

A Software Engineering Approach For GIS Developing

Wu Sheng Wang Jiayao

(Surveying and Mapping Institute of PLA, Zhengzhou 450052)

Abstract This Paper introduced an object-oriented software engineering approach, unified software development process (USDP). Applying to the developing of Military Geographic Information System, it's ideas and modeling elements are detailed described. The ideas include use-case driven, architecture-centric, iterative and incremental. The modeling elements include workers, activities, artifacts and workflows.

Kommentar [gxx1]: Applied

Kommentar [gxx2]: are described in detail

Keywords Software Engineering, Unified Software Development Process

1 Introduction

In the early stage of computer development, a computer system's budget was mainly on hardware. While there were very strict engineering methods to control and ensure the quality of hardware, there were hardly any systematic approaches for software development, which was comparatively simple at that time. The software environment was of personality, just thinking that a programmer would get up at 2 o'clock to implement a deliberate algorithm. As to software design, it was only an illegible thought in his head. This thought and its implementation, of course, would not be put into any normative documents. In general, software development at that time was in a state of disorder, but as the early GIS was simply and there were few developers being concerned with it, the development of GIS would always succeed.

With the increasing of GIS and the expansion of GIS software library, programmers found themselves being more and more busy: correcting all the programs to kill the bugs; modifying the programs to meet the changes of the customers; adapting the software to the update of hardware. All these work called software maintenance. Resources put into software maintenance increased at striking ratio. The expenditure for software maintenance exceeded that for the original development.

Recent years, a trend of GIS is the change from traditional description of the abstract point, line and area to the description and management of objects in the real world. The increasing practicality and maturation of Object-Oriented tech just support the change. Today, Object-Oriented technology contains all the software engineering approaches and tools: Object-Oriented Analysis, Object-Oriented Design and Object-Oriented CASE etc. In the 1990s,

Object-Oriented tech and Object-Oriented software engineering were adopted by many software especially information system software. In this paper, the latest Object-Oriented software engineering approach——Unified Software Development Process will be introduced. The development of the project “Military GIS” exemplified USDP successfully, which was developed by the 15th institute of the Department of Information Industry and the Zhengzhou Institute of Surveying and Mapping.

2 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 aspects: Use-Case Driven, Architecture-Centric and Iterative and Incremental.

2.1 Use-Case Driven

Use-Cases defined for the system are the foundation of the whole development. The concept of Use-Case is like a “red thread” through the whole work, playing different roles in the four primary workflows: requirements analysis, design, implementation and test. Firstly, Use-Case model is the result of requirements analysis. In this stage, GIS Use-Cases model the system functions from the view of the users. So Use-Cases construct the fundamental concept accepted by both users and developers. Secondly, in the stage of analysis and design, Use-Cases are implemented by design model. In a design model, the implementation of GIS Use-cases is described by the interaction of objects. Thirdly, design model is the criterion in the system implementation stage, which is implemented by designing the class. And in the last stage, GIS Use-case is the foundation to discriminate between test instance and test process.

2.2 Architecture-Centric

Architecture is the system blueprint illustrating how it works. Architecture describes primary character of the system (major 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 understand the technical limits and venture; customers can see the outline of what they have ordered; Project managers use architecture to arrange the work of software development team; designers locate their design border after grasp the major principles of the architecture; other development organizations learn how to interact with the system; software engineers use architecture to expand and reuse the system. Building blueprint has become 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.

Kommentar [gxxh3]: “高层”
是指人事上的还是技术上的?

2.3 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, and each of the iterations will practise the process of analysis, design, implement and test.

Iterative increment distribute the technical venture to every increment, which reduce the whole system' technical venture to the lowest level so as to be controllable. Another advantage of the iterative increment is that it produces exercisable increment continuously, which is helpful to increase the developers' self-confidence and the interest to go on with the development.

3 Modeling Elements

USDP development process contains four primary modeling elements: workers, activities, artifacts (Figure 1) and workflows.

