STP-CMM: A Grading Practice Capability Maturity Model for Software Talent Cultivation

Tao Wu
School of Software Engineering of Huazhong University of Science and Technology, Wuhan, China
Email: dittaowu@gmail.com

Feng Ning
School of Software Engineering of Huazhong University of Science and Technology, Wuhan, China
Email: fengninghust@gmail.com

Lai-Yuan Xiao
School of Software Engineering of Huazhong University of Science and Technology, Wuhan, China

Chuan-Bo Chen
School of Software Engineering of Huazhong University of Science and Technology, Wuhan, China

***Abstract*—Nowadays in China, there is a huge gap between the capability of newly graduate students and that required by software industry, which caused a severe supply-demand contradiction between universities and software industry. To solve this problem, this paper proposed a novel practice capability maturity model for software talent cultivation (STP-CMM). It consists of four levels: Awareness Level, Curriculum Level, Project Level and Enterprise Level, which represent a path of continuous improvement for those universities who want to gradually improve the practice capability of their students. STP-CMM has been running for years in the Software Engineering Discipline of Software College of Huazhong University of Science and Technology (HUST), the results have shown that STP-CMM is of great instructive significance to the practice capability training of software talent.**

***Index Terms*—software talent, practice system, practice capability, key process area, career ability**

I. INTRODUCTION

In recent decades, software industry has become the focus of the latest international competition and the strategic pilot industry for which various countries are struggling. Having a group of high-level, compound, engineering-type and competitive employees is the most important guarantee of the sustainable development of software industry. However, the traditional training modes used in the universities of China are focused on the theoretical study, lacking appropriate systematic practice training. There exists a huge gap between the capability of newly graduate students and that required by software industry, which caused a severe supply-demand contradiction between universities and software industry [1]. For example, on one hand more and more graduates can't find jobs; on the other hand it is hard for some software companies to find qualified employees even promising a fair salary.

To determine how to cultivate qualified students, we did some investigation and research on the training modes used in some countries, whose software industries are comparatively developed, such as India and Ireland [2]. In those countries, the effect of practice is strongly emphasized. Companies and enterprises are encouraged to communicate with universities, so the quality of the students could meet the requirements of industry as far as possible. Based on the research, we concluded that there are four fundamental qualities which are essential to be a qualified software employee: basic theories, specialized knowledge, professional skills and career ability, as shown in Figure 1. Here, career ability contains communication skills, honesty, teamwork, interpersonal skills, motivation and work ethic. All these qualities are supposed to be formed coordinately in curriculum system and practice system.

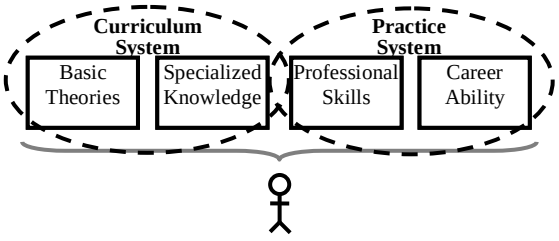


Figure 1. Required qualities of software employee

As to curriculum system, some international standard knowledge systems have been introduced to China, such as CSDA Learning System [3] and Guide to the Software Body of Knowledge (SWEBOK) [4,5,6] from IEEE and Software Engineering Education Knowledge (SEEK) from ACM [5,6]. It is obvious that all these international standard knowledge systems have systematic classification about basic theories and specialized

doi:10.4304/jsw.6.4.636-642

knowledge for different disciplines. However, these curriculum systems are hard to systematically define the maturity levels of professional skills and career ability about these disciplines, or to improve the practice capability gradually of the students, such as software engineering.

Following the characteristics of international software training mode, combined with the experience and thoughts gained in the education of our school, a novel practice capability maturity model was put forward for software talent cultivation (STP-CMM), which can be used to guide the training of professional skills and career ability. The model is organized into four maturity levels: Awareness Level, Curriculum Level, Project Level and Enterprise Level.

