

# Software Development Processes(Methods)

Zanlai Hu\*

Zhejiang Normal University  
Software Project Management

## Abstract

In this sample paper, we describe the information about Software Development Processes(also known as a system development methodology, software development life cycle, software development methodology, software process). This article could provide you a Comparatively comprehensive knowledge about it , as information would be analyzed and systematized. And we know that it involves a lot of practice. The knowledge of software development processes is far more than this article could present - however, it gives a brief and brief and concise explanation to software development processes. It's not that we surely excluded anything from this article, but it would be unrealistic to try and cover every possible aspect in a few pages. For the sake of practicality, we discuss its main life-cycle models and necessary steps, such as, waterfall model and identifying needs.

**Keywords:** Software Development Processes, Software Process Model, process

## 1 Introduction

Software development process(also known as a system development methodology, software development life cycle, software development methodology, software process) a splitting of software development work into several key parts to make better planning and management. It enables you to manage and deliver complex projects on time and within budget. Most importantly, it enables you to create better software solutions - more reliable, scalable and efficient. This article gives a brief introduction to software project management and project management in general. We address the following concepts in this article:

- ✓ What is software development process?
- ✓ What is Software Process Model?
- ✓ Why Modelling?
- ✓ Which model to choose?

**What is software development process?** Software development process(also known as a system development methodology, software development life cycle, software development methodology, software process) is a splitting of software development work into several key parts to make better planning and management. Also, software Process can be defined as the set of activities, methods, and practices that are used in the production and evolution of software. Generally speaking, Software development process could be divided into eight processes:

- **Identifying Needs** The project manager talk to stakeholder to clarify and identify needs, then make a list of correlative function module by professional tools. Besides, smaller function modules are needed to contain more details so that preliminary interface can be defined and give stakeholder a clear expression. And project manager talk to stakeholder and confirm needs again. This is an effective way

to show stakeholders that their needs are understood and that those specific needs will be addressed with a vision and scope document

- **Outline Design** Developers should make outline(also known as system design) to discuss the design of software includes organization structure, function allocation, use cases, data design and so on which is a base of the detailed design.
- **The detailed design** The detailed design is based on the Outline design. In detailed design, developers need to describe every main algorithm to realize concrete function modules, data structure and the hierarchy and method call relation of class. It is required to explain every program in every hierarchy so that coding and testing could be succeed. The detailed design should be as minute as it can.
- **Coding** In the stage of coding, developers begin to program and realize every part of project which is on the basis of the detailed design. At present, this phase should cost no more than a second of the time. It often cost a third of time to complete coding. Coding could be more effective if the part of design is very clear and good. Developers must be very careful in this part. Little mistake may cause big problem. What's more, the earlier the mistake happen, the worse it would cause. So communication between teammates and risk plans are playing an important role in it. As we said, Bug will always exist.
- **Testing** Testing the program that has been finished and give it to stakeholder if there is nothing wrong. Testing can be split into built-in-test and external test by executing, or it can be split into module testing and Overall test by its scope, or it can be split into Normal operation test and Abnormal situation test by Condition, or it can be split into Full coverage testing and sample test. Testing is an significant process in developing. It starts when the project is designed and ends after it finished. It runs through all processes of software developing.
- **Software Delivery** When the project has been tested and up to grade. It will be delivered to stakeholders with Data dictionary, User guide, demand report, design report, testing report and contract.
- **Check and Accept** Stakeholder accept software.
- **maintenance** The software should be altered and improved by stakeholders' requirement.

**What is Software Process Model?** Software Process Model is one specific embodiment of a software process architecture. There are various kinds of models.

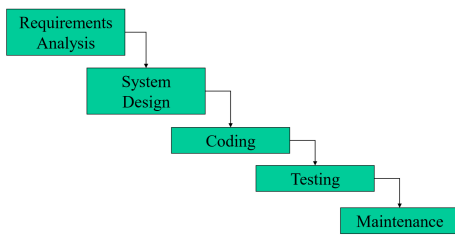
- Waterfall Model
- V Model
- Spiral Model
- Prototyping Model
- Phased Development Model(incremental development model and iterative development model)
- Operational Specification Model
- Transformation Model

**Why Modelling?** Software Process Model is one specific embodiment of a software process architecture. There are various kinds of models.

- To provide a common understanding
- To locate any inconsistencies, redundancies and omissions
- To reflect the development goals and provide early evaluation
- To assist the development team to understand any special situation

**Which model to choose?** Look at risks and uncertainties e.g.

\*e-mail:361206768@qq.com



**Figure 1: Waterfall Model**

- are requirements well understood?
- are technologies to be used well understood?

Look at the type of application being built e.g.

- information system? embedded system?
- criticality? differences between target and development environments?

Clients' own requirements

- need to use a particular method

## 2 Software Process Models

Process models are processes of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. Since the process model is at the type level, a process is an instantiation of it. The same process model is used repeatedly for the development of many applications and thus, has many instantiations. One possible use of a process model is to prescribe how things must/should/could be done in contrast to the process itself which is really what happens. A process model is roughly an anticipation of what the process will look like. What the process shall be will be determined during actual system development.

