

PAPER • OPEN ACCESS

Development Trend of Software Process Model

To cite this article: Liangliang Hao *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **569** 052077

View the [article online](#) for updates and enhancements.

Development Trend of Software Process Model

Liangliang Hao, Jinsheng Zhang and Xiaoyu Ma

Rocket Force University of Engineering Xi'an, ShaanXi ,710025 China

haoliangliangno1@126.com

Abstract. With the continuous development of software technology, the scale and complexity of software have never been higher, and the difficulty of design and development has also increased. The software process changes the chaotic situation of software development and provides an effective roadmap for the software development team to make the development process more controllable and stable. This paper briefly introduces the software process, and introduces the existing software process model in detail. Finally, the development trend of software process model is forecasted.

1. Introduction

The concept of "software engineering" was first proposed at the North Atlantic Treaty Organization (NATO) Software Engineering Conference held in Germany in 1968. With the continuous development of network technology, software engineering has also developed rapidly. Software engineering is not only a creative process, but also a gradual process. Software engineering usually follows a series of steps, which we call software process. By controlling the software development process, the software engineering activities become more stable, controllable and organized, and the software development activities become orderly and efficient. In this process, different models, namely Software Process Model (SPM), are produced.

2. Overview of Software Process Models

Software process is the process of developing software, that is, when developing products or building systems, we must follow a series of steps. Howard Baetjer Jr describes the software process as follows:

Software process is a process of dialogue, in which knowledge needed to be transformed into software is acquired and realized. Software process... It is an iterative process in which the evolving tool itself acts as a medium of communication, from which participants can gain more useful knowledge.

After nearly half a century of development, software developers gain experience from past projects and summarize the best practices for producing high-quality software. It is found that software development usually includes the following stages:

- | | |
|-------------------|----------------------|
| ● Demand analysis | ● Unit testing |
| ● System Design | ● Integrated testing |
| ● Programming | ● System delivery |
| ● Programming | ● Maintenance |

Each stage is itself a set of activities, including constraints, outputs and resources. However, in the actual software development process, different organizations will have different software production processes. This leads to different process models.



Software process model provides a specific roadmap for software engineering work, which specifies the activity flow, action, task, degree of iteration, work product and how to organize the work to be completed. Software engineers adopt a process model according to the requirements and follow the model throughout the software development process.

In the software process model, some are prescriptive, planning software development activities; others are descriptive, reflecting the way actual software development activities are carried out. In theory, the two models should be similar or identical, but not in practice. Therefore, it is particularly important to establish process models and discuss their sub-stages. It can help the development team understand the gap between the ideal situation and the actual situation of the development process, so as to improve the development process according to the actual situation.

There are also the following reasons for modeling software processes:

- 1) When the development team records and describes the development process, it helps the whole development team to have a common understanding of the activities, resources and constraints in software development.
- 2) Identify inconsistencies, redundancies and omissions in the process and its components in a timely manner to make the process more effective and ultimately reflected in the product;
- 3) The model reflects the development objectives, and the development team can evaluate the suitability of the development activities according to the objectives.
- 4) In the process of software development, every process should be tailored according to the specific situation. Building process model is helpful for team to tailor its process, and the development process is more rapid.

