IPD-CMMI Model of Embedded Software Engineering under IEC-CDIO Framework

Tiejun Pan
Zhejiang Wanli University, Ningbo, 315100, China
Email:pantiejunmail@126.com

Leina Zheng
Zhejiang Wanli University, Ningbo, 315100, China
Corresponding Author Email: mdsryx@126.com

Abstract—**With the increase of the embedded system in the post-PC era of computing, the Innovation & Enterprise Capability training becomes a task of high priority. The drawback of the current software engineering to Higher Education is systematically analyzed and the IPD-CMMI (Integrating Product Development based on CMMI) is proposed by integrating the hardware and software engineering and tailoring unnecessary software process for teaching. Then, according to the Long-term Education Reform and Development Plan, IEC-CDIO (Innovation & Enterprise Capability-CDIO) framework is proposed stressing engineering education in the context of Conceiving—Designing—Implementing—Operating (CDIO) real-world systems and products. In the end, a support system is created based on ODC (Orthogonal Defect Classification) reference model which includes activity, trigger, severity attributes and defect origin, content, type. It can carry on Process and Quality analysis by ODC database and forecast the Plan Progress which helps the teacher to schedule the balance of study and work term.**

Index Terms—**Capability Maturity Model Integration, CDIO, software engineering, embedded system, ODC, innovation**

I. Introduction

For decades, solving the software crisis was paramount to researchers and companies producing software tools. Seemingly, they looked for the new technology and practice as a silver bullet to solve the software crisis. Tools, discipline, formal methods, process, and professionalism were regard as silver bullets: Tools: Especially emphasized object-oriented programming, CASE tools, documentation, and standards.

Discipline: Excellent discipline of programmers.

Formal methods: Some believed that if formal engineering methodologies would be applied to software development.

Process: Many advocated the use of defined processes and methodologies like the Capability Maturity Model (CMM).

Professionalism: This led to work on a code of ethics, licenses, and professionalism.

Fred Brooks (1986) published the No Silver Bullet article, arguing that no individual technology or practice would ever make a 10-fold improvement in productivity within 10 years. Eventually, almost everyone accepted that no silver bullet would ever be found. Yet, claims about silver bullets pop up now and again, even today. [1]

In spite of whether silver bullets exist or not, the popularity of CMMI (Capability Maturity Model Integration), XP (Extreme Programming) and other software engineering technology would be very helpful to improve our software process.

The post-PC era is seeing the emergence of embedded systems consist of software and relevant hardware, such as small and portable computer, Smartphone, Information appliances and computers embedded in everyday applications. Everyone always make an individual interact with several computers every day, so that it is necessary to study the coordination of the software engineering for the embedded system.

An embedded system is a computer system designed to do one or a few dedicated and/or specific functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today. [2]

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, smart phone share some elements with embedded systems such as the operating systems and microprocessors that power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems that do not expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call "embedded".

In the post-PC era of computing, the use of computer systems based on real-time and embedded technologies has already touched every facet of our life and is still growing at an unprecedented pace. While embedded processing and Internet-enabled device have now captured everyone’s imagination, they are just a small fraction of applications that have been made possible by real-time systems. [3]

The search for a single software solution to success never worked in embedded system. All known technologies and practices have only made incremental improvements to productivity and quality. Yet, there are no standard solutions for a verity of embedded system, either. Others interpret no ultimate model as proof that embedded software engineering has finally matured and recognized that projects succeed due to hard work, especially in the repaid developing embedded system. [4]

However, it could also be said that there are, in fact, a range of good model and technologies today, including lightweight methodologies, Personal Software Process (PSP), Team Software Process (TSP), CMMI and Rational Unified Process (RUP) that solve all kinds of problems in embedded system development in different level. Nevertheless, the field of embedded software engineering appears too complex and diverse to improve most issues, and each issue accounts for only a small portion of all software problems. In this paper, we pay more attention to the software and hardware coordination model in embedded software engineering, the engineering capability Training and relevant supporting system.