[https://en.wikipedia.org/wiki/Process\\_modeling](https://en.wikipedia.org/wiki/Process_modeling)

### Waterfall Model

- classical
- one-shot approach
- effective control
- limited scope of iteration
- long cycle time
- not suitable for system of high uncertainty

The sequence of events for the waterfall model are analysis, design, coding, testing and maintenance. The linear sequential model of software development is also known as the Waterfall model (or Classical life cycle model)

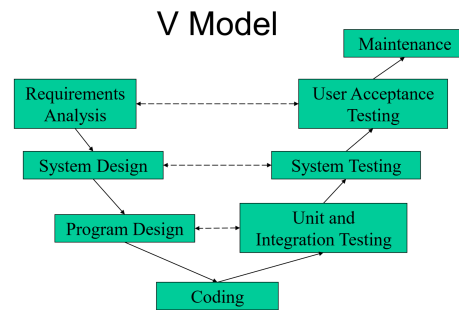
Waterfall Variations

- Sashimi waterfall: activities are allowed to overlap
- Waterfall with sub projects: implementation of different components proceeds in parallel
- Waterfall with risk reduction: an initial risk analysis helps mitigate risks in later phases of implementation

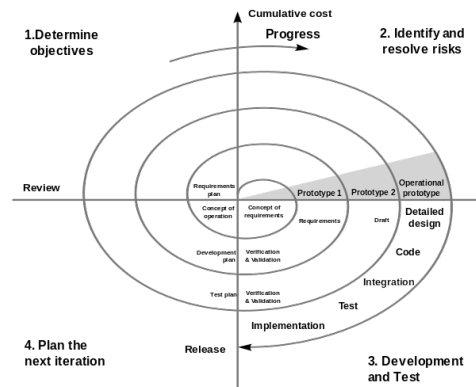
### V Model

- Additional validation process introduced
- Relate testing to analysis and design
- Loop back in case of discrepancy

Why V-Model is used instead of waterfall and spiral? Explain advantages and disadvantages?



**Figure 2: V model**



**Figure 3: Spiral model**

The V-model is considered to be an extension of the Waterfall model. Instead of a downward linear path the process moves downward until it reaches the coding stage whereupon it begins moving upward until it passes user acceptance. Significant difference between V-and Waterfall models is that the former includes well-defined Verification and Validation phases.

Advantages

- V-model is easy to understand and provides a great deal of structure.
- It is most suitable when requirements are stable and unchanging and when potential problems can be reasonably predicted.
- The V-model presents verification and validation processes with more specificity than does the Waterfall model.

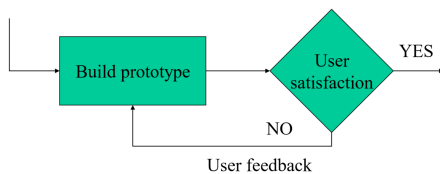
Disadvantage

- Same disadvantages as the Waterfall model.
- Like Waterfall does not allow enough flexibility and that projects can't move forward at a satisfactory pace.
- Most companies that want to turn a profit don't want to spend too much time and effort in the planning stage.
- Not only that but most software projects are complex and often change requirements or make enhancement requests throughout the development life cycle

### Spiral Model

- Evolutionary approach
- Iterative development combined with risk management
- Risk analysis results in "go, re-do, no-go" decision
- Four major activities(Planning, Risk analysis, Engineering, Evaluation)

The most important feature of spiral is risk management. In the spiral model 'risk analysis' is performed in every loop.



**Figure 4: Prototyping Model**

### Prototyping Model

- Additional validation process introduced
- Relate testing to analysis and design
- Loop back in case of discrepancy

#### 1.Goals

- meet (some) user requirements at an early stage
- reduce risk and uncertainty
- verify a design or implementation approach

#### 2.Should always answer specific questions; goals must be identified Classification of Prototype

- Throw-away After users agree the requirements of the system, the prototype will be discarded.
- Evolutionary Modifications are based on the existing prototype.
- Incremental Functions will be arranged and built accordingly.

#### Benefits of Prototyping

- Learning by doing
- Improved communication
- Improved user involvement
- Clarification of partially-known requirements

#### Prototyping Sequences

- Requirements gathering
- Quick design
- Prototype construction
- Customer evaluation
- Refinement
- Loop back to quick design for fine tuning
- Product engineering

#### Benefits of Prototyping

- Demonstration of the consistency and completeness of a specification
- Reduced need for documentation
- Reduced maintenance costs
- Feature constraint
- Production of expected results for testing real system

#### Drawbacks of Prototyping

- Users sometimes misunderstand the role of the prototype
- Lack of project standards possible
- Lack of control
- Additional expense
- Machine efficiency
- Close proximity of developers

