

# SUPPORT STRUCTURE OF KNOWLEDGE MANAGEMENT IN SOFTWARE PROCESS IMPROVEMENT

Wan Jiangping<sup>#</sup>, Yang Jianmei<sup>#</sup> and Han Huiyuan<sup>\*</sup>

<sup>#</sup>SCUT Guangzhou, Guangdong. 510640, P.R.China, <sup>\*</sup>Huazhong University of Science & Technology, Wuhan Hubei 430074, P.R.China

**Abstract:** This paper reviews software process improvement, science of complexity and knowledge technology and software knowledge technology. The software process improvement knowledge support structure, software engineering cognition support structure and knowledge-based enterprise model for software process improvement are illustrated. We reach conclusion that software production could systematize in term of enterprise model through conceptualization, and standardize in term of software process management through interactive cognitively process, structure in term of software engineering project through organizational learning, finally realize its business value through engineering approach. It is also explained how to do that by examples.

**Key words:** Structure, Cognition, Software Process, Software Engineering, Science of Complex, Enterprise Model, Knowledge Technology, Object Technology, Knowledge Management, Knowledge Portal.

## 1. INTRODUCTION

Knowledge Management (KM) is a kind of new thought tide on management. It demands all the employees to participate in and the company to build knowledge friendly culture. It emphasizes the tight combination between computer information system and employee's cognitive so as to form knowledge network with creative & cognitive capacity in companies. All members of learning organization work heart and soul and should learn

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35604-4\\_20](https://doi.org/10.1007/978-0-387-35604-4_20)

continuously, and experience life meaning during working, and create superior alternatives for themselves and future <sup>[5,6,12,13]</sup>.

The software process is the set of tools, method, and practices we use to produce a software product. The objectives of Software Process Improvement (SPI) are to produce products according to plan while simultaneously improving an organizational capability to produce better products. It is clear that a fully effective software process must consider the relationships of all the required tasks, the tools and methods used, and the skill, training, and motivation of the people involved <sup>[3,6,16,17]</sup>.

We argue that Software Process Capability Maturity Model (SW-CMM) is a framework of KM; every process in a software organization can be viewed as knowledge process in the framework <sup>[3,6]</sup>.

The software products are generally more complex than other engineering products and the software discipline is not grounded in natural science. As an artifact of human ingenuity, software does not rest on a stable foundation of physical principles <sup>[3]</sup>.

Prof. Cheng argues that management always depends a structure and evolves to an adaptable process <sup>[16]</sup>.

How to deal with the complex (chaos)? John Briggs and F.David Peat argue that the chaos is evolving from theory of science to cultural metaphor and give seven life lessons of change <sup>[14]</sup>.

In our understanding, KM in SPI also depends a knowledge support structure and evolves to an interactively cognitive process, which is a continuously organizational learning process.

We argue that Science of Complexity might help us to know why software process is so complex <sup>[4,7,8]</sup>, and the knowledge of Software Engineering (SE) could help us in SPI more effectively and efficiently <sup>[2,10,11]</sup>.

In this paper, we try to deal SPI with KM, including (1) SPI knowledge support structure; (2) Knowledge-based Software Enterprise Model for SPI; (3) KM in software organization.

## **2. SCIENCE OF COMPLEXITY**

### **2.1 Science of Complexity**

J.N.Warfield argues that normal problems involve local logic or occasionally intermediate logic, but complex problems involve deep logic. Since deep logic is generally absent from representations, or if being embedded in thicket-like prose often masks present, a consequence often is

under conceptualization and under documentation, as well as poor communication<sup>[8]</sup>.

A science of complexity integrates all of material in order to show both the theory of complexity and the empirical evidence that has been accumulated to show the validity of the theory. There are twenty laws of complexity science to guide how to apply it in organization<sup>[4]</sup>.

A principal outcome of the Science of Complexity is called The Work Program of Complexity (WPOC). This Work Program consists of two main steps: (1) Discovery; (2) Resolution. Discovery consists of description and diagnosis. Resolution consists of design and implementation. The necessity of these various activities, and the necessary products of those activities lead us to understand why SPI with KM is done in such the way that it is done.

## **2.2 The Categories of Mindbugs**

J.N.Warfield have tried to identify and name each distinctive origin of one or more behaviorally related symptoms (as "mindbugs" to bring the language in line with contemporary computer languages). So far, twenty-five mindbugs have been identified. These are envisaged as falling within four categories:

Mindbugs of Minsinterpretation: those where concepts are misconstrued or misattributed, because of faulty interpretation, Type M.