II. IPD-CMMI Model

With the expanding demand for embedded software in many smaller organizations, the need for inexpensive software solutions led to the growth of simpler, faster methodologies that developed running software, from requirements to deployment, quicker & easier. The use of rapid-prototyping evolved to entire lightweight methodologies, such as Extreme Programming (XP), which attempted to simplify many areas of software engineering, including requirements gathering and reliability testing for the growing, vast number of small software systems. Very large software systems still used CMMI methodologies, with many volumes in the documentation set; To embedded system development, however, the tailored CMMI is necessary that should have a simpler, faster alternative approach to managing the development and maintenance of software and hardware integration, including calculations and algorithms, information storage/retrieval and display.

Embedded Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software based on relevant hardware, and the study of these approaches; that is, the application of engineering to embedded software. The best theory result may be the CMMI that is a process maturity model used by software organizations to ascertain the current maturity level of their development processes and to provide measures for achieving higher maturity levels [6]. CMMI certification is popular in IT organization, so that it is reasonable to design the embedded software engineering model integrating software and hardware product developing according to CMMI named IPD-CMMI.

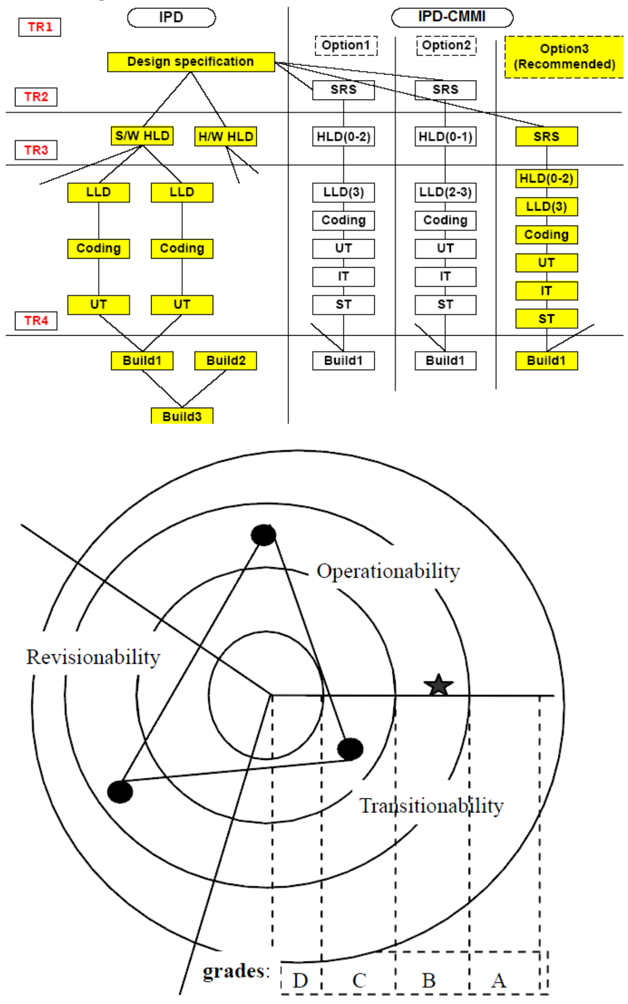

Figure 1. IPD-CMMI process

IPD-CMMI is mainly designed for the profession capability training of college students developing embedded system shown in Fig.1. It includes the phases Planning, Design, Code, Compile, Test, Build and Postmortem. Software and hardware design are separately divided into High Level Design (HLD) for conception and Low Level Design (LLD) for realization. Test is divided into Unit Test (UT), Integrated Test (IT) and System Test (ST). In general, the embedded software requirements analysis always is backward in time with the hardware’s because the stable hardware platform is a prerequisite for software development. So that, the software requirements specification (SRS) relevant hardware may be planed during the hardware’s High Level Design (HLD) phases, so does the other procedure of embedded software development until the build stage. The baselines are always setup at four milestones including TR1 at project startup point, TR2 at hardware design specification checkpoint, TR3 at embedded system HLD checkpoint, TR4 at embedded system test (ST) checkpoint. IPD-CMMI is used throughout the development lifecycle. The radar chart (Kiviat) is used as quantitative measurement to evaluate software quality and describe the evolution of the IPD-CMMI graphically. (solid dots represent evaluation grades of quality characteristics, the solid five-pointed star represent comprehensive evaluation grades of embedded software quality). It offers relevant assessment staff an intuitive and effective method to examine comprehensive evaluation result of software quality and the quality level of each quality characteristic. This Essential of IPD-CMMI focuses on organizing a project, integrating software and hardware development.

