

# On Practice of Big Software Designing

Yang Lin

School of Information Technology, Beijing Normal University (Zhuhai), Guangdong 519085  
Email: yanglin@gmail.com

Sun Wei

School of Information Technology, Beijing Normal University (Zhuhai), Guangdong 519085  
Email: angela--wei@163.com

Zhao Chi-long

School of Information Technology, Beijing Normal University (Zhuhai), Guangdong 519085  
Email: zhaochilong@sina.com

Tu Hong lei

School of Information Technology, Beijing Normal University (Zhuhai), Guangdong 519085  
Email: tuhonglei@bnu.edu.cn

**Abstract**—In this article, we make a comparison between the development of software engineering and industrial engineering, big software designing and some new thinking about the development of software are proposed. Then we promote a “six-oriented practical theory” which was extracted from the practical experience and could serve as a start point of software development practice. The theory not only unravels the relationship between the software engineering and industrial engineering, but also will direct the practice of software development in various related fields.

**Index Terms**—big software designing, methodology, practical theory, software engineering, engineering, software craftsmanship.

## I. THINKING ABOUT THE PROCESS OF SOFTWARE DEVELOPMENT

Summer in 2007, The financial tsunami caused by the U.S. subprime mortgage crisis sweeping the globe, it forced the world to re-examine financial markets and financial regulation. Faced with it, the new system of global economic governance will show more flexibility and diversity. As a software industry practitioner, we pay more attention to the fact that the new economic pattern will produce a far-reaching impact on new industrial pattern and new software industrial pattern. Now we also need timely and profound reflection of the development of software industry.

### A. The proposed of software engineering

The term software engineering first appeared in the 1968 NATO Software Engineering Conference and was meant to provoke thought regarding the current "software crisis"[1] at the time. Since then, it has continued as a profession and field of study dedicated to creating software that is of higher quality, more affordable,

maintainable, and quicker to build.[2] Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software.[3] Since the field is still relatively young compared to its sister fields of industrial engineering, there is still much work and debate around what software engineering actually is, and if it deserves the title engineering.

### B The proposed of software engineering methodology

The software engineering methodology also known as software development methodology or system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system.[4]

The framework of a software development methodology consists of:

- a) A software development philosophy, with the approach or approaches of the software development process
- b) Multiple tools, models and methods, to assist in the software development process.

A serial software development methodology which include Structured Programming[5], Object-Oriented Programming[6], Rapid Application Development[7], Extreme Programming [8]and Rational Unified Process[9] and a serial software development approaches which include Waterfall[10], Prototyping[10], Incremental[10], Spiral[10] and Rapid Application Development [10]consist of the collection of software engineering methodology. The collection is growing, because the methodology which meets the need of practice and software engineering have not yet found.

*C The proposed of Software craftsmanship*

After active practice of software engineering, the shadow of software crisis seems to continue with us. In 2001 Pete McBreen[11] suggested that software developers need not see themselves as part of the engineering tradition and that a different metaphor would be more suitable. There is a general discussion these days over software engineering and software craftsmanship, we should be thinking of the dialectical view. The proposed of software craftsmanship force us to re-examine the nature of software development and the relationships with software engineering.

*D Summary and thinking*

Three typical stages of software development that has been discussed above, we focus our energy on the debate about the software engineering and software craftsmanship. Now we make a general comparison on the software engineering and industrial engineering. Which help us understand the process of the software industry development, and provide some new directions of thinking. See the Table1 and Table2

Table 1  
The evolution of engineering

Industrial engineering	
Predisposing factor	Development process
Ancestral skills	Folk artists
Meet expanded requirements	Workshop
Meet the greater requirements and Reduce costs	Industrialization
Further reduce costs	Industry Standardization
Industry Standardization and Low-cost to meet the individual requirements	Production of industrial standardization and production of individual requirements

Table2  
The evolution of software engineering

Software engineering	
Predisposing factor	Predisposing factor
Used in scientific computing	Scientific research
Used in scientific computing and Military affairs	Project with personal heroism
Used in scientific computing, Military affairs and aerospace	Software engineering
Used in scientific computing, Military affairs, aerospace and General industry	Software engineering Standardization
Used in scientific computing, Military affairs, aerospace, and almost all industry	?