Mindbugs of Clanthink: those where concepts are very widely perceived to be correct, but which are demonstrably incorrect, Type C.

Mindbugs of Habit: those, which involve ingrained behavior, evinced with essentially no conscious thought, Type H.

Mindbugs of Error: just plain mistakes, Type E.

A fifth category that is under consideration has been designated as "Mindbugs of Specific Human Shortcomings". This category is based on a hypothesis that there may be something inherent in people as people that cause mindbugs, which can never be corrected.

## **3. KNOWLEDGE TECHNOLOGY**

### **3.1 Knowledge and Knowledge Technology**

Prof. Wang prefers the definition: knowledge is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and

information. It originates and is applied in the minds of knowers. In organizations, it often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices, and norm<sup>[5]</sup>.

Knowledge Technology (KT) is the discipline about the methodologies, methods, procedures and tools of knowledge identification, acquisition, storage, processing, dissemination, and creation for the purpose of adding value to human economic and social activities. The object of KT is not only the explicit, but also tacit knowledge and their interrelations. The approaches are not only technique-oriented, but also human-behavior-oriented.

In his View, KM refers primarily to the knowledge cycle that includes the capture analysis and communication of knowledge within an organization. It also relates to problem of searching and finding useful information.

### **3.2 Software Engineering Body of Knowledge**

The IEEE Computer Society defines SE as: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software; (2) The study of approach as in (1) <sup>[19]</sup>.

The IEEE Computer Society and the Association for Computing Machinery are working on a joint project to develop a guide to the Software Engineering Body of Knowledge (SWEBOK). Articulating a body of knowledge is an essential step toward developing a profession, because it represents a broad consensus regarding the contents of the discipline. It is also important to identify what disciplines share a boundary and often a common intersection with SE <sup>[2,10]</sup>.

### **3.3 Object Technology**

Object Technology (OT) has outstanding merits as following <sup>[25,26]</sup>: (1) to conform to the habit of human thought; (2) better stabilization of system which is developed by means of OT; (3) better reusable of system which is developed by means of OT; (4) better maintenance of system which is developed by means of OT.

The foundation of OT is type theory<sup>[26]</sup>. People are accustomed to thinking in terms of: (1) Object and Attributes; (2) Wholes and Parts; (3) Group and Member.

Now most of distributing network computing model, development environment, communication protocol and standards etc. are based on OT<sup>[26,27]</sup>. For examples, three basic technologies of Web Services (WS):

Simple Object Access Protocol (SOAP), Universal Description, Discovery, Integration (UDDI), and Web Services Description Language (WSDL) etc. Java promotes distributing object computing and software reuse.

The Unified Modeling Language (UML)<sup>[24]</sup> is an object-oriented language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. M.Folwer argues that The UML is called a modeling language, not a method. Most methods consist, at least in principle, of both a modeling language and a process. The **modeling language** is the (mainly graphical) notation that methods use to express designs. The **process** is their advice on what steps to take in doing a design.

The Rational Unified Process (RUP)<sup>[23]</sup> captures many of modern software development's best practices in a form suitable for a wide range of projects and organizations, including development software iteratively, manage requirements, use component-based architectures, visually model software, continuously verify software quality, control changes to software.

Architecture Based Design (ABD) method is for the high-level software for a product line or long-lived system, which is difficultly to be design because detailed requirements are not known in advance. The ABD method fulfills functional, quality, and business application relies on an understanding of the architecture mechanisms used to achieve this fulfillment<sup>[28]</sup>.

#### 4. SPI KNOWLEDGE SUPPORT STRUCTURE

We make use of the idea from J.N.Warfield in Interactive Management support structure<sup>[8]</sup> and construct SPI knowledge support structure in the following: The figure 1 shows a support structure. What, precisely, is a support structure? As you can see, it consists of boxes that contain text and arrows that connect the boxes. The arrows are all oriented upward. They show the directional flow of support. If there is a directed path from some box to some higher-level box, the one below supports the one at the other end of the path. This is the basic rule for understanding this support structure. The word "support" is key. It means that you will find necessary information in the lower box for the development of what appears above. In this instance, you can see, for example, that two sciences and three processes support SPI. The three sciences are all supported by these two principal components: (1) **Thought about thought** (which J.N.Warfield calls "second-order thought"); (2) **Behavioral pathologies** (which is information about human limitations, individually, in small groups, and in larger

organizations; the information coming from research in psychology, sociology, and organizational studies).

