

# The Evolvement of GIS Developing Approach

Wu Sheng    Wang Jiayao

*(Surveying and Mapping Institute of PLA, Zhengzhou 450052)*

**Abstract**    The software engineering approach used in GIS is developing along with computer science and software engineering technology and meeting the GIS developing better and better. This Paper reviewed the developing approach's history of GIS. An object-oriented approach, unified software develop process is detailed introduced Also, The issue to select an engineering approach for GIS Project is discussed.

Kommentar [gxx1]: how to

**Keywords**    Software Engineering,    Linear Sequence Model,    Prototyping,    Rapid Application Develop,    Object-oriented Approach,    Unified Software Develop Process

## 1 Introduction

Great changes have taken place in the area of GIS during the past decade. The increasing hardware performance and software engineering technology contribute to the advent of more powerful, credible and complex GIS. The complication could result in miracle if only the GIS software developed successfully. But more often, the complication brings GIS developers problems and adds many difficulties to the developing work.

By and large, GIS is a big and complicated software system. To secure the developing quality and reduce cost, a scientific developing approach is indispensable.

Developing approaches are not used in the development of software project in the early days. In these days software programming was regarded as some kind of “art” and there were no rules to restrict these arbitrary “artists”. Softwares at that time used to be small and independent, and programmers hardly need any communication between each other. There were not any general and standard developing approaches put into practice until the advent of the first software project that needs a group of organized workers to work together. Mistakes and lessons get from the cooperative developing process are precious experience. When they were written in document and distributed to Development Team members as rules to be complied with, they were practically standards. Now all the perfect and successful approaches are written into textbooks and there are lots of tools based on these standards for the software developers. So when we complain the complication and cost of GIS development, at least one fact can console us——we needn't to devise the developing approach by ourselves.

## 2 GIS Developing Approaches

### 2.1 Linear Sequence Model

Linear Sequence Model was also called Life Cycle, Waterfall etc. During the development process, developing work orderly come from previous stage to next stage. The entire process contains system inquisition, demand analysis, system design, programming, testing and maintenance. Sometimes, information got from certain stage obliges the developing work to go back to previous stage (picture 1).

The biggest problem of this approach is that customers cannot practise the system until it is accomplished entirely. So it is very difficult to develop a product to satisfy customer at lowest cost. If customers cannot provide clear requirements, or the development team cannot grasp them, it's cost much to modify the complete system. Another disadvantage of this developing approach is that developers are often obstructed in one stage and cannot carry on with the following stages. For example, the developing work is delayed in the design stage, the programming stage, which depend much on the design, would not commence. In fact, time wasted in wait is much longer than in work.

To avoid these problems, an experienced developer would find most fault and danger in the early stage of the Life Cycle, "if you cannot actively find the dangers existing in the project, they will attack you actively"(Tom Gild). Usually, after each stage is finished, developers will check it carefully and communicate with customers to evaluate if their work meets their requirements, and in this way developer can control the quality of the system.

Anyway, the approach has been mature and formulaic, and it's the most widely used process model in the software engineering and dominates the GIS developing.

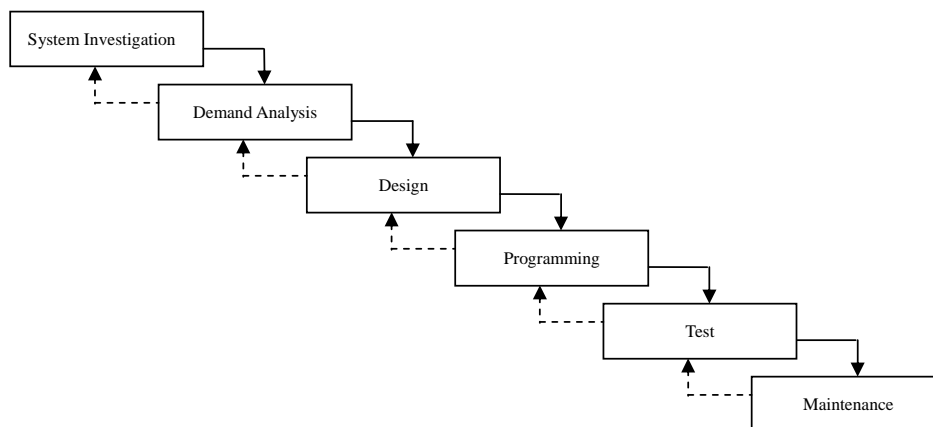


Figure1. Traditional life cycle developing approach

## 2.2 Prototyping

What shall we do when we don't know what to do? The key is Prototyping. Sometime customer may give general target, but cannot describe the detailed requirements of input, output and processing etc. In this case developer can build a primary version for customer to try, and modify it according to customer's feedback to form the second version, and so as the third version until it is perfect to customer finally (Figure 2).

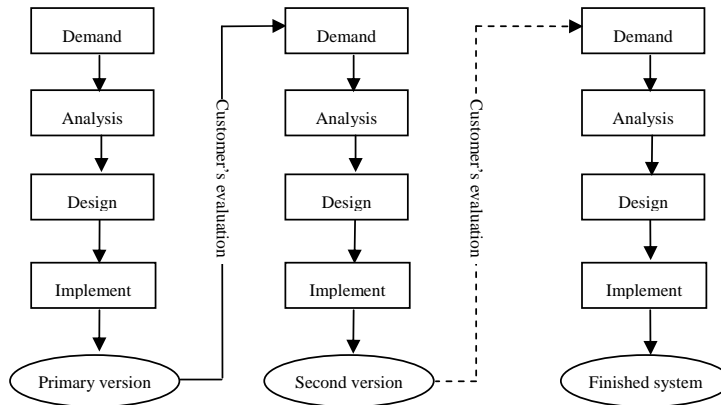


Figure2. Prototyping

There are two means to implement Prototyping. One is to build the core of the system first, and add its function component step by step. The other is to build main part of all the function components first, and add subordinate part. To either means, prototype is adjusted and corrected according to the customer's feedback. The whole process is iterative.

As the prototype can be provided to customer quickly, customer can evaluate the prototype and give their suggestion in time. This is the advantage of the prototyping approach.

After the "pleasure" of seeing the prototype begin to work quickly, the customer and developer will find problems like the programming languages being used in the system are not suitable for improving the system speed (for example, not using Visual C++ but Visual Basic); the algorithm adopted is not efficient or not precise enough (it was adopted only because it was available). The worse, system quality and longtime maintainability are not taken into account at all. The result is on one hand the customers ask the system to be modified ceaselessly, and on the other hand developers will reject or limit their demands for lack of fund. If they can't compromise, the controversy would come into endless contradiction.

Although it comes with problems, prototyping is a kind of effective project developing approach. The most important thing of it is that the customer and developer must accord with each other. The prototype is build only to define requirements, and should be abandoned (or partly abandoned) after that. By the way, in our country, GIS customers haven't clear idea about the functions and requirements of GIS, so Prototyping is used frequently in GIS development.

### 2.3 Rapid Application Develop, RAD

Governmental decision-making efficiency is sometime lower than expected, and once a decision is made, officials hope to see the fruit immediately regardless of the lack of fund; modern commercial must be adapted to information society and have the capability to deal with emergencies. When a GIS is to be accomplished in 6 months, RAD is the quickest approach.

To shorten the development process, RAD projects always apply former designed technology, method and tools, and some projects may be build up based on COM. There are many developing tools on the market supporting Rapid Application Developing, like Powersoft's Power Builder and Microsoft's Visual Basic. In a RAD project, GIS customers attach themselves with the project development more tightly than in other developing approaches. During the development process, customers spend much time on the design and implementation of the system with GIS developer.

RAD is characterized by short developing circle, and so developers of these kind of project have intense pressure. Everyday long-time work is a physical and spiritual test to them. If the RAD project uses many new technologies, it has high venture in technology and risks failure. On account of the secondary development technology of commercial GIS platform is very dependable, applied GIS developed based on commercial GIS platform usually have low venture, and this kind of GIS can use RAD approach.

### 2.4 Object-Oriented approach

We live in a world full of objects, and Object-Oriented approach is an abstract method modeling the world. It analyzes and resolve problem according to the way human thinking. Among the many advantages of Object-Oriented approach, object-reuse can develop software quickly. In addition, Object-Oriented software can be expanded and maintained easily because its structure is internal incompact catena.

For years, "Object-Oriented" indicated software-developing approach using Object-Oriented computer language (example, C++, Smalltalk). Today, Object-Oriented technology is applied to entire software engineering area, containing all the software engineering viewpoints and tools: Object-Oriented Analysis, Object-Oriented Design and Object-Oriented Computer Aid Software Engineering etc. In the 1900s object-technology and Object-Oriented software engineering are widely adopted by many software especially information system.

The popularity of object-tech produced many Object-Oriented analysis approaches, like famous Booch's approach (emphasizing operation and activity, being good at its definition to class and succession), Rumbaugh's approach (emphasizing modeling of data and objects), Jacobson's approach (Use-case-oriented) and Code/Yourdon's approach (a intelligible approach) etc. All these approaches introduce the stage of system analysis of a project, and have their Object-Oriented modeling symbol system, that is, modeling language. There was a longtime debate about which was the best one of these approaches, and the argument focus on the modeling language. It was not until 1995 did the three world famous object-tech experts walk together and produced Unified Modeling Language (UML). By the end of 1996, UML had possessed 85% of the Object-Oriented technology market. On 17 November 1997, OMG adopted UML as uniform modeling language of Object-Oriented technology. The designers of UML synthesized each of their system analysis process in 1998 and established a software developing approach named Unified Software

Kommentar [gxx2]: “继承”  
一词是这么说吗？

Development Process (USDP). It is a universal approach applicable to all kinds of software system as well as application area and development organize. The big project “MGIS” was the customer and beneficiary of USDP, which was developed by the 15<sup>th</sup> institute of the Department Of Information Industry and Surveying and Mapping Institute of PLA.

## 2. 5 USDP, Unified Software Development Process

Unified Software Development Process is a software project developing approach based on component and UML. Its main idea can be generalized into three points: Use-Case Driven, Architecture-Centric and Iterative and Incremental.

### 2. 5. 1 Use-Case Driven

Software exists because customers use it. The customers contain not only people but also other systems interact with the software. A Use-Case is a function to perform once interactivity between the software system and the customers. Use-Case models the system function from the view of the customer. So Use-Case construct the fundamental concept accepted by both customer and developer.

Use-Cases defined for the system are the foundation of the whole development. All of the Use-Cases compose Use-Case Model, which is the conclusion of requirement analysis. Use-Case Model takes the place of traditional functional criterion, and describes the whole system function. Functional criterion answers the question “what can the system do? ”, while the tactic of Use-Case is “what can the system do *for the customer*”. Use-Case drives the entire development process from end to the end (design, implement, test). Developers design a model and put it into practice to implement a use-case; testers build a testing use-case to comment if the components implementing the model implement the use-case correctly. Use-Case is like a “red line” through the whole work.

**Kommentar [gxx3]:** 原文中这一段不太容易理解, 不知道我理解得对不对?

Although Use-Case drives software engineering, the selection of Use-Case is not independent. Use-Case drives system structure and the latter influence the selection of the former. Both of them maturate gradually with the extension of the development life cycle.

### 2. 5. 2 Architecture-Centric

A building’s structure can be expressed from different aspects by the blueprint of landscape, perspective, cable system, vent system and piping etc. By the same means architecture of software system can be described from different views like system function, logic structure and physical distribution on the system platform. It is technically at risk to develop a complex software project without architecture or with a bad architecture.

Architecture is the system blueprint illustrating how it works. Architecture describes primary character of the system (like primary classes, main process, thread, permanence mechanism, communication mechanism, layer, subsystem, and port) while ignores the detail.

A lot of workers participate in the architecture-centric software project, and each of them interests in different facet of the software architecture. System analysts use architecture to organize and describe the requirements, and try to comprehend the technical limits and venture; customers can see the outline of what they have ordered; Project managers use architecture to arrange the work

**Kommentar [gxx4]:** “高层”是指人事上的还是技术上的?

of software development team; designers locate their design border after grasp the primary principles of the architecture; other development organizes learn how to interact with the system from architecture; software engineers use architecture to expand and reuse the system. Building blueprint has become the standard, which is intelligible to investor, designer, builder, and client. In order to communicate, review, comment and improve a software project's architecture, there should be a standard, unified method to express it.

### 2. 5. 3 Iterative and Incremental

In a Unified Software Development Process, the software is not submitted as a whole at the end of the project. It is separated into several "mini" projects to develop and submit one by one. The whole development is a series of repetitious periods, and each period contains four stages: Inception, Elaboration, Construction and Transition. Each of these stages is composed of several iterations (Figure 3), and each of the iterations will practise the process of analysis, design, implement and test.

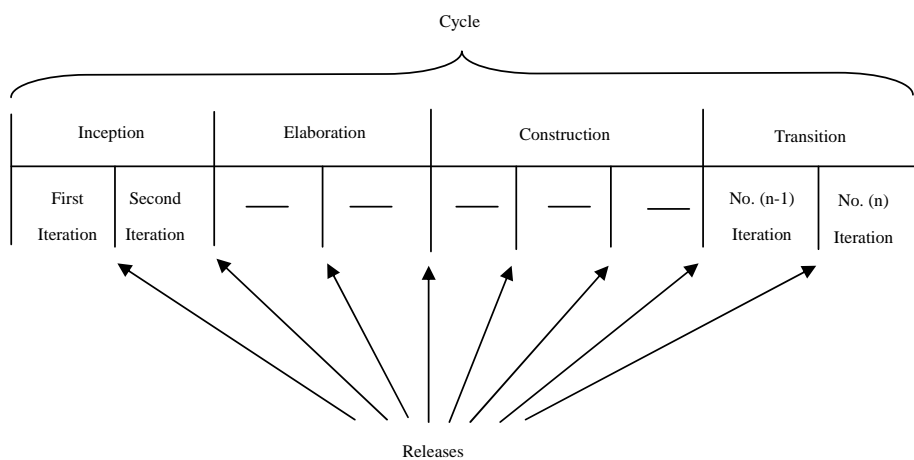


Figure 3. Stages and iterations of a cycle

### 3 Choose A Suitable Developing Approach For The GIS Project

Usually, choosing a developing approach is very subjective. It is influenced by many facts like customer requirements, personal (or the group's) favor, developing tools available and the fear of changing the existing developing approach etc. When the members of a developing organize get familiar with a developing approach, it's hard to change it like one's relief. So, to the project developing team or the software developing organize, the most familiar approach is the best one. If the project needs to adopt new technology widely (for example, Object-Oriented tech), or has changed the developing tools or DBMS, it's better to consider new developing approach.

Recent years, a trend of GIS is the change from traditional description of the relation between abstract point, line and area to the description and management of objects in the real world. Object-Oriented tech and the increasing practicality and maturation of Object-Oriented Spatial

Data Base support the change. With the advent of UML, which has become the standard modeling language, the developing approach based on the method of UML and adopting CASE tools has become the new leading GIS developing approach.

### References:

- 1 Philippe Krechten • The RATIONAL UNIFIED PROCESS: An Introduction • Addison Wesley Longman, Inc, 1998
- 2 Ivar Jacobson, Grady Booch, James Rumbaugh •The UNIFIED Software Development Process •Addison Wesley Longman, Inc, 1999