Topic 1: Extreme increase in the number of demand led to the industrial revolution that the turning point in the development of the industry, the impact of a wide range and a lot of demand for industries is one of its main

features. However the causes of the software crisis were linked to the overall complexity of the software process. And question concentrated in the military and aerospace applications: Whether is the engineering the right way to solve the problem?

Topic2: The industrial sector has gone through such a process, folk artists, workshop, industrialization, industry standardization and production of industrial standardization and individual requirements. Whether the software industry follows the same process of development? After the standardization of software engineering, how to achieve mass production? How to meet the individual requirements, and reduce costs, and achieve small batch production?

Topic3: The basic industry of the production process including design, production, sale and maintenance. Typical software development process including: requirement analysis, design, coding, testing, release, sale and maintenance. From the Table 3 we can see the similarities between the engineer and software engineer. But we must take into account the fact that software development process is only equivalent to the design phase of the industrial sector. From this point of view software development process is a big design. If we recognized this fact, we must also accept the full treatment of software development with the engineering is not feasible.

Table 3  
Comparison of industry processes and the software development process

Industry processes	software development process
design	requirement analysis
	design
	coding
	testing
	release
Sale	sale
maintenance	maintenance

What is Design? Dino Dini states that the design process can be defined as "The management of constraints"[12]. He identifies two kinds of constraint, negotiable and non-negotiable. The first step in the design process is the identification, classification and selection of constraints. The process of design then proceeds from here by manipulating design variables so as to satisfy the non-negotiable constraints and optimizing those which are negotiable. It is possible for a set of non-negotiable constraints to be in conflict resulting in a design with no solution; in this case the non-negotiable constraints must be revised. For example, take the design of a chair. A chair must support a certain weight to be useful, and this is a non-negotiable constraint. The cost of producing the chair might be another. The choice of materials and the aesthetic qualities of the chair might be negotiable.

Draw on this idea, we can analyze some major two types of constraint in software development. For non-negotiable constraint, we can use software engineering

methods to deal with, for negotiable constraint, we can draw on the idea of software craftsmanship to allow designers to free to play. In order to achieve this goal that designers need to constantly improve their skill level. For example, High Cohesion and Low Coupling is an abstract design methodology, which is non-negotiable or negotiable constraint?

## II. SOFTWARE ENGINEERING METHODOLOGY OF USE

The method of software development is different from the model of software development. The model refers to the development pathway that the whole process should follow up with, such as the water fall model, incremental model, prototype model, RUP (Rational Unified Process) and iterative model. These models all have their own specific cycles.

The method, on the other hand, means the development tenets, technologies, description methods and tools that are used in the software development process. The set of methods used in the software development forms the software engineering methodology.

Until now, how many methods have been employed in the software engineering and what are their relationships? These are the problems we will discuss in the paper.

### A *Meta-data Oriented Method*

The meta-data oriented (MDO) method we mentioned here is not the traditional dataflow oriented method or the Jackson method, which is data structure oriented. The MDO method refers to an approach which is using meta-data as the main guide to go through the process of requirement analysis, design, implementation, test, maintenance and management of software. [13][14]

Meta-data is the data about data, which refers to the data that organizes, defines and manages the original data. For example: the entity and attribute names used in the database conceptual design, the table and column names in the database physical design, etc, which are all the meta data. A specific instance or a row in the database table is not a meta-data, but an object or a record, which are organized by the meta-data. The programming languages offered by the Relational Database Management Systems are powerful programming tools on the data stored in database tables. The typical examples of programs on data are stored procedures and triggers. Developer 2000 in the Oracle database management system is a meta-data oriented programming tool. Oracle Designer combined with Developer 2000 consist a complete development environment for information systems based on meta-data.

To summarize the main points of meta-data oriented method, we get:

- a) The data is positioned as the center of the information system and is organized by the meta-data. Information system is a system that is handling the data input, process, transport, query and output.
- b) As long as its business direction and content are steady, the meta-data of the enterprise is stable and the data model based on the meta-data is unchanged.

