Purposes of Modelling

- To provide a common understanding
- To locate any inconsistencies, redundancies and omissions

## Incremental Model (cont'd)



**Figure 4:** The activities of the software development process represented in the incremental model.

- To reflect the development goals and provide early evaluation
- To assist the development team to understand any special situation

## 3 Conclusion

This article has stressed the need examine each project carefully to see if it has characteristics that suggest a particular approach or process model. These characteristics will also suggest the addition of specific activities to the project plan.

The classic waterfall process model, which attempts to minimize iteration, should lead to projects that are easy to control. Unfortunately, many projects do not lend themselves to this structure. Prototyping often reduces project uncertainties by allowing knowledge to be bought through experimentation.

## References

- BOEHM, B. 1986. A spiral model of software development and enhancement. *Annalen der Physik.* 2
- BOEHM, B. W. 2000. *Software Cost Estimation with Cocomo II*. Prentice Hall. 2
- FOR MEDICARE & MEDICAID SERVICES (CMS) OFFICE OF INFORMATION SERVICE, C., 2008. Selecting a development approach. 1, 2
- RICHARD H. THAYER, B. W. B. 1986. *software engineering project management*. Computer Society Press of the IEEE. 2