

Software Project Management

Zachary Zhang 13211242

Zhejiang Normal University

Software Development Processes

Abstract 1

A software development methodology is a system development, software development life cycle 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.

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.

Introduction

1. Waterfall Development 1

The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases, typically:

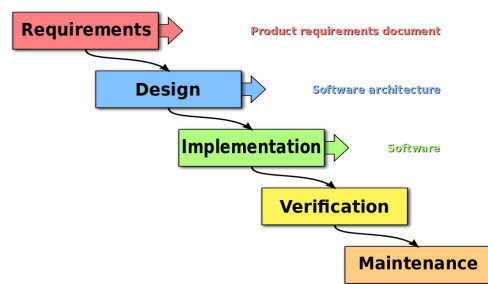
- Requirements analysis resulting in a software requirements specification

- Software design
- Implementation
- Testing
- Integration, if there are multiple subsystems
- Deployment (or Installation)
- Maintenance

The first formal description of the method is often cited as an article published by Winston W. Royce in 1970 although Royce did not use the term "waterfall" in this article. The basic principles are:

- Project is divided into sequential phases, with some overlap and splashback acceptable between phases.
- Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.
- Tight control is maintained over the life of the project via extensive written documentation, formal reviews, and approval/sign off by the user and information technology management occurring at the end of most phases before beginning the next phase. Written documentation is an explicit deliverable of each phase.
- Iterate:- Create a prototype, then the real system. Repeat each phase using new information gained and the entire process at least once before delivering the live

system.



The waterfall model with no iteration or prototype is known as the 'traditional' engineering approach applied to software engineering. As a consequence, unless part of the project plan, a strict waterfall approach discourages revisiting and revising any prior phase once it is complete. This "inflexibility" has been a source of criticism by supporters of other more "flexible" models. It has been widely blamed for several large-scale government projects running over budget, over time and sometimes failing to deliver on requirements due to the Big Design Up Front approach. Except when contractually required, the waterfall model has been largely superseded by more flexible and versatile methodologies developed specifically for software development. See Criticism of Waterfall model.

The waterfall model is also commonly taught with the mnemonic A Dance in the Dark Every Monday, representing Analysis, Design, Implementation, Testing, Documentation and Execution, and Maintenance.

2. Prototyping 1

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.

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.

3. Incremental development 1

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.

The basic principles are:

- 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.

4. Rational Unified Process 2

RUP is based on a set building blocks and content elements, describing what is to be produced, the necessary skills required and the step-by-step explanation describing how specific development goals are to be achieved. The main building blocks, or content elements, are the following:

- Roles (who) – A role defines a set of related skills, competencies and responsibilities.
- Work products (what) – A work product represents something resulting from a task, including all the documents and models produced while working through the process.
- Tasks (how) – A task describes a unit of work assigned to a Role that provides a meaningful result.

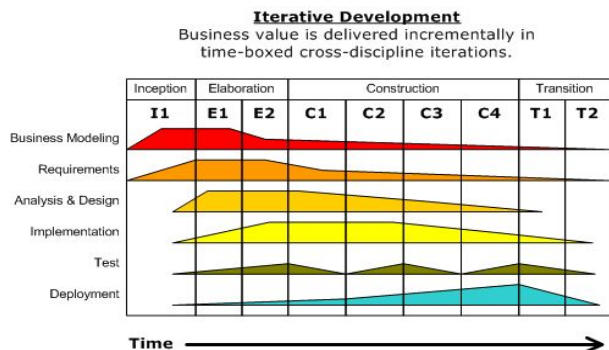
Within each iteration, the tasks are categorized into nine disciplines:

- Six "engineering disciplines"
- Business modelling
- Requirements
- Analysis and design

- Implementation
- Test
- Deployment

Three supporting disciplines

- Configuration and change management
- Project management
- Environment



References

1. https://en.wikipedia.org/wiki/Software_development_process
2. https://en.wikipedia.org/wiki/Rational_Unified_Process