- c) The processing methods to the meta-data are changeable and flexible. The steady meta-data combining with the flexible processing methods is a rational method.
- d) The core of the information system is the data model. Data model includes the Conceptual Data Model (CDM) and the Physical Data Model (PDM). The representation of the Data Model is E-R diagrams, which could be designed via CASE tool. The tools such as Power Designer, Oracle Designer or ER win not only have the function of forward design but also have the function of backward analysis, which could facilitate the fast prototyping method.
- e) The implementation (coding) of the system is to use the meta-data oriented programming in the database layer and employ the object-oriented programming in other layers.
- f) The users (clients) should involve in the whole process of analysis, design, implementation and maintenance of the information system from the beginning to the end.

Meta-data oriented method includes meta-data oriented requirement analysis, meta-data oriented design, meta-data oriented programming, meta-data oriented test, meta-data oriented maintenance and meta-data oriented management.

Meta-data oriented requirement analysis is to find out all the meta-data in the information system to fully meet the requirement of the data process, query, transportation and output.

Meta-data oriented design is to acquire the meta-data through requirement analysis and use CASE tool to design the conceptual data model CDM and physical data model (PDM);

Meta-data oriented programming is to build up the database tables and views based on the functional requirement and business rules. Then use the database programming language to program the stored procedures and the triggers.

Meta-data oriented test is to test all the functions and performance of the system after the data is loaded in the database and the stored procedures and triggers have been implemented.

Meta-data oriented maintenance is to clone, backup and recover the database records.

Meta-data oriented management is to adjust the table or view structures if it is necessary.

In the recent twenty years, meta-data oriented method became the fundamental approach for building up the database system, data warehouse and information systems.

Chinese Yantai Harbor Information System was comprised with several subsystems which were Ship Scheduling, Goods Transportation, Container Management, Device Management, Human Resource Management, etc. [15] The Yantai Eastern China Electronic Coop. started from this small application and steadily grown up to be a famous Harbor ERP system developer and business integrator. The secret of success of this company is that it has been persisting on the meta-data method and used the Oracle relational database

management system and the CASE tools. The tools include design tools such as Power Designer, Developer 2000 and the basic business platform Justep X3.

### *B Procedure-Oriented Method*

Procedure-oriented method also called structure oriented or structured method.

Its main thought is to capture the flow of execution. The programs stress on some notions such as "single entry point and single exit point". It also started to propagate some good design principles, such as: "top to down design, step by step refinement from coarse to fine granularity". The merit of this method is that it is simple and easy for learning. But it only puts emphasis on the procedure information and ignores the hierarchical or aspect information and their relationships of the real world. Its only relying on the temporal information (which could be represented by sequential, conditional and loop structures) makes it an insufficient tool for describing a complicated and randomly organized information world. Solely using this method will lead to a chaotic situation in design, development and maintenance of software.

Procedure-oriented method started in Sixties of 20th century and matured in seventies and became popular in eighties. It began to give up its dominant standing with the appearance of object-oriented method and the meta-data oriented method, which are more powerful in the description capability. Currently the method is mainly used in the procedure programming situations such as object methods, scientific calculation, real-time tracking and control systems.

The procedure-oriented method has some good applications in the military and aviation real-time tracking and monitoring systems. For example, the systems that capturing, tracking and predicting the course of satellites; the missile defendant systems that tracing the course of missiles from the enemy side, etc. These systems are all mainly using the procedure-oriented method for design and implementation, which is quite appropriate for automatic controlling and tracking systems.

### *C Object-Oriented Method*

This method includes object-oriented (OO) analysis, OO design, OO programming, OO test, OO maintenance and management.

The essential point of OO method is to map the things in the real world (problem domain) into objects. OO Design and programming is to abstract the classes ( which represents the common properties of a class of objects ) from the concrete objects and running the OO program is, on the other hand, running the objects created from classes. This method has emphasized on the attributes of the objects and the hierarchical relationships of classes. It incorporated the procedure as the behaviors of the objects. Therefore it is a superset of the procedure-oriented method and has more powerful description capability. The advantage of OO method is that it makes programs easy for maintenance and extensible.

With the appearance of large-scaled software, OO is widely used in developments. In web programming, OO method is broadly used in client-side and server-side program, which enables users to browse various pages, input data and interact with the servers. Besides, this method becomes very popular in new software fields such as game design, implementation and maintenance. The rapid development of window operating systems and the Internet has brought and will bring great opportunities for OO method.