IPD-CMMI help college students' deal with quality attributes by providing integration methodology to cooperate or coordinate software and hardware process in the product development. IPD-CMMI describes how all development actions should behave in particular circumstances (Higher Education). For example, embedded system software development should add scheduling, debugging, logging, or locking control into all objects of particular hardware.

IPD-CMMI is a development model that was created after study of data collected from universities, enterprises and organizations that contracted with the Chinese Ministry of Education, who founded the research. It can be viewed as a set of structured levels that describe how well the behaviors, practices and processes of an organization can reliably and sustainably produce required embedded system product. IPD-CMMI may provide: a place to start for the freshman in IT fields, the benefit of a university's prior experiences, a framework for prioritizing actions and a way to define what improvement means for university or IT organization. It can be used as a benchmark for comparison and as an aid to assess different organizations where there is something in common that can be used as a basis for comparison. In the case of the IPD-CMMI, the basis for comparison would be the university's embedded system teaching processes.

IPD-CMMI involves the following five aspects according to CMMI for development whose architecture shown in Fig. 2:

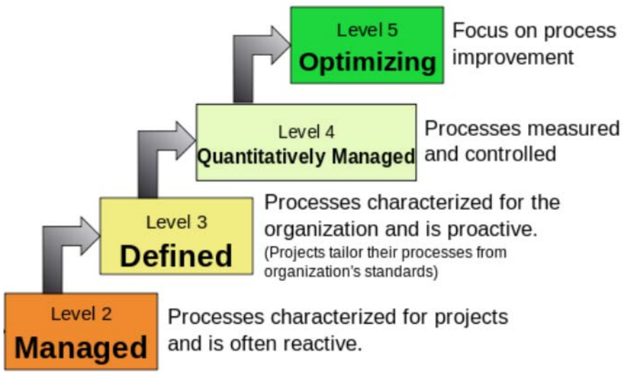
Level 5
Optimizing
Focus on process improvement
Level 4
Quantitatively Managed
Processes measured and controlled
Level 3
Defined
Processes characterized for the organization and is proactive.
(Projects tailor their processes from organization's standards)
Level 2
Managed
Processes characterized for projects and is often reactive.

Figure 2. IPD-CMMI Maturity levels

Maturity Levels: a 5-level process maturity continuum - where the uppermost (5th) level is a notional ideal state where processes would be systematically managed by a combination of teaching process optimization and continuous process improvement. However, IPD-CMMI maturity level ratings are awarded for levels 2 through 5 because of the relatively perfect teaching system.

Key Process Areas: a Key Process Area (KPA) identifies a cluster of related teaching or training activities that, when performed together, achieve a set of goals considered important. Some unnecessary KPAs for teaching are removed from IPD-CMMI.

Goals: the goals of a key process area summarize the states that must exist for that key process area to have been implemented in an effective and lasting way. The extent to which the goals have been accomplished is an indicator of how much capability the organization has established at that maturity level. The goals signify the scope, boundaries, and intent of each key process area.

"Education and training of outstanding engineers plan" published by Ministry of Education in China is the goal of IPD-CMMI, which implement the "Long-term Education Reform and Development Plan (2010-2020)" and "Long-term talent development program (2010-2020 years), " It is not only a major reform projects, but also to promote engineering education in China. The program aims to train a large number of engineers having innovative ability to adapt to economic and social development needs of various types of high-quality engineering and technical personnel for the country to take a new path of industrialization, serving the strategy of building an innovative country.

Common Features: common features include practices that implement and institutionalize a key process area. There are five types of common features: commitment to Perform, Ability to Perform, Activities Performed, Measurement and Analysis, and Verifying Implementation. The 12 standards of CDIO, the famous engineering education methodology, are integrated into the instruction to training the embedded system engineer.