3. Existing software process models

3.1. Waterfall Model

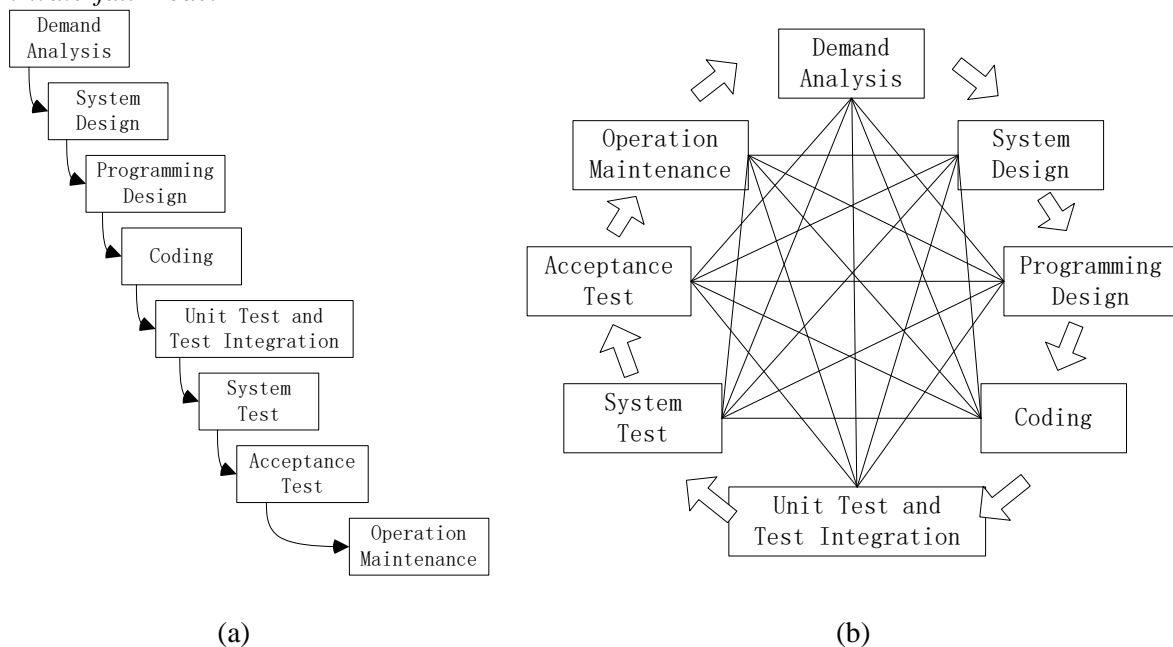


Fig.1 Waterfall Model

Waterfall model, also known as classic life cycle, provides a systematic and sequential approach to software development. It starts with user requirements description and provides complete software support through design, coding, testing, operation and maintenance processes. It requires each process activity to have milestones and deliverables associated with it so that it can judge how much work remains to be done before the final completion. The flow chart is shown in Figure 1 (a).

However, due to practical constraints, activities in software development process do not always have milestones, such as requirements analysis, developers are not always able to fully understand the requirements, or customers will change the requirements. Therefore, if the actual development process is not controlled, it may lead to the development process as shown in Figure 1 (b): turning from one job to another.

3.2. V model

V-model is the deformation of waterfall model. It describes the relationship between quality assurance action and communication, modeling related action and early construction related action. As software development progresses step by step along the left side of the V-model, the requirements of basic problems are refined step by step, forming a detailed and technical description of problems and solutions. After the coding work is finished, push up along the right step and perform the quality assurance action. Waterfall models usually focus on documents and products, while V models focus on activities and correctness.