### *D Formalized Method*

Formalized method is using the formal semantic definitions and logic deductions to describe world, which is base on discrete mathematics and formal languages. This method, which may have high theoretical value, is currently still in the research field and not being commonly used in industry. Its potential application needs to be proved in practice, which is not a main point being discussed here.

### *E Summary of Software Methodology*

We firstly discussed the meta-data oriented method in detail and then summarized that procedure-oriented, object-oriented, meta-data oriented, and formalization methods formed a relatively complete software engineering methodology. We mainly researched on meta-data oriented method, procedure-oriented method and object-oriented method and briefly mentioned the formalized method. In the large-scaled B/S based information systems, the relationship between these methods could be summed up as: meta-data oriented method is mainly used in database design and implementation; object-oriented method is mainly applied in the system implementation other than the database layer; procedure-oriented method is used in the function or behavior design and implementation; formalized method is used to prove the correctness of some core programs. From this analysis, we could draw a conclusion that the view of "procedure-oriented method is traditional and object-oriented method is modern" is quite shallow or incomplete.

All these methods are not independent, disconnected or conflicting from each other, but are interconnected, correlated and compensated for each other. They together consist of a multi-polar world of software methodology.

## III. CASE STUDY

TMS (Teaching Management System) is an information management system for the teaching process management in the universities. Purpose is to realize informationization of teaching process and be convenient for the exchange between the teachers and the students

The system adopt the RIA frame .The client is FLEX2 technology, which is beneficial in raising speed of response and the consumer experience



Figure 1. System architecture of TMS

TMS has three roles: administrator, teacher and student

TMS has five chief modules

- a) Course resource management
- b) Course assignment management
- c) Course management
- d) System management
- e) Forum

*A Main Functions of Teacher roles*

Upload electronic resource according to teaching requirement, e.g. courseware and source code.

Arrange assignment and fill in related descriptions and requirements.

Have full control on individual assignment and group assignment.

Correct, grade and give comment on students' assignment.

Show the statistics of assignment according to various conditions, e.g. student, assignment and class.

Manage student, divide into classes or divide into groups.

*B Main Functions of Student roles*

Download course resource to prepare for class.

Submit assignment with the form of attachment or plain text.

Check the mark and comment if teacher has already marked assignment.

*C Main Functions of Administrator*

Course management: managing courses, managing different parallel classes and user privilege.

Organization management: managing information of academy, class and student.

*D Main Functions of Forum*

The forum is the platform that teacher and student exchange ideas.

IV. PRACTICE THEORY OF BIG SOFTWARE DESIGNING

From the experience of information system construction, we compared the properties of various methods and figured out a practical theory, which is “personal-oriented skill improving, flow-oriented analysis, metadata-oriented design, object-oriented implementation, functionality-oriented testing and procedure-oriented management” and named it as “six-oriented practical theory”.

*A. Personal-oriented skill improving*

In this paper, we make a point that the software development process is a big design process. To ensure the quality of the software, we learn from thinking of software craftsmanship, the designers must constantly improve their skill level. Here refers to the designer

include all the members of the whole software life cycle, not only refers to software programmer. Through practice, we found that this method requires a supporting environment of software industry.

First, Educational institution’s educational philosophy is the most important aspect, for the student who will join into IT industry. A conviction must be established that Individual skills is a process of gradual increase and gradual rich. Some people say the IT work is only suitable for young people, such mistake idea need to be subverted by educational institution.

Second, Practitioners need to specify a suitable plan to upgrade their own skill.

Third, Project managers need provide flexible management and Differential treatment for design and engineering factors in the process of software development

The last point, leading organizations of Software industry should act as a role to supervise and promote Practitioner’s Skills improving.

*B Flow-oriented Analysis*

Flow-oriented analysis is an analysis method mainly concerning on the workflow and the data flow of the system.

At the requirement analysis phase, analysts should analyze the business workflow and the flow of information and capitals. Only after a thorough analysis on these flows, could an efficient system business model and functional model (including the module interfaces and performance model) being built up. Intrinsically computers only could recognize the binary data flows and the flows obtained from the analysis stage mentioned above will ultimately being projected into the main data flows of the information system. See the Figure 2

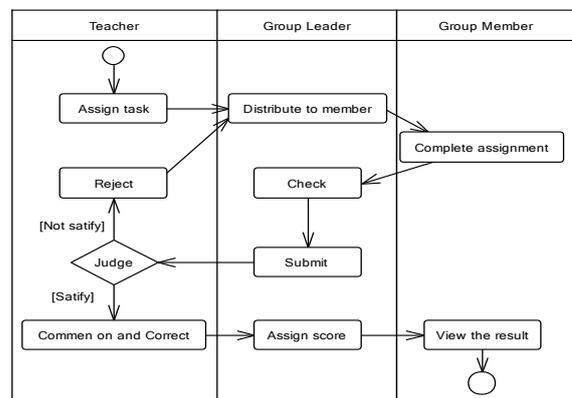


Figure 2. Business flow of assignment teaching management system

*C Meta-data Oriented Design*

Meta-data oriented design is to design the meta-data based on the flow analysis and the rough data collected from the analysis phase.

At the data design phase, analyst should use the meta-data oriented method to accomplish the task. Their main work is to build up the data model, which includes the Conceptual Data Model, the Physical Data Model, and

the processing routines such as stored procedures and triggers which stand for the business rules. The meta-data should be fully studied and analyzed, which is a prerequisite for an excellent data model (including

appropriate data organization, efficient data query and storage). In a B/S structured system, meta-data oriented design is mainly used in the database design.

See the Figure3 and Figure4.

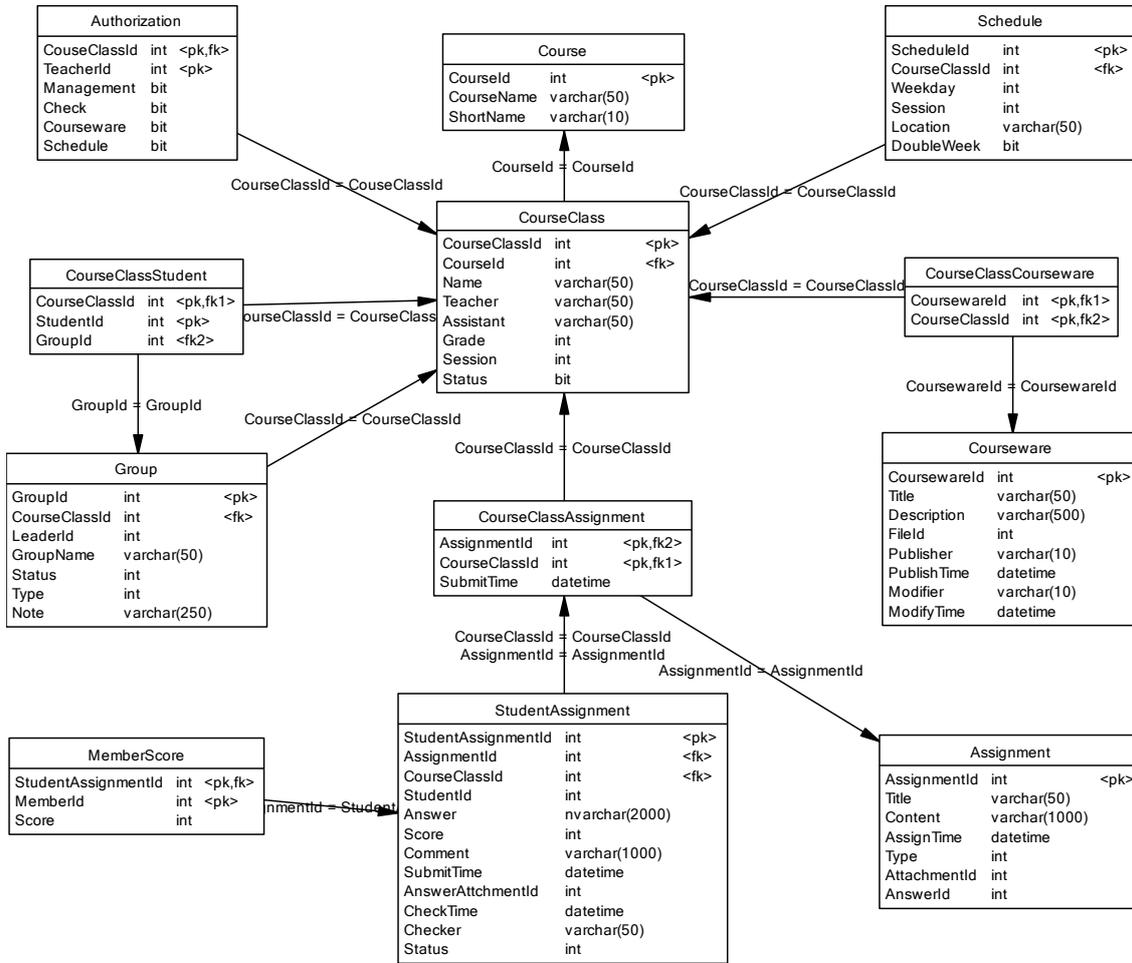


Figure 3 Physical data model of teaching management system.