Key Practices: The key practices describe the elements of infrastructure and practice that contribute most effectively to the implementation and institutionalization of the KPAs. The CDIO systematic training projects are designed to help the students practice.

There are four levels defined along the continuum of the IPD-CMMI except the initial Level,

Managed - the process is at least documented sufficiently such that repeating the same steps may be attempted which is important to the ordinary teaching or training.

Defined - the process is defined/confirmed as a standard teaching or training process, and decomposed to levels 1 and 2 (the latter being Teaching Instructions).

Managed - the process is quantitatively managed in accordance with agreed-upon metrics which can match the CDIO capability/process metrics for the excellent engineer training.

Optimizing - process management includes deliberate process optimization/improvement. It is a characteristic of processes at this level that the focus is on continually improving process performance through both incremental and innovative technological changes/improvements, which is our ultimate goal.

Within each of these maturity levels are Key Process Areas (KPAs) which characterize that level, and for each KPA there are five definitions identified: Goals, Commitment, Ability, Measurement, and Verification.

There are four maturity levels in IPD-CMMI. However, Higher Education system has setup the software training process according to the national education standard in China, so that IPD-CMMI maturity level ratings are awarded for levels 2 through 5. The process areas below and their maturity levels are listed for Higher Education model in which unnecessary process areas is cut off:

Maturity Level 2 – Managed

CM - Configuration Management
MA - Measurement and Analysis
PMC - Project Monitoring and Control
PP - Project Planning
PPQA - Process and Product Quality Assurance
REQM - Requirements Management

Maturity Level 3 - Defined

DAR - Decision Analysis and Resolution
IPM - Integrated Project Management
OT - Organizational Training
PI - Product Integration
RD - Requirements Development
RSKM - Risk Management
TS - Technical Solution

Maturity Level 4 - Quantitatively Managed

QPM - Quantitative Project Management

Maturity Level 5 - Optimizing

CAR - Causal Analysis and Resolution

SAM (Supplier Agreement Management) in the Maturity Level 2 is not necessary because the logistical department of University has setup the comprehensive supply chain system for Higher Education. What IPD-CMMI can do is only providing the interoperating interface to access the logistical system. Moreover, the university has its comprehensive HR system and management of organizational process, the evaluation if organizational process and its performance are beyond the range of capability training for embedded software engineering, so that it is not necessary of OPD (Organizational Process Definition), OPF (Organizational Process Focus) in the Maturity Level 3, OPP (Organizational Performance Management) in the Maturity Level 4 and OPM in the Maturity Level 5. Owing to the training lab is relatively closed environment, and the train projects are relatively simple to the real world, so that VAL (Validation) and VER (Verification) in the Maturity Level 3 are simply. We pay more attention to the teaching framework to train the college student's innovative and enterprise capability.

III. IEC-CDIO Framework

CDIO (Conceive—Design—Implement—Operate) is an engineering education initiative that was formally founded by Massachusetts Institute of Technology [1] in the late 1990s. In 2000 it became an international collaboration, with universities around the world adopting the same framework. CDIO collaborators recognize that an engineering education is acquired over a long period and in a variety of institutions, and that educator in all parts of this spectrum can learn from practice elsewhere. They setup CDIO network therefore welcomes members in a diverse range of institutions ranging from research-led internationally acclaimed universities to local colleges dedicated to providing students with their initial grounding in engineering.

Throughout the world, CDIO collaborators have adopted CDIO as the framework of their curricular planning and outcome-based assessment which is prevalent in the United States and Western Europe, and being accepted and respected by the higher education sector in China. During the "Post-crisis" period, innovation and enterprise capability cultivation is the only way of upgrading industry from "made in China" to " invented in China", so that IEC-CDIO (Innovation & Enterprise Capability-CDIO) is proposed by T. J. Pan (2010) which is an innovative educational framework for producing the next generation of engineers. The framework provides students with an education stressing engineering fundamentals set in the context of Conceiving—Designing—Implementing—Operating (CDIO) real-world systems and products. The essence of IEC-CDIO is adopting product life cycle as the carrier, theory course as knowledge "point", module-level practical courses for direction "line", actual enterprises needs as application "surface" , the "point, line, surface" integrating project is running throughout the education system of information engineering. In the conception and design stage, "research-teaching" methodology is used to train innovating thought, and in the implementation and operation stages "cooperative teaching" methodology is used to train entrepreneurship, thereby improving the student's overall ability, this capability includes not only the individual's theoretical knowledge and professional skills, but also innovation and entrepreneurship. [4]