#### Forms of Prototypes

- Mock-ups
- Simulated interaction
- Partial working model

#### Prototype Products

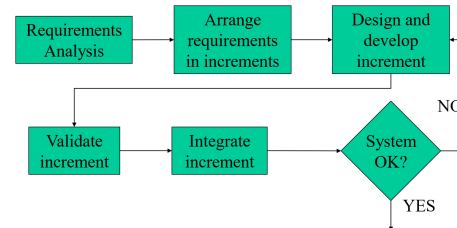
### Iterative



### Incremental



**Figure 5: Iterative and Incremental**



**Figure 6: Incremental Model**

- Human - computer interface
- System functionality

#### Prototype Changes

##### Three categories

- Cosmetic (35 percents) screen layout
- Local (60 percents) screen processing
- Global (5 percents) multi-parts processing and design review

The prototyping model of software development is a useful approach when a customer cannot define requirements clearly.

### Phased Development

- Reduce cycle time
- Two parallel systems: operational system (Release n) and development system (Release n+1)
- Two approaches incremental and iterative

### Iterative and Incremental

#### Incremental Model

- Break system into small components
- Implement and deliver small components in sequence
- Every delivered component provides extra functionality to user

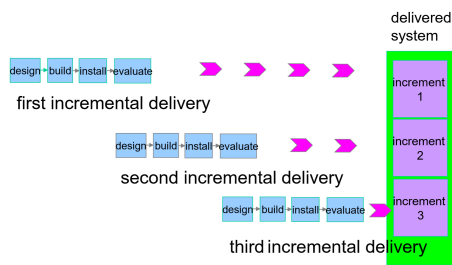
The incremental model of software development is a good approach when a working core product is required quickly.

#### Iterative Model

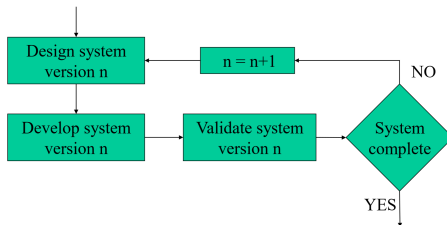
- Deliver full system in the beginning
- Enhance functionality in new releases

#### Combined Incremental and Iterative Model

- Every new release includes extra functionality and enhancement of existing functionality
- Popularly used in software industry



**Figure 7: Incremental delivery**



**Figure 8: Iterative Model**

#### Ranking the Increments

- Rank by value to cost ratio
- V = value to customer (score 1 - 10)
- C = cost (score 1 - 10)
- Value to cost ratio =  $V/C$

#### Advantages of Phased Development

- Early feedback
- Less possible requirement changes
- Early benefits for users
- Improved cash flow
- Capture early market
- Easier to control and manage
- Facilitate early training
- Can be temporarily abandoned
- Increase job satisfaction

#### Disadvantages of Phased Development

- 'Software breakage'
- Reduced productivity

#### Operational Specification Model

- Executable or translatable specification
- Use executables to demonstrate system behaviour
- Resolve requirement uncertainties in early stage
- Merging functionality and system design

#### Transformational Model

- Transform a specification into delivered system
- Requires automated support
- Relies on formal specification method

### 3 History of Software Development process

The software development process 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 methodol-

ogy 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".[2]

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.

[https://en.wikipedia.org/wiki/Software\\_development\\_process#Approaches](https://en.wikipedia.org/wiki/Software_development_process#Approaches)

### 4 Project Considerations

- Control systems
- Information systems
- General applications
- Specialized techniques
- Hardware environment
- Safety-critical systems

### 5 Overview

software development process(also known as a system development methodology, software development life cycle, software development methodology, software process) a splitting of software development work into several key parts to make better planning and management. It is consider a subset of the systems development life cycle. Different kinds of frameworks have its own recognized advantages and disadvantages. To discuss and analyze them, and choose one appropriate for developing.

### 6 Methods/Techniques

#### Technical Constraints

- Type of the system to be developed
- Risks and uncertainties of the project
- User requirements concerning implementation

#### Technical Approach

- Selected methodology or process model(s)
- Development methods
- Required software tools
- Target hardware/software environment

#### Technical Implementation

- Development environment
- Maintenance environment
- Training

#### Technical Implications

#### Project products and activities

- effect on schedule duration and overall project effort

#### Financial

- report used to produce costings

## 7 Conclusion

This article gives a brief explanation to the Software Development Process. And it includes professional knowledge which is accepted by professional organizations. what it express can provide a high-level view of Software Development Process and give some example of Software Process Model with particular explanation.

## Acknowledgements

I would like to thank my classmates and reviewers for taking time out of their valuable schedule to help make this report more concise and clearer.

## References

- [1]Software Project Management Theory and Case Analysis by Jiyi Wu
- [2]Software Project Management by Bob Hughes and Mike Cotterell
- [3]Introduction about tne Management Technologg and Rpplication of Software Project by Yuanjun Wu
- [4]From <http://baike.baidu.com/>
- [5]From Wikipedia, the free encyclopedia