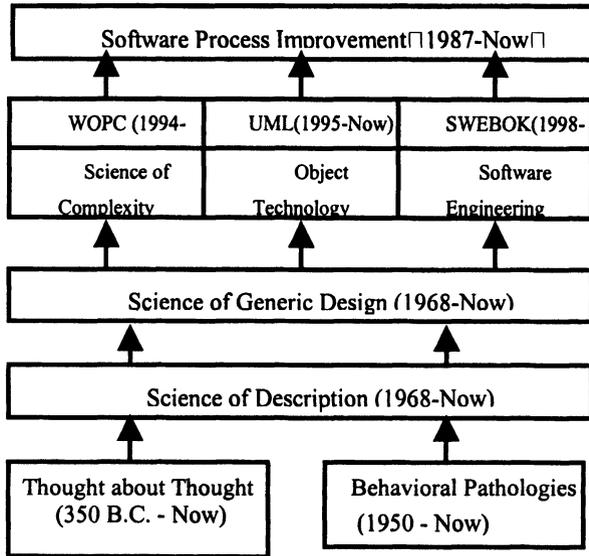


Figure 1: SPI Knowledge Support Structure

You will also notice that each box contains some time information. On the lower left, for example, we see that thought about thought extends from 350 B.C. to now. What this means is that the best work of many scholars has been brought into one place, organized, and drawn upon to support the components lying above. The date 350 B.C. corresponds roughly to the period of Aristotle’s work in logic, in which he developed the first formal statement of deductive logic, along with the concept of categorizing topics so that it would be possible to work with and compare collections of ideas, as well as with the individual ideas themselves. J.N. Warfield has described the pattern that evolved over the many centuries that brought thought about thought into our view today, so that those ideas find their way into SPI, where they help people resolve complexity of SPI.

A Science of Description: People have been describing things for as long as people could talk to one another, or to use written communication or sign language [4].

A Science of Generic Design. This science focused upon the use of information coming from a high-quality description to develop a set of options from which an alternative design could be chosen for implementation. It focuses upon the design of sociotechnical systems [7].

When working in a software project, software engineers apply principles/concepts based on foundation to software developing process and

product through specific and pragmatic methods/technology and building environment/tools<sup>[11]</sup>. Figure 2 illustrates SE cognition support structure.

In our understanding, the UML might be considered a kind means of OT, which has relationship with science of complexity, and SWEBOK might be applied in software organizations through software process (e.g. RUP). WPOC is better than PDCA which does not consider complexity deeply enough and is based mainly on experiences from traditional manufacture industry. The reasons are that PDCA cycle was put by Walter Shewhart in 30's and was further developed by Deming and Joseph Juran in 50's<sup>[15]</sup>, but it's known to all of us that software development is cognitive process intensively, which is intelligence activity intensively. We can apply Science of Complex by UML in SE, and the Science of Generic Design might be considered as the generic foundation of Science of Complex, OT and SE. It's time to consider what is the foundation of SE seriously. Dr. Norman argues that problem solving and human skill are expected in the cognitive process<sup>[26]</sup>. We consider that so are mindbugs solving and SPI.

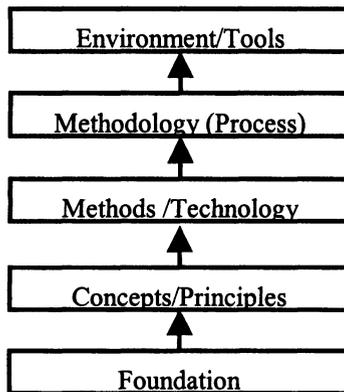


Figure 2: SE Cognition Support Structure

## 5. KNOWLEDGE-BASED ENTERPRISE MODEL

### 5.1 Knowledge-based Enterprise Model for SPI

A model is an abstract representation of reality expressed in term of some formalism. An enterprise model may be used for various purposes such as facilities design, systems architecture, organization, simulation, optimization, performance measurement and benchmarking. Here it will be used to describe and understand the operational of a software organization<sup>[1]</sup>.

An enterprise model for software organizations with general structure is

illustrated in figure 3. The notation of IDEF<sub>0</sub> is used. The outer boundary frames business operation. Input is services and products, money, information and knowledge. Output is services and products, money, information and knowledge. Governmental regulations and other frame conditions may be regarded as control. Owners and external organizations comprise the mechanism. Business operation box includes market, sale, Customer Relationship Management (CRM) and Customer Service (CS). Research and Development (R & D), technological and administrative support functions, infrastructure resources (of any kind, e.g. Human Resources and Intranet etc.), organizational knowledge portal (the knowledge keys to successfully software project operation), as well as SPI. The market is goal, sale is means, and CRM & CR are assurance. R&D is the core element, technological and administrative supporting are forms and infrastructure resources are assurance. The inner box, software production (development), including software process (inception, elaboration, construction and transition) and core workflows and activities supporting.

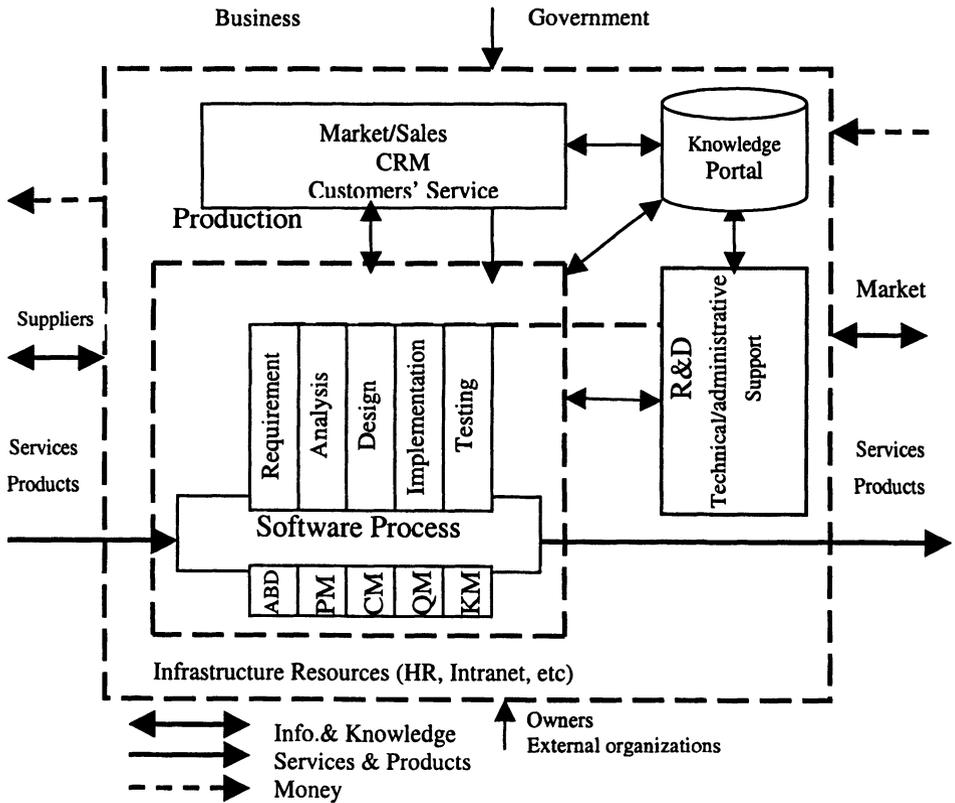


Figure 3: Knowledge-based Enterprise Model for SPI

The core workflows are in the following:

They are requirement, analysis, design, implementation and testing. All of development activities are architecture based, including configuration (change) management (CM), project management (PM, plan, control, cooperation and standard etc.), quality management (analysis and assurance), Knowledge portal is facility for knowledge accumulating, sharing and communication in organization wide, KM should be supported by HR, Intranet etc.

## **5.2 Managing Complexity through Design**

Software process is a collection of complex activities that have strictly time sequences, some are currently asynchronous, some are condition each other and some feedback each other. In fact, the activities in software process have complex network relationship. Software development is highly dynamic cognitive process, and the dynamic change can be found in all phases in software process, such as requirement specification, assignment tasks, debugging, development policy, tools and support environment etc. These changes are often unpredictable and their influences often couldn't be identified because software process is people-oriented and software product is intangible itself, this results are that the software development process is hard to control and its work quality isn't ease to assess, therefore software process has great blurred.

J.N.Warfield argues that one of the primary keys to designing processes that can deal with all of these matters is to choose as the basis for the design of such processes something that is common to all of them <sup>[7]</sup>: namely operations carried out with ideas, such operations not being dependent upon any particular specific kind of idea, nor upon any particular narrow field of study, but simply with ideas themselves.

The variety of fundamental operations that can be carried out with ideas is quite limited. Almost everything that needs to be done can be conceived as (a) generating ideas, (b) clarifying ideas, (c) structuring ideas, (d) interpreting structures of ideas. (e) amending ideas.

The limits number of "idea actions" means that the variety of processes that are need can also be quite limited. One needs only to get processes for generating ideas, processes for clarifying ideas, processes for structuring ideas, and processes for interpreting the structure produced.

Processes reduce personal cognitive burden primarily through the way in which information is sequenced. If members of groups are given rein to choose the topic of their discussion at random, and if several speak at once,

sensible discussion that leads to some organized product is hard to obtain. But if the process is designed so that the subject is broken down into a series of carefully designed questions which are presented under computer control for group discussion and resolution, the dialog becomes focused and the products of the dialog can be aggregated and organized with ease.

In our understanding, software production could systematize in term of enterprise model through conception with knowledge support structure of SPI, and standardize in term of software process management through interactive cognitively process with cognition support structure of SE, structure in term of software engineering project through organizational learning (with UML etc.), and realize its business value through engineering approach (with ABD etc.) finally.

## **6. KM IN SOFTWARE ORGANIZATIONS**

### **6.1 Knowledge-based SPI<sup>[6]</sup>**

Successful SPI requires infrastructure can provide supporting when it is need, including (a) provide resources when they are needed, (b) provide counseling about the direction, scope, and speed of the effort, (c) remove roadblocks so the SPI program proceeds smoothly.

Its purposes are in the following: (a) Maintain visibility for the SPI Program, (b) Facilitate and encourage information-sharing, (c) Capture and retain lessons learned and improvements developed, (d) Provide a support resource. They are all relevant to KM.

It is known to all of us that software organization is based on knowledge. The purpose of SPI is to make software organization quickly response to market and improve software organization's market competence by means of intangible, pragmatic, institutional, systematical and integrated knowledge, this business model is based on effectively and efficiently management on organizational knowledge assets and its relative knowledge processes.

### **6.2 Knowledge Portal in Infosys Corp.**

The Infosys perspective is in the following <sup>[27]</sup>: (1) All organization learning is leveraged in delivering business advantage to the customer; (2) Every Infoscion must have the full backing of the organization's learning behind him/her. Its goal is "Learn Once, Use Anywhere", and conforms to OT 's philosophy.

Evolution of KM at Infosys is in the following: Body of knowledge was established in 1992, Technical bulletin and online learning was available in 1995; Company wide Intranet was built in 1996, Marketing system and process assets was running in 1997; Project leaders' toolkit was constructed in 1998; People knowledge map and organization KM initiative was in action in 1999, in 2000, Integrated Kshop portal was a great milestone of KM implementation in Infosys Corp.

### **6.3 Quaternary Solution of Chinese Enterprise**

Quaternary solution of Chinese Enterprise (QSOCE) is put by Guangzhou keenfox system engineering corp.(<http://www.keenfox.com.cn>), which is a small Chinese software organization, and based on "measuring and managing integrated performance of organization" in term of finance, customer, internal process and learning and growing. The office automation, finance management, CRM and Enterprise's Resources Planning (ERP) are integrated information flow, money flow and material flow through workflow and are seamlessly connected by data mining in order to support decision, management and operation. The QSOCE is also applied to itself except ERP; the knowledge-base SPI is working on the way.

## **7. CONCLUSION**

This paper reviews SPI, science of complexity and KT and software KT, including OT, UML, RUP and ABD etc. We illustrate the SPI knowledge support structure, SE cognition support structure and knowledge-based enterprise model for SPI. We reach the conclusion that software production could systematize in term of an enterprise model through conceptualization with knowledge support structure of SPI, and standardize in term of software process management through interactive cognitively process with cognition support structure of SE, structure in term of software engineering project through organizational learning (with UML etc.), and realize its business value through engineering approach (with ABD etc.). It is also illustrated how do that by examples.

## **8. ACKNOWLEDGMENTS**

J.N.Warfield provided me (first author) with many relative references kindly, and exchanged ideas with me by e-mail, Prof. Wang zhongtuo, Mr.

Shi Chen, Mr. Du Jianhua, Mr. Xia Xunhua, Mr.Li Jianzhang and Mr.Han Huifeng gave me much encouragement and help.

## 9. REFERENCES

- [1] A.Rolstadas, Business Operation by Project, 16<sup>th</sup> World Computer Congress 2000,IT Business Management Forum, 21-25, August, 2000, Beijing, 1-12
- [2] Pierre Bourque, Robert Dupuis, and Alain Abran, et al, Development Consensus on the Software Engineering Body of Knowledge, ICS, 16th World Computer Congress 2000, August 21-25, 2000, Beijing, China
- [3] Watts S.Humphrey, Managing the Software Process, Addison-Wesley, 1989
- [4] J.N.Warfield, Twenty Laws of Complexity Science Application: Application in Organizations, Systems Research and Behavioural, 1999, 16(1), 3-40
- [5] Wang Zhongtuo, Multidisciplinary View of Knowledge Technology, 17-28, Proceedings International Symposium on KSS'2001, Dalian, September 25-27, 2001
- [6] Wan Jiangping, Yang Jianmei, Zhang Zhengang, Knowledge Management in Software Process Improvement, 147-152, Proceedings International Symposium on KSS'2001, Dalian, September 25-27, 2001
- [7] J.N.Warfield, A Science of Generic Design: Managing Complexity Through Systems Design, IOWA State University Press, Second Edition, 1994
- [8] J.N.Warfield, Managing the Unmanageable, <http://www.gmu.edu/library/specialcollections/warfield.htm>
- [9] J.N.Warfield, Mentomology, <http://www.gmu.edu/library/specialcollections/warfield.htm>
- [10] P.Bourque et al, Guide to the Software Engineering Body of Knowledge: <http://www.swebok.org/>, May 2001
- [11] Alan M. Davis. 201 Principles of software Development, McGraw-Hill, Inc. 1995
- [12] Wang Tao, Zhang Hingham, Construction of The Cultural Innovation Knowledge-Based Corporation in The 21st Century, Proceeding of ICM, Xi'an, China, May 5-7, 2001, 972-975
- [13] Senge, P., The fifth discipline, New York: Doubleday, 1990
- [14] John Briggs and F.David Peat, Seven Life of Chaos: Timeless Wisdom from the Science of Change, HarperCollins Publishers, Inc., U.S.A, 1999
- [15] Renato L. Della Volpe, Marcelo S.P.Pessoa, Mauro Spinola, The Role of Software Process Improvement into Total Quality management: An industrial Experience, the Proceedings IEMC-2000, 29-34, August 13-15, Albuquerque, New Mexico, USA
- [16] Cheng Zhongying: C Theory: The Management philosophy of China (in Chinese): ShangHai, XueLin Press, 1999, 315
- [17] Wan Jiangping, Kong Xuedong, Yang Jianmei, Research on CMMI, Application Research of Computer (in Chinese), Vol.18, No.10, 10~13, Chengdu, China
- [18] Wan Jiangping, Yang Jianmei, Wan Jun, People Capability Maturity Model and Its Assessment Framework, Application Research of Computer (in Chinese), Vol.18, No.1, 16~18, Chengdu, China
- [19] IEEE Standard Glossary of Software Engineering Terminology, IEEE, Piscataway, NJ STD 610.12-1990, 1990
- [20] Wan Jiangping, Yang Jianmei, Some Consideration on Interactive Management (in Chinese), Accepted by ISSS2002 Conference in China, August 2-6, 2002, Shanghai, China

- [21] Wan Jiangping, Yang Jianmei, On the Work Program of Complexity (in Chinese), to be appeared
- [22] Peng geng, Li mingqiang, Research on Organizational Learning and Learning Organization (in Chinese), Beijing: China Soft Science, No.11, 117-120
- [23] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addison Wesley Longman, Inc. 1999
- [24] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Modeling Language Manual, Addison Wesley Longman, Inc. 1999
- [25] Cai Xiyao, New development and features of software technology, electronic technology report (in Chinese), Vol.10, 1998, 9-12
- [26] Ronald J. Norman, Object System Analysis and Design □ Prentice Hall, Inc, 1996
- [27] Kochikar, Managing Knowledge: The Infosys Experience, <http://www.qai.com>, 2001
- [28] Bachmann, F; Bass, L.; Chastek, G, et al, The Architecture Based Designed Method, <http://www.sei.cmu.edu/publications/documents/00.reports/00tr001.html>, 2000