IEC-CDIO framework includes innovation and entrepreneurial goal-oriented training, integration and improvement of curriculum, revised syllabus, which remodeling teaching system as course - course group - integrated application based on CDIO standards, forming IEC-CDIO project throughout training practice such as smart meters, mobile payments and other items oriented new generation IT industry, and breaking down these large projects into relatively independent, but inter-related sub-projects for different courses shown in Fig. 3.

At the same time, IEC-CDIO project is repeated referenced in the theoretical teaching.

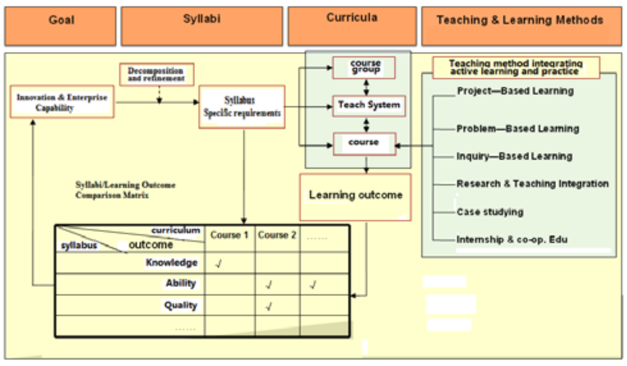
Goal
Syllabi
Curricula
Teaching & Learning Methods
Innovation & Enterprise Capability
Decomposition and refinement
Syllabus Specific requirements
course group
Teach System
course
Learning outcome
Teaching method integrating active learning and practice
Project—Based Learning
Problem—Based Learning
Inquiry—Based Learning
Research & Teaching Integration
Case studying
Internship & co-op. Edu
Syllabi/Learning Outcome Comparison Matrix
curriculum
syllabus
outcome
Course 1
Course 2
Knowledge
Ability
Quality

Figure 3. IEC-CDIO Framework

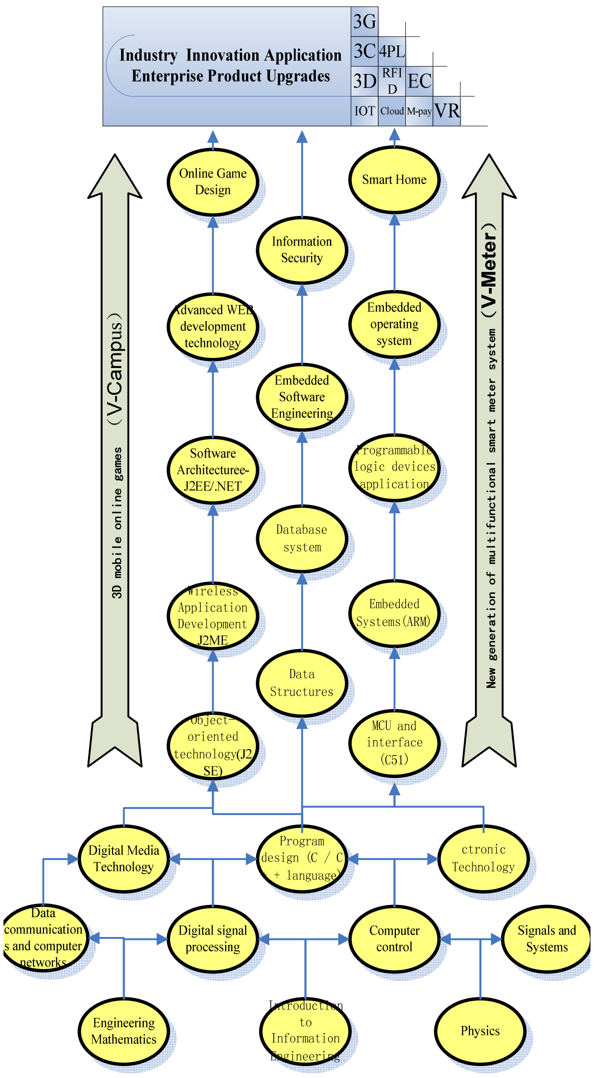
Industry Innovation Application
Enterprise Product Upgrades
3G
3C
4PL
3D
RFID
EC
IOT
Cloud
M-pay
VR
3D mobile online games (V-Campus)
New generation of multifunctional smart meter system (V-Meter)
Online Game Design
Smart Home
Information Security
Advanced WEB development technology
Embedded operating system
Embedded Software Engineering
Software Architecture-J2EE/.NET
Programmable logic devices application
Database system
Wireless Application Development J2ME
Embedded Systems (ARM)
Data Structures
Object-oriented technology(J2SE)
MCU and interface (C51)
Digital Media Technology
Program design (C / C++ language)
Electronic Technology
Data communications and computer networks
Digital signal processing
Computer control
Signals and Systems
Engineering Mathematics
Introduction to Information Engineering
Physics