Since practice system and curriculum system are the essential parts of capability cultivation system for the students of software engineering, we should follow the ideas when design the training program for software engineering discipline. Figure 2 shows the cultivation system used in Software Engineering discipline of School of Software Engineering of HUST. The training program consists of two parts: curriculum system and practice system, which are interrelated according to different semesters. The knowledge system of Curriculum System takes the fifteen content domains in CSDA (Certified Software Development Associate) as a reference. The hierarchy of the maturity levels of practice system used in STP-CMM is quite similar to CMM [7].

II. IDEAS OF IMPROVING PRACTICE CAPABILITY

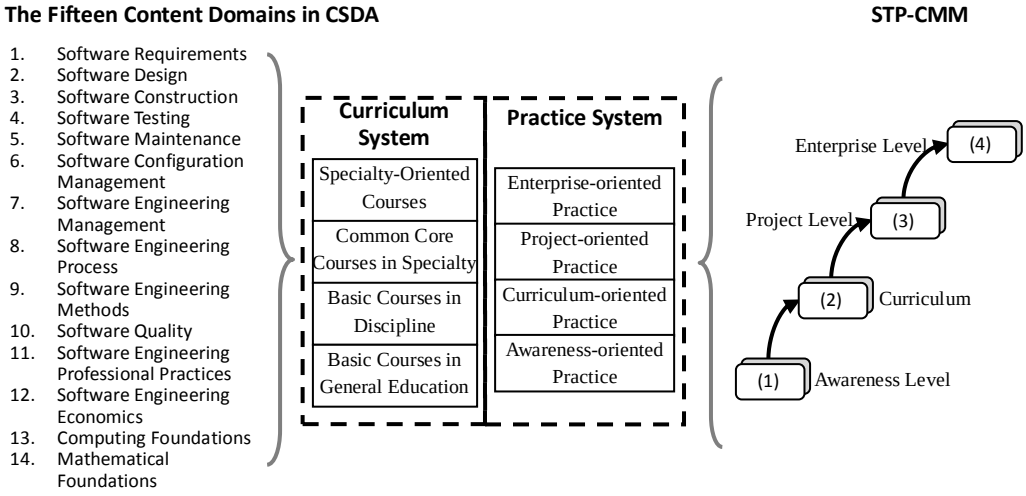


Figure 2. The cultivation system used in SSE, HUST

To improve students' practice ability effectively, we should follow the points below when arranging practice activities:

(1) Different contents and goals will be set according to different level of STP-CMM.

(2) Standard enterprise-like workflow is strictly carried out throughout the whole practice process.

(3) Advanced developing tools and technology accompanied by modern software engineering and project management are adopted.

(4) The basic ideas of different practice activities by stages are shown in Table I.

TABLE I
ARRANGEMENT IDEAS OF PRACTICE ACTIVITIES

Maturity Level	Key Points of Curriculum System	Arrangement of Practice Activities
Awareness Level	Build up system-level cognition of computer system and information technology; Learning some basic knowledge of software developing.	To experiment some basic concept of computer system. In programming aspect, starting to use the latest tool and platform, by the mean time, learning the essential skills.
	The related courses include "*C Programming Language", "College Computer Basics"* etc.	
Curriculum Level	Study the fundamental knowledge system about software engineering, which includes "*Data Structure and Algorithm Analysis*", "*Principle of Operating Systems*", "*Personal Software Process*", etc.	Enhancing the understanding about the basic courses. Getting to know software engineering architecture and experiencing software development life circle through projects and practices based on the course projects.
Project Level	Learning to apply theory and technology of software development and software engineering to a certain kind of application. .	Getting acquainted with the details of team software process; Cultivating the ability of cooperation, communication and innovation.
Enterprise Level	Tracing advanced knowledge and the latest trends in software industry.	Experiencing the real developing process of industry; Accumulating skills or experience of project management and using front technology.

III. SOFTWARE TALENT PRACTICE CAPABILITY MATURITY MODEL – STP-CMM

STP-CMM describes an evolutionary path of continuous improvement from an inexperienced freshman to well-trained graduate. It is a comprehensive model used to measure a student's practice capability, and also a reference guideline to the training program of IT discipline, such as software engineering.

When students participate in and get through the arranged practice activities prepared in STP-CMM, their practice capability will grow gradually following the maturity levels. A maturity level is a well-defined evolutionary platform towards achieving mature practice capability. When goals of all the key process areas for a level are achieved, the maturity level will be raised to the next level.

A. The Four Levels of STP-CMM

Figure 3 shows the key process areas in different maturity levels, each maturity level contains several different key process areas (KPA). The KPAs identify the key points that should be conformed to achieve a maturity level.

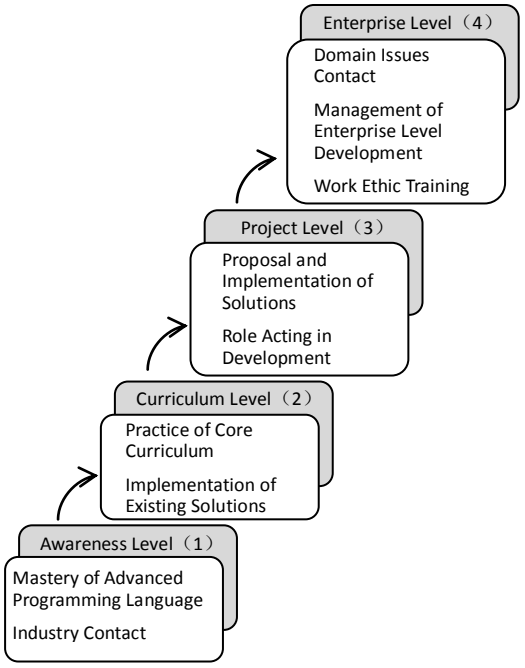


Figure 3. The key process areas by maturity level

1) Level 1 – Awareness Level

At Awareness Level, we assume that the newly enrolled students have little impression about software developing and computer system. In this stage, students will attend classes to learn the foundation knowledge of their discipline. The typical courses includes "*Computer Science：An Overview*" and "*Introduction to Information System and Technology*" etc. At the same time, students will master an advanced programming1 language, such as C, C++ and Java.

Basically, the students of practice capability Level 1 don't have deeply cognition about software system and industry. The practice contents of this level focus on acquainting with computer system and preliminary programming.

The key process areas for level 1 are given below:

(1) Mastery of Advanced Programming Language

The purpose of this KPA is to let students gain some basic knowledge on programming and have the ability to use an advanced programming language to solve some primary programming problems.

(2) Industry Contact

The purpose of this KPA is to let students get deeper understanding of software system or industry and get a basic impression about the application of software engineering.

2) Level 2 – Curriculum Level

At Curriculum Level, students have already known the fundamentals of software engineering, and they will learn basic courses of their discipline. The typical courses include "*Introduction to Software Engineering*", "*Data Structure and Algorithm Analysis*", "*Personal Software Process*" etc. The specialized knowledge and programming ability of students will be improved a lot at this level.

Practice capability of the students in Level 2 will have a prominent leap. The practice contents of this level contain both programming exercises in courses and the implementation of existing solutions of problems in certain application domain, such as telecommunications, finance or gaming etc. And students will start to know some simple project management knowledge.

The key process areas for level 2 are given below:

(1) Practice of Core Curriculum

The purpose of this KPA is to guide students to apply learned knowledge flexibly to finish the programming exercises in courses.

(2) Implementation of Existing Solutions

The purpose of this KPA is to guide students to apply theoretical knowledge and programming ability to implement some existing solutions in certain domain.

3) Level 3 – Project Level

At Project Level, students would have laid a solid foundation both theoretically and practically. At this level, they will take core courses in their specialty, such as "*Requirements Engineering*", "*Software Testing*", "*Software Architecture in Practice*" etc.

The practice contents of this level are not limited to a certain knowledge area any longer. Students begin to be involved in the development of small and medium-size projects. They will experience team software process, including basic role playing and team cooperation.

The key process areas for level 3 are given below:

(1) Proposal and Implementation of Solutions

The purpose of this KPA is to train the students to investigate and analysis some problems in certain domain and propose their own solutions, and then implement them.

(2) Role Acting in Development

The purpose of this KPA is to guide students acting different roles in a development team and cultivate their team work spirit and communication ability.

4) Level 4 – Enterprise Level

At Enterprise Level, students are ready to work as software engineers. At this level, they will majorly learn some specialty-oriented courses, like lectures of advanced knowledge and the latest trends of industry etc.

The practice contents of this level are mainly about how to improve the career ability of the students. The students will take part in different roles in real software developing environment and try to propose and implement industry-depth solutions.

The key process areas for level 4 are given below:

(1) Domain Issues Contact

The purpose of this KPA is to guide students to do deep investigation and researches about industry problems, then try to propose solutions and finally implement them.

(2) Management of Enterprise Level Development

The purpose of this KPA is to guide students experience real management of software development in software companies, in order to improve their career ability.

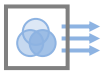
(3) Work Ethic Training

The purpose of this KPA is to train the students with strong work ethic. The students will commit to achieve their objectives of the organization and provide benefit to other colleagues when they start working.

B. Model Analysis

Based on the brief introduction of the four levels of STP-CMM, we will carry out a further analysis of STP-CMM whose goals and features are all shown in Table II.

TABLE II
ANALYSIS OF STP-CMM

	Awareness Level	**Curriculum Level**	**Project Level**	**Enterprise Level**
Practice Goals	Freshman year Get access to software industry, and stimulate interest in software engineering	Sophomore year Teach them the ideas of software development life circle and some fundamental knowledge development life circle	Junior year Make students understood team developing process, such as roles playing and cooperation.	Senior year Training the ability of student to play roles taking in the development of enterprise-level project and to propose and implement industry-oriented solutions
Practice Features	Preliminary programming practices Initial contact with different application in industry Involved in small-size project with little management Time-consuming: 2-3 weeks	Course centered application practice Concerned about small size problems in certain domain Involved in simple project management Time-consuming: 3-5 weeks	Practice roles acting in industry Concerned about problem with special objectives s in industry Involved in complex project management Roles acting in industry Time-consuming: 6 weeks	Practice work placement or internship in software industry Concerned about problems with multiple objectives in industry Involved in more complex project management Roles taking in industry Time-consuming: 2 months
Key Process Area	Mastery of Advanced Programming Language Industry Contact	Practice of Core Curriculum Implementation of Existing solutions	Proposal and implementation of solutions Role Acting in Development	Domain Issues Contact Management of Enterprise Level Development Work Ethic Training
Curriculum Goals	Build up primary cognition of computer science and learning some basic knowledge of software developing	Studying fundamental knowledge of software engineering and having access to theory of software engineering	Applying learned knowledge and relevant technology to software development	Studying advanced knowledge and tracing latest trends of industry
Representation Forms				

"Representation Forms" in row 6 clearly illustrates that how practice system affects the application of basic theories and specialized knowledge. At Awareness Level, The correlation of different knowledge areas learned by students is quite low, showing a scatter pattern. At Curriculum Level, having studied basic courses in their discipline, students can apply learned knowledge to solve some complex problems, the learned knowledge areas of students show a joint pattern. At Project Level, students can develop small and medium-sized projects using the knowledge of a specific domain flexibly, now the learned knowledge areas of them show a merge pattern. At Enterprise Level, students start to use their knowledge freely and flexibly to solve industry-oriented problems, and they will take part in enterprise-level project using modern developing technology and tools accompanied by a specific corporation culture.

IV. IMPLEMENTATION OF STP-CMM

For years, the practice training plan of software engineering discipline in SSE of HUST had been following the grading path described in STP-CMM. Due to the close relationship with famous software companies both domestic and abroad, diversity training methods are applied during the implantation process of STP-CMM, including making use of industry oriented projects or projects cases, using the latest tools and technology in development, inviting guest lecturers from industry to give speeches occasionally etc.

A. Awareness Level

Practice activities at Awareness Level are mainly preliminary programming, detailed in Table III. The time period at this level could be organized in different styles which are centralized style, distributed style and decentralized style. The duration of the three styles is 2 weeks, 3 weeks and 3 weeks respectively.

Based on the characteristics of this level, we divide the standard implementation process of awareness level into the following stages: opening, project start, implementation, testing, project acceptance and ending. Because the scale of the practice activities at this level is very small, the project plan, requirement description and system design are given to students in project start stage.

TABLE III
PRACTICE CONTENTS AT AWARENESS LEVEL

Key Process Area	Involved Courses	Practice Activities		Duration
Mastery of Advanced Programming Language *Industry Contact*	*"College Computer Basics"* *"Introduction to Information Technologies"* *"C Programming Language"*	*A tool for browsing files and folders*		*3 days*
		Introduction to CoDeSys Developing Platform of 3S company		*1 day*
		Console Application 24-point rummy		*2 days*
		Library Affair Management System	*select one*	*7 days*
		Salary Management System		*7 days*

B. Curriculum Level

Practice training at Curriculum Level will take about 1 month, and the implementing period could be organized in centralized style, distributed style and decentralized style, too. Choosing different style, the duration will change from 3 weeks to 5 weeks, detailed in Table IV.

Based on the characteristics of this level, combined with Personal Software Process 3 [8], we divide the standard process into eight stages: opening, project start, project plan, system design, implementation, testing, project acceptance and ending. Requirement description will be given to students in project start stage. And because of the scale and complexity of the practice activities are not large, students only have to carry out high-level design in project design stage, without doing outline design.

TABLE IV
PRACTICE CONTENTS AT CURRICULUM LEVEL

Key Process Area	Involved Courses	Practice Activities	Duration
Practice of Core Curriculum *Implement-ation of Existing solutions*	*"Software Engineering"* *"Software Quality and Testing"* *"Data Structure and Algorithm Analysis"* *"Personal Software Process"* *"Object Oriented Analysis and Design"* *"Computer Networks"* *"Principle of Operating Systems"* *"Compiler Techniques"*	*Data Exchange between Microsoft Excel and Microsoft SQL Server*	*5 days*
		Ajax Web Interface Data Exchange	*5 days*
		Web Control of Google Map	*7 days*
		Minesweeper Game Application	*10 days*

C. Project Level

Practice training at Project Level will take about 6 weeks, and it's divided into nine stages: opening, project start, project plan, requirement analysis, system design, implementation, testing, project acceptance and ending. At this level, we recommend using centralized style and distributed style to organize the training process, detailed in Table V.

TABLE V
PRACTICE CONTENTS AT PROJECT LEVEL

Key Process Area	Involved Courses	Practice Activities		Duration
Proposal and implem-entation of solutions *Role Acting in Develop-ment*	*"Principle of Database Systems"* *"Advanced Internet Programming"* *"Software Architecture"* *"Unified Modeling Language"* *"Analysis and Design of Information System"* *"CMM Lectures"* *"Requirement Engineering"*	*Microsoft Content Management Service Manager (team work)*	*select one*	*6 weeks*
		Point-to-Pint Application for Black Berry Mobiles (team work)		*6 weeks*
		Chat Tool with Customizable Interface (team work)		*6 weeks*

D. Enterprise Level

Practice training at Enterprise Level will take about 2 months, and it's divided into nine stages: opening, project start, project plan, requirement analysis, system design, implementation, testing, project acceptance and ending, detailed in Table VI.

At this level, the training process should be organized in centralized style. Projects at this level will be held in practice center, a special place promoted by School-enterprise cooperation. The whole process of different practice teams are guided by tutors from

different software companies with special enterprise culture.

TABLE VI
PRACTICE CONTENTS AT ENTERPRISE LEVEL

Key Process Area	Involved Courses	Practice Activities		Duration
Domain Issues Contact *Management of Enterprise Level Development* *Work Ethic Training*	*"Advanced Software Developing Technique"* *"ERP and Supply Chain"* *"Information Catch and Communication Skills"* *"Project Management Case Study"* *"Entrepreneurship Lectures"*	*Human Resources Management System (Yokogawa, Singapore) (team work)*	*select one*	*2 months*
		Design and Implementation of Wireless Network Driver (Microsoft Research Asia) (team work)		*2 months*

V. ACCOMPLISHMENT OF RUNNING STP-CMM

For years in the software engineering discipline of software college of HUST, the STP-CMM had been implemented to improve the practice capability of the students. The STP-CMM shows its advantages in two aspects. On one hand, the students had won splendid prizes from different famous competitions, as shown in table VII.

TABLE VII
AWARD INFORMATION

Year	Competition List	Awards	Award-Giving Department
2008	Imagine Cup	World No.3 in IT challenge World No.6 in Software Design	Microsoft Corp.
2007	Imagine Cup	Champion in China World No.3 office design	Microsoft Corp.
2006	Citi Financial & IT Competition	The Third Price	Citigroup Inc.
2005	Citi Financial & IT Competition	The First Price	Citigroup Inc
2004	Innovation Cup	The Third Price	Microsoft Research Asia

On the other hand, the graduates have received widely recognition by the companies or research institute of IT industry. Most of them have been recruited by famous domestic enterprises or MNCs, such as IBM, Microsoft, Citigroup, Adobe, Baidu, HUAWEI, SunGard and Infosis etc.

VI. CONCLUSION

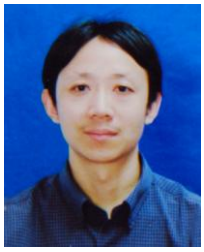
In this paper, a novel software talent practice capability maturity model (STP-CMM) is presented to improve students' practice capability gradually. STP-CMM is a dispensable part of the training program of software engineering, and it can be used as a standard to manage, implement or guide the practice process of the students. Besides, it can be easily extended to the practice process of different disciplines. The results of running STP-CMM for years in Software Engineering Discipline of SSE in HUST have shown that it is of great importance to the training of students' practice capability. The application of STP-CMM makes the software talents more qualified for the software industry.

ACKNOWLEDGEMENT

Many people have made invaluable contributions to our research, both directly and indirectly. We would like to thank Prof. Thomas B. Hilburn and Henry Ho from IEEE for introducing CSDA and its Learning System to us. Besides, we want to thank Prof. Bing Wu, Prof. Ronan Bradley and Lecturer Michael Collins from D.I.T Ireland for the ideas of industry Oriented Education for Computing.

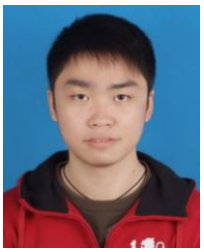
REFERENCE

[1] Wu Tao, Cao Ming, Chen Chuanbo. Research on Case-Driven Software Talents Training Plan. Proceedings of 2007 Annual Conference of Chinese Universities Education in Software Engineering Discipline. 2007
[2] Zhai Yuqin. Thoughts on the Investigation of India and Ireland. Computer Education, 2004, (11): 45-46
[3] IEEE Computer Society. CSDA learning system [M]. Los Alamitos, 2008
[4] Personal Practices Committee of IEEE Computer Society. Guide to the Software Engineering Body of Knowledge 2004 Version. 2004
[5] IEEE Computer Society/ACM Joint Task Force. Software Engineering 2004 – Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. 2004
[6] Bourque, P. Robert, F. Lavoie, J. M. Lee. Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Knowledge (SEEK) - a preliminary mapping. Software Technology and Engineering Practice, 2002. STEP 2002. Proceedings. 10th International Workshop. 2002, 8-23
[7] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber. Capability Maturity Model[SM] for Software, Version 1.1. Pennsylvania: Software Engineering Institute of Carnegie Mellon University, 1993
[8] Watts S. Humphrey. Using a defined and measured Personal Software Process. IEEE Software, 1996, (13): 77-88



Tao Wu received the Ph.D. in Mechanical Science and Engineering of Huazhong University of Science and Technology, China in 2002. Since then, he has been working as an associate professor at School of Software Engineering, Huazhong University of Science and Technology, China. And he was selected as the head of Software

Science and Engineering Department in 2007. His research interests include Software Engineering and Virtual Reality.



Feng Ning received the B. S. in software engineering from Huazhong University of Science and Technology, China in 2008. Now he is a M. S. student in School of Software Engineering, Huazhong University of Science and Technology, China. His research interests are Software Process Management and Trusted Software Engineering.

Lai-Yuan Xiao is a professor and the vice-president in School of Software Engineering of Huazhong University of Science and Technology, China. His research interests include Domain Engineering, Modern Design Theory and Methods and Software Engineering.

Chuan-Bo Chen is a professor and the president in School of Software Engineering of Huazhong University of Science and Technology, China. His research interests are Pattern Recognition, Machine Intelligence and Software Engineering.