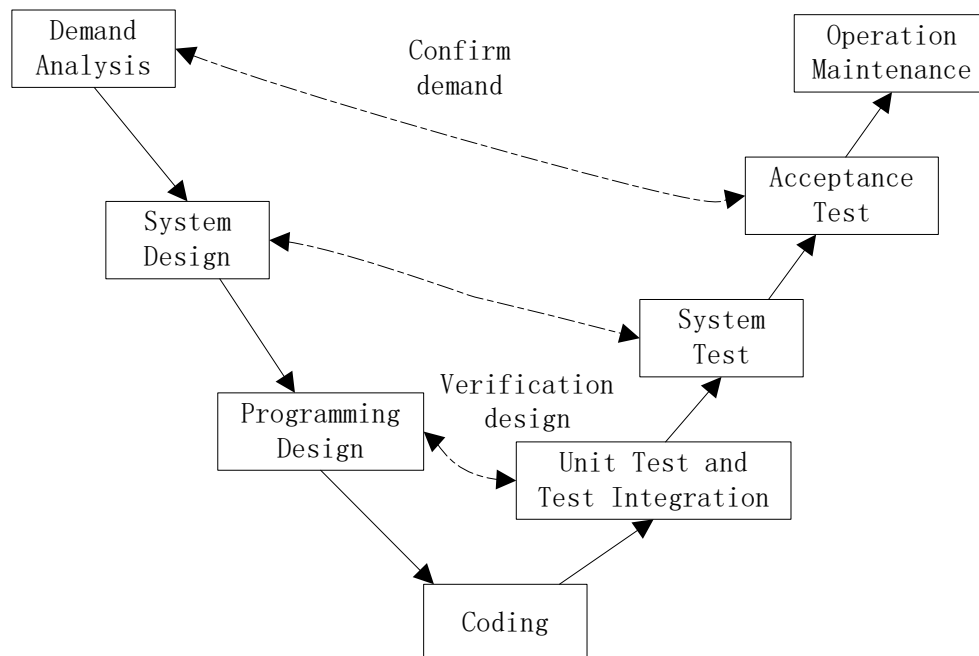


Fig.2 V-Model

3.3. Incremental Development Model

In incremental development model, software is divided into different subsystems. In the development process, a small functional subsystem is first developed, and then new functions are added to each new delivery until all functions are developed. This model is suitable for defining the initial software requirements clearly, but the linear model is not suitable for the whole process, or users are eager to provide a set of products that can meet the basic requirements.

When using the model, the first delivery is usually the core product, which meets the basic requirements, but many additional features are not yet available. After the user uses it, he evaluates it, and the developer develops the next incremental development according to the evaluation. Each increment adds features and functions until the final product is produced.

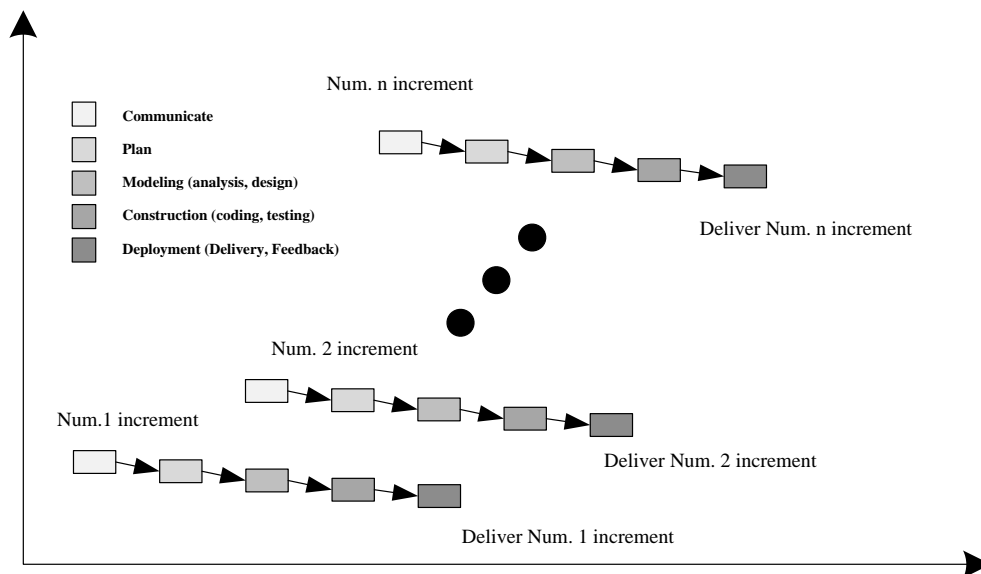


Fig.3 Incremental Development Model

3.4. Prototyping Model

Prototype is a product with incomplete functions. Prototype development is to quickly construct the whole system or part of the system by developers to understand or clarify problems, and to enable users and developers to check the planned system to determine whether it is appropriate. At the same time, the prototyping model allows users to update requirements or designs repeatedly, to ensure that developers and users have a common understanding of requirements and final products, and to reduce risks and uncertainties in development.

3.5. Helix Model

Spiral model is a risk-driven process model generator, which is usually used to rapidly develop progressively improved versions. It uses the spiral approach to deepen the definition and implementation depth step by step. In each iteration process, it carries out risk analysis of requirements and constraints, so as to reduce the risk, so that users and developers can recognize and be satisfied with the solution. At the same time, according to the results of risk analysis, different choices are weighed, and the feasibility or expectation is verified by prototyping.

3.6. Unified Process

Unified process (UP) is a kind of software process model that is "use case driven, architecture-centered, iterative and incremental". It tries to realize the good characteristics and properties of traditional software process through agile process, powerful software structure, and provide architects with correct goals.

The unification process mainly includes the following stages:

- 1) Initial stage: communication with users and planning activities;
- 2) Elaboration stage: Modeling activities of communication and general process model;
- 3) Construction stage;
- 4) Transition phase: Delivery of customer testing and activity feedback. By the end, increments become releasable versions;
- 5) Production phase: Provide operating environment, submit and evaluate.

4. Development Trend of Software Process Models

The software development model provides us with a problem solution which can be used for reference. However, with the deepening of developers' and users' understanding of the problem and the summary of past experience, the knowledge, technology and business in different fields are merged to get a new and appropriate solution, and the software process model is developed in this process.

4.1. Component-based Development

Component-based development model refers to the integration of software components that provide specific functions into the software under construction through well-defined interfaces. Component-based development model is essentially an evolutionary model. It has the characteristics of spiral model to construct software by selecting generations. It aims to build a reusable standard set so that software can be built by choosing a standard structure and reusable framework. That is to say, software products can be realized by combining reusable components.

The top priority of modeling and building activities is to select reusable components. Without considering component development technology, the component-based development model consists of the following steps (using evolutionary approach):

- 1) Analyse the requirements, research their fields, and evaluate the available component products.
- 2) Consider the problem of component integration;
- 3) Design software architecture to accommodate these components;
- 4) Integrating components into the architecture;
- 5) Perform adequate tests to ensure normal functioning.

Component-based development model can achieve software reuse, which can bring huge benefits, shorten the development cycle and reduce project development costs.

4.2. Formal Method Model

Formal methods model provides defect-free software by generating formal mathematical specifications for computer, which enables developers to explain, develop and validate the system with strict mathematical symbols. Formal method can avoid ambiguity, incompleteness and inconsistency in software development by applying mathematical analysis. In the design stage, formal methods can be verified by program, so that software developers can find and correct problems easily ignored, and produce high-quality software products. Therefore, this method is usually used in the development of software with high security concerns, or in the process of software development, such as aircraft and medical facilities, which can easily lead to significant economic losses. However, the formal model has not been popularized yet, mainly because:

- 1) Formal model development is time-consuming and costly.
- 2) Only a few programmers have the background of applying formal methods and need a lot of training.
- 3) It is difficult to use this model to communicate with customers with low technical level.

4.3. Layer-based Model

Based on Aspect-based model, the software development process is divided, and developers focus on specific modules to achieve modularization, which maximizes the cohesion of each module and minimizes coupling. It can reduce the risk of regression errors by focusing on specific modules to isolate code and so on. Its main goal is to mark the boundary of various modules in the requirements as early as possible, then specify, modularize and write code for them.

5. Conclusion

In the relatively short history of software, practitioners and researchers have developed a series of software process models, which effectively promote the fundamental changes in the way computer software is built. Software process, as the best way to improve software quality and software

development efficiency, is bound to move towards the direction of high quality, high efficiency and high security with the introduction of various new software technologies and tools.

References

- [1] Stephen R. Schach. Object-Oriented and Classical Software Engineering (Eighth Edition)[M]. Machinery Industry Press,2012
- [2] Shari Lawrence Pfleeger, Joanne M. Atlee. Software Engineering: Theory and Practice (Fourth Edition)[M], People's Posts and Telecommunications Publishing House,2014
- [3] Roger S. Pressman,Bruce R. Maxim. Software Engineering: A Practitioner's Approach (Eighth Edition)[M]. Machinery Industry Press,2016(in Chinese)
- [4] Shuting W, Development Trend of Software Engineering in China [J],Youth,2012.07: 378-379(in Chinese)
- [5] Fuqing Y, Development of Software Engineering Technology [J], Journal of software , 2005, 16(1): 1-7(in Chinese)
- [6] Na L The Development of Software Engineering Technology [J], Software Magazine ,2014,35 (3) : 204-205(in Chinese)