Figure 4. IEC-CDIO Framework of Information Engineering

The module core curriculum, capacity requirements and teaching arrangements of Information Engineering major that pay more attention to embedded system development is shown in Fig. 4. A mobile online 3D game (V-Campus) is the one systematic IEC-CDIO project oriented mobile application development belongs to embedded software engineering throughout the wireless application model. And the other project of new generation of multifunctional smart meter system (V-Meter) oriented the Strong Smart Grid pay more attention the integrating development of software and hardware throughout the embedded system model. Both projects are parts of the Industry, academia and research cooperation project. All relevant courses are organic composition of the knowledge points to support IEC-CDIO Framework.

V-Campus is broken into V-Campus for 2D which integrating theory teaching by Digital Media course and engineering practice by Multimedia software applications, V-Campus for 3D which integrating theory teaching by Multimedia technology (Maya) course and engineering practice by Mobile game development, V-Campus Client which integrating theory teaching by Wireless Application Development (J2ME) course and engineering practice by Engineering training, and V-Campus Server which integrating theory teaching by Software Architecture (J2EE) course and engineering practice by Graduation design, the enterprise tutors train the college students throughout the start point Game design basis to Online Game Design end points, so that the available graduates can directly be recruited by the enterprise seamlessly shown in Fig. 5.

V-Meter is broken into V-Meter for C51 which integrating theory teaching by MCU System(C51) course and engineering practice by SCM curriculum design, V-Meter for ARM which integrating theory teaching by Embedded System (ARM) course and engineering practice by Embedded Systems Engineering Training, V-Meter for FPGA which integrating theory teaching by FPGA System course and engineering practice by Engineering training, and V-Meter for ucLinux which integrating theory teaching by Embedded OS course and engineering practice by Graduation design, the enterprise tutors train the college students throughout the start point Data acquisition and processing technology to Smart Home end points, so that the available graduates can directly be recruited by the enterprise seamlessly shown in Fig. 5.

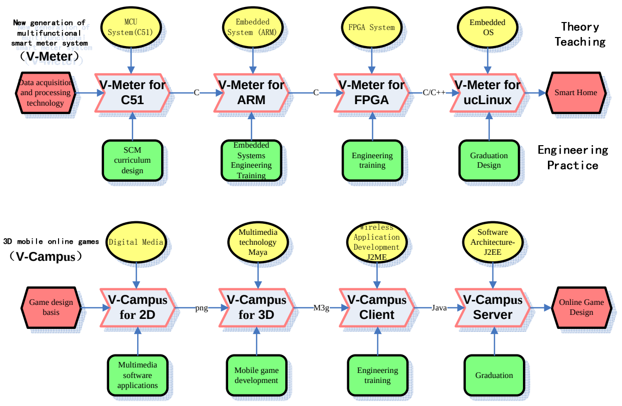

Figure 5. IEC-CDIO project break down.

Inquiry-based learning and Co-operative education is consistently throughout the IEC-CