Software Development Processes

Yang Xiang(杨翔)*
Zhejiang Normal University
Software Project Management

Abstract

When we work in software engineering, we can easily feel that if we not have a specific method to develop software, it will cost a lot of money and time. Thus, applying an efficient methodology to our project is necessary. And now, Software Development Processes (Methods) can address these kind of problems effectively. Besides, the studies about SDP have matured and widely used, and it is still developing.

Keywords: software engineering, project management, planning, risk assessment

1 Introduction

The software development methodology (also known as SDM) framework didn't emerge until the 1960s. According to Elliott (2004) the systems development life cycle (SDLC) can be considered to be the oldest formalized methodology framework for building information systems. The main idea of the SDLC has been "to pursue the development of information systems in a very deliberate, structured and methodical way, requiring each stage of the life cycle—from inception of the idea to delivery of the final system—to be carried out rigidly and sequentially"[2] within the context of the framework being applied. The main target of this methodology framework in the 1960s was "to develop large scale functional business systems in an age of large scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines".(e.g., [Elliott 2004]).

Methodologies, processes, and frameworks range from specific prescriptive steps that can be used directly by an organization in day-to-day work, to flexible frameworks that an organization uses to generate a custom set of steps tailored to the needs of a specific project or group. In some cases a "sponsor" or "maintenance" organization distributes an official set of documents that describe the process. Specific examples include:

1970s

- Structured programming since 1969
- Cap Gemini SDM, originally from PANDATA, the first English translation was published in 1974. SDM stands for System Development Methodology

1980s

- Structured systems analysis and design method (SSADM) from 1980 onwards
- Information Requirement Analysis/Soft systems methodology

1990s

- Object-oriented programming (OOP) developed in the early 1960s, and became a dominant programming approach during the mid-1990s

*e-mail:424178617@qq.com

Software Project Management (2016/17)
Author Name: Yang Xiang(杨翔)
University: Zhejiang Normal University
Title: Software Development Processes
Supervisor: Dr. Kenwright

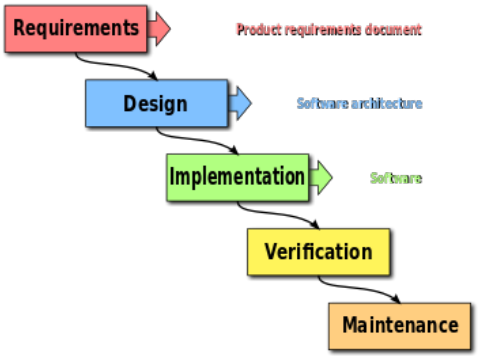

Figure 1: *Waterfall model - the classic waterfall approach to programming. The waterfall model uses a series of phases to move the project along. Each phase creates a deliverable, usually a document that captures what the phase has accomplished.*

- Rapid application development (RAD), since 1991
- Dynamic systems development method (DSDM), since 1994
- Scrum, since 1995
- Team software process, since 1998
- Rational Unified Process (RUP), maintained by IBM since 1998
- Extreme programming, since 1999

2000s

- Agile Unified Process (AUP) maintained since 2005 by Scott Ambler
- Disciplined agile delivery (DAD) Superseded of AUP

(e.g., [Wikipedia])

2 Acceptable System Development Methodologies

Waterfall Model(Figure1)

In Royce's original waterfall model, the following phases are followed in order:

1. System and software requirements: captured in a product requirements document
2. Analysis: resulting in models, schema, and business rules
3. Design: resulting in the software architecture
4. Coding: the development, proving, and integration of software
5. Testing: the systematic discovery and debugging of defects
6. Operations: the installation, migration, support, and maintenance of complete systems

Thus the waterfall model maintains that one should move to a phase only when its preceding phase is reviewed and verified.

Various modified waterfall models (including Royce's final model), however, can include slight or major variations on this process.[3] These variations included returning to the previous cycle after flaws were found downstream, or returning all the way to the design phase if downstream phases deemed insufficient.

Spiral Model (adapted from Boehm 1987)(Figure2)(e.g., [Wikipedia])

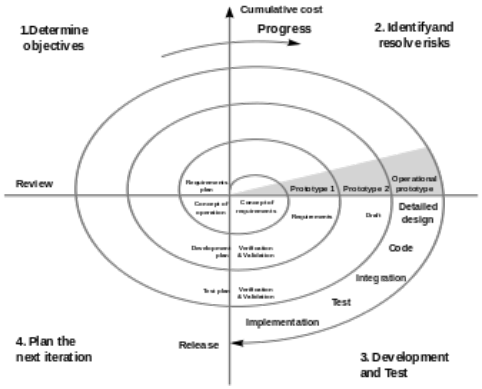

Figure 2: *Spiral model (Boehm, 2000). A number of misconceptions stem from oversimplifications in this widely circulated diagram (there are some errors in this diagram).*

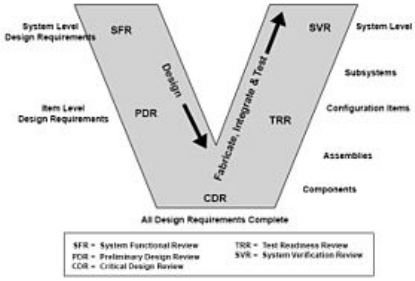

Figure 3: *Systems engineering and verification.*

Sequentially defining the key artifacts for a project often lowers the possibility of developing a system that meets stakeholder "win conditions" (objectives and constraints).

This invariant excludes "hazardous spiral look-alike" processes that use a sequence of incremental waterfall passes in settings where the underlying assumptions of the waterfall model do not apply. Boehm lists these assumptions as follows:

1. The requirements are known in advance of implementation.
2. The requirements have no unresolved, high-risk implications, such as risks due to cost, schedule, performance, safety, security, user interfaces, organizational impacts, etc.
3. The nature of the requirements will not change very much during development or evolution.
4. The requirements are compatible with all the key system stakeholders' expectations, including users, customer, developers, maintainers, and investors.
5. The right architecture for implementing the requirements is well understood.
6. There is enough calendar time to proceed sequentially.

In situations where these assumptions do apply, it is a project risk not to specify the requirements and proceed sequentially. The waterfall model thus becomes a risk-driven special case of the spiral model.

V Model(Figure3) (e.g., [vmo 2001])

The two streams

Specification stream

The specification stream mainly consists of:

- User requirement specifications
- Functional requirement specifications Design specifications

Testing stream

The testing stream generally consists of:

- Installation qualification (IQ)
- Operational qualification (OQ)
- Performance qualification (PQ)

The development stream can consist (depending on the system type and the development scope) of customization, configuration or coding.

3 Overview

A software development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One system development methodology is not necessarily suitable for use by all projects. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations. CMS has considered each of the major prescribed methodologies in context with CMS' business, applications, organization, and technical environments. As a result, CMS requires the use of any of the following linear and iterative methodologies for CMS systems development, as appropriate.

4 Conclusion

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.

References

ELLIOTT, G. 2004. *Global Business Information Technology: an integrated systems approach*. Pearson Education. 1

2001. *Systems Engineering Fundamentals*. Defense Acquisition University Press. 2

WIKIPEDIA. Software development process. 1