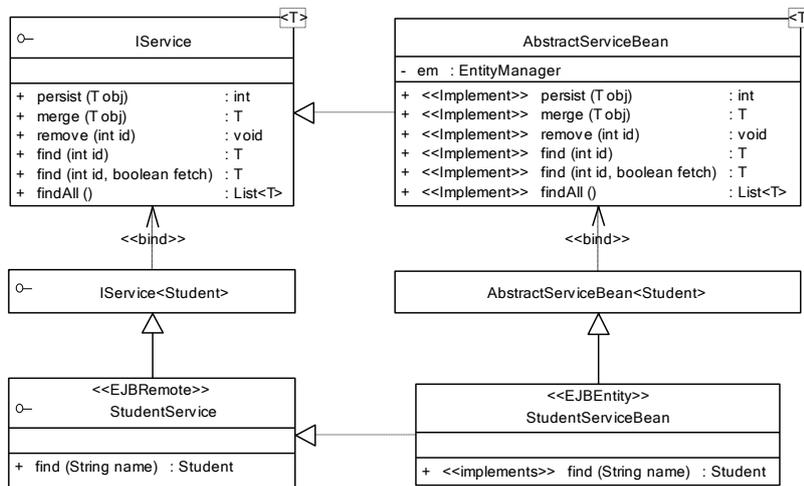


Figure 4 Persist service design of teaching management system.

*D Object-oriented Implementation*

Object-oriented implementation is a method that emphasizes on the use of object-oriented method in detail design and coding.

Currently most popular programming languages are object-oriented. Many matured software companies usually use the foundational class libraries provided by the language developers to build up their own business class libraries on behalf of their individual requirements. This work is very important for the software industry since the components built and accumulated by the vendors has already shaped a good develop environment for the developers and will ultimately form a well-organized, shared and efficient resource library serving the related industry. Essentially, detailed design and programming is to use the components that are reusable to build the system business model and functional model (including interface model and the performance model). Only after a good understanding of these three models (business model, functional model and data model) could we achieve a high quality design and coding. See the Figure5

*E Functionality-oriented Testing*

Functionality-oriented test means that the unit test, integrated test, Alpha test and Beta test should be accomplished by focusing on the system functionalities.[17]

Functionality-oriented test is also called black-box test. With the development of the fourth generation of programming languages and the technology of component, this test method is being used more and more broadly. The white-box test method (such as execution path test) will be limited within the group of developers. Therefore the testers should mainly grasp the methods of functional-oriented test. They need to design the test cases and finish the functional test based on the system functional models according to the requirement analysis.

*F Procedure-oriented Management*

Procedure-oriented management means the efficient management and control policies should be enforced to supervise the lifecycle of the software production.

The improvement of the software quality is depending on the improvement of production procedure of the software companies. CMMI (Capability Maturity Model Integration),ISO 9000, and the enterprise culture of Microsoft all advocate the improvement of the companies competence on behalf of the lifecycle of the software production.

The procedure-oriented improvement of the software companies focuses on procedures, both on the production procedures and the management procedures. The perceptible, controllable and optimized procedures are the guarantees for the high-quality software products. Quality is coming from procedure and procedure needs constant improvement. Continual improvement is the fundamental doctrine of CMMI.[18]

```

package tms.security
{
    import mx.core.Application;
    import mx.collections.ArrayCollection;
    import flash.events.EventDispatcher;
    import tms.config.AppConfig;

    public class LoginStatus extends EventDispatcher
    {
        public var userId : int;
        [Bindable]
        public var userName : String;

        public var authCode : String;

        public var role : int;

        public function getClientView(isDebug:Boolean) : String{
            if(isDebug){
                if(role == 3)
                    return "StudentClient-debug.swf";
                else if(role == 2)
                    return "TeacherClient-debug.swf";
                else if(role == 1)
                    return "AdminClient-debug.swf";
                else
                    return "";
            } else {
                if(role == 3)
                    return "StudentClient.swf";
                else if(role == 2)
                    return "TeacherClient.swf";
                else if(role == 1)
                    return "AdminClient.swf";
                else
                    return "";
            }
        }

        override public function toString() : String {
            return userId + "," + userName + "," + authCode;
        }

        public static function get current() : LoginStatus {
            return mx.core.Application.application.loginStatus;
        }
    }
}
    
```

Figure 5 a small piece of code

*G Summary on Practical Theory of big software designing*

The “six-oriented practical theory” is rooted from the long time practice of the software companies. It is a flexible application of the software methodology in the practice. It will lead the software companies to develop information systems.

For the other software other than the information systems, as long as the meta-data concept is appropriately incorporated, which guaranties that the meta-data is the data organizes the original data, the “six-oriented practical theory” is still applicable in those systems.[18]

V. CONCLUSION

In this article, we make a comparison about the development of software engineering and engineering, big software designing and some new thinking about the development of software are proposed. Then we promote a “six-oriented practical theory” which was extracted from the practical experience and could serve as a start point of software development practice.

ACKNOWLEDGMENT

We first thank those of you who have supported this paper by reading and modifying it in previous editions. We are honored.

With each new edition, the list of those who have contributed to this paper as reviewers and consultants

grows. Now this list includes Cai Xiao li and Sun Yuan. To these individuals we give our sincere thanks.

#### REFERENCES

- [1] Software engineering: Report of a conference sponsored by the NATO Science Committee. 1968
- [2] [http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering)
- [3] Guide to the Software Engineering Body of Knowledge - 2004 Version.
- [4] [http://en.wikipedia.org/wiki/Software\\_engineering\\_methodology](http://en.wikipedia.org/wiki/Software_engineering_methodology)
- [5] [http://en.wikipedia.org/wiki/Structured\\_programming](http://en.wikipedia.org/wiki/Structured_programming)
- [6] Armstrong, The Quarks of Object-Oriented Development. In descending order of popularity, the "quarks" are: Inheritance, Object, Class, Encapsulation, Method, Message Passing, Polymorphism, Abstraction
- [7] Whitten, Jeffrey L.; Lonnie D. Bentley, Kevin C. Dittman. (2004). Systems Analysis and Design Methods. 6th edition. ISBN 025619906X.
- [8] "Extreme Programming" (lecture paper), USFCA.edu, webpage: USFCA-edu-601-lecture.
- [9] Ivar Jacobson, Grady Booch, and James Rumbaugh (1999). The Unified Software Development Process
- [10] SELECTING A DEVELOPMENT APPROACH. Revalidated: March 27, 2008. Retrieved 27 Oct 2008.
- [11] Pete McBreen, Software craftsmanship, 2001
- [12] <http://en.wikipedia.org/wiki/Design>
- [13] Zhao Chilong: Practical Software Engineering, Beijing: Electronic Industry Publishing House, March 2003
- [14] Zhao Chilong, Yang Lin, Sun Wei: Practical Software Engineering (2nd Edition), Beijing: Electronic Industry Publishing House, September 2006
- [15] Zhao Chilong: Large-scaled MIS Should Emphasize on Database Design, Beijing: Computer World, 1994: Vol. 12
- [16] Zhao Chilong, Zhang Anbo: Reference Guide for Information System Construction, Beijing: Application of Computer Systems, 1996, Vol. 6
- [17] Ron Patton. Software Testing, Beijing: Mechanics Industry Publishing House, February 2002
- [18] Lin Rui, Peng Guoming: CMMI and Integrated Management on Software Research and Development, Beijing: Electronic Industry Publishing House, January 2008

**Yang Lin**, male, senior engineer, main research field is software engineering and computer numerical control.

**Sun Wei**, female, main research field is software testing, and Unix/Linux OS.

**Zhao Chilong**, male, senior engineer, main research area includes software engineering, software procedure improvement, database design and literature.

**Tu Honglei**, male, senior engineer, main research field is Object-Oriented Design, Pattern Recognition.