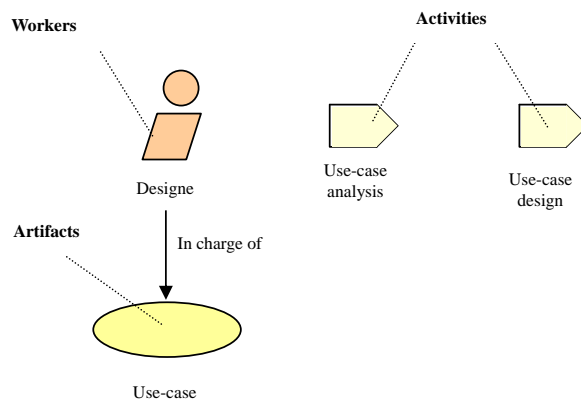


Figure 1. Workers, activities and artifacts

3.1 Workers

GIS is generally a big and complex software project, which needs many workers to work together. How to group them and make the group develop the software efficiently is vital to the project. An efficient development needs all the participants' effort, which contain project manager, system analysis, designer, programmer, supporter, documenter, customer and end user (Figure2).

3.1.1 Project Manager

To the project manager, project management is the management of the personnel, that is, how to recruit, to train them, pay them, bring up collectivism and enterprise culture. The project manager needn't to be a GIS expert, as his work is at the human activity, inspiring everyone to make the best of his technique, creativity and potential. A good project manager should make GIS developers devote themselves to the software development, without any extra worry having nothing to do with the project.

3.1.2 System analyst

In a GIS project, the system analysts (advanced system analysts are called Chief Designers or general designers) are experts in GIS analysis and design, leading and harmonizing the drafting of

requirements and demarcation of the system. They play different roles when communicate with the project manager, development team, supporters and customers.

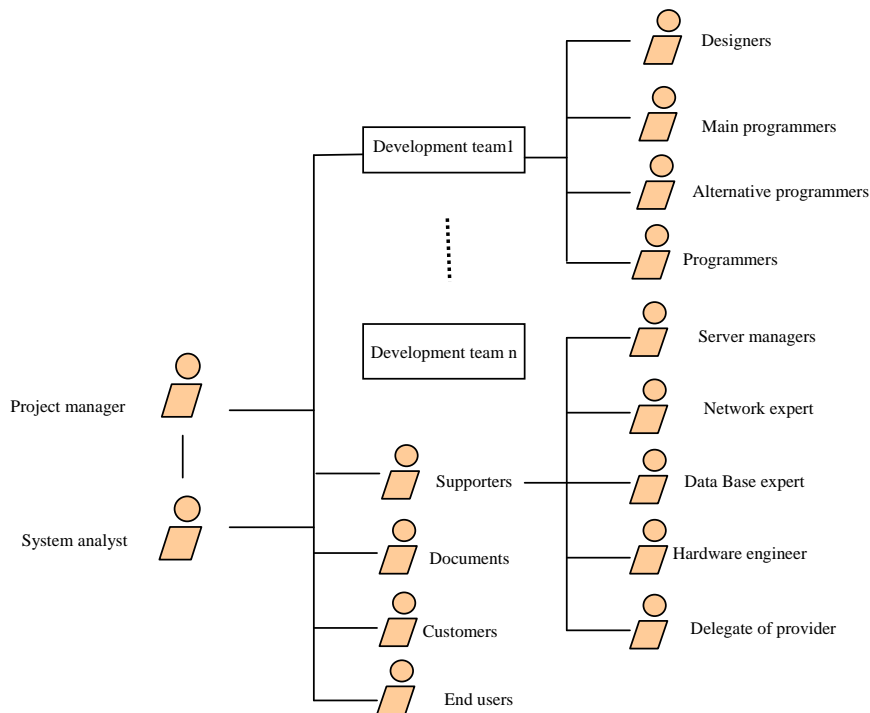


Figure 2. Structure of GIS development personnel

3.1.3 Software Development Team, Team Leader, Designer, Programmer

In a GIS project, Software Development Team is the least unit of the GIS development organization, which is composed of some developers and in charge of one or more GIS functional component or module.

Development teams vary in the organizational form and scale, depending on project management style, difficulty grade, modularization degree and the number of the developers etc. In some teams, there are not fixed team leaders; team members make decision together. This loose form is propitious to innovation in technique, on condition that the team members' initiative spirits are greatly activated. But this loose form may be inefficient if the team have the experience of developing a similar project. There should be a fixed leader in this condition, who takes charge in the communication with other teams together along with the project manager, planning and harmonizing the technical activities of the team members.

A GIS project development team is composed of designers, main programmer, main programmer in support and other programmers. Designers answer for the definition of the class function, operation, properties and the relation between classes, deciding how to modify the class to adapt the change of environment. The test designers also constitute test plan and test model; implement the test; evaluate the range, result and validity of the test. The main programmer must be able to

organize the developing work, experienced in programming and able to settle problems emerging during the development. Sometimes main programmer participates in the entire GIS project's design and program; arrange team development. Main programmer in support supports the main programmer's work; substitutes the chief programmer at lowest cost during the development. Other programmers develop and modify software according to the main programmer's plan.

Development team must set up an effective method to mediate the relation of participators of the project; ensure the communication mechanism of inter-team and inter-member, such as personal intercourse and regular meeting. The issue listed in the work plan and problems encountered should be incorporated into the meeting minutes as open states, and reviewed at next meeting to see whether they were closed. At the end of the project, all the open states should be closed.

Sometimes there are nonscheduled meetings to settle certain problems, but they should not take up too much time. On account of this, only the related people are asked to participate the meeting; otherwise, distribute meeting minutes to the related person—they can find what they are interested in.

The global information highway opened all the work in the world as a form never it was. Big projects like GIS always need participators at different location. Internet provides effective means (e-mail, BBS etc.) for the geographical distributed developers to have a meeting. Some difficulties can be resolved under the help of experts even out of the project. Software engineers can join the newsgroups relate to the technique he interested in; questions posted in the newsgroup can always be answered by the meddlesome around the world. An inquiry indicates that among all the mediation and communication, the most valuable is that with people out of the project.

As long as the team members pursue collectivism, common definition of success, common goal and culture, their composition of forces will produce great impetus and high software productivity, no matter how the GIS development team is organized.

3.1.4 Supporter

Experts in areas out of GIS, such as server manager, experts in network, communication, database, hardware engineer, and agent of CASE tools provider.

3.1.5 Documenter

Serving for several teams, a documenter maintain and control all the software configuration and security, which contain documents, source code, data etc; help to gather, collect and convert data; classify and index reusable software component.

3.1.6 Customers

People who explain the software requirements of the GIS project to be developed.

3.1.7 End user

Once the developed GIS come into a formal product, end users directly interact with the software.

3.2 Activities

In the glossary of Object-Oriented, a worker is a active object, and the worker's activity means operation to the object, following are some activities:

- Arrange one time iteration: executed by project manager
- Find use-cases and their executors: executed by system analyst
- Design reviewing: execute by reviewing designer
- Performance test: executed by performance tester

Activities can be carried out in steps. Following are three main steps:

- Thinking steps: comprehend task character, collect and check the input, design output.
- Performing steps: build and update the [artifacts](#).
- Reviewing steps: compare artifacts with criterion.

3.3 Artifacts

Artifacts are the information being produced, modified and used during software development; they are tangible product in the project development.

They may be sheer documents when exterior inputs of the project are simple descriptive information. More commonly, USDP don't encourage printing everything to paper, on the contrary, keeping them by artifacts-builder tools is the most effective and practical means. For example, deposit them in the design model of Rational Rose, in the project plan of Microsoft Project, in the project requirement of Requisite Pro etc. Of course, these tools can create documents automatically and quickly whenever they are in need.

USDP classifies artifacts into five kinds, or information set, these artifacts progress with the iterative development cycle.

- Management set, artifacts relate with the management of GIS project, for example, development plan, state evaluation.
- Requirement set, artifacts relate with GIS system definition, for example, business model, use-case model.
- Design set, contains system description of GIS. For example, design model, description of system frame, test model.
- Implementation set, contains source code, executable file etc.
- Deployment set, contains setup manual, user's manual and training material etc.

3.4 Workflows

Only the workers, activities, and artifacts can't constitute the whole GIS development process. We need a series of operation to describe the interactive relation of the workers, to produce the artifacts we need. We call this series of operation workflows. USDP's four primary workflows are [business model](#), requirement analysis, system analysis and design, implementation, test and

deployment.

4 Conclusion

In our time, computer and information technology's progress ratio is quite beyond the evolution ratio of human brain and human education. USDP provide us an important technique and means to develop GIS, but the most important fact of a successful software project is not the tools but the man. In our country, the recognition of software developer with creative talent sometimes doesn't accord with the manager's declaration. This may be correlated closely with the long time low cost of non-creative labor. When they found themselves couldn't cope with the increasing and more complex software because they had lost the genius, some organizations realized: it's important to attract, train, encourage, keep genius and provide them nice work environment. Human beings are not only the contriver of the technique and tools, but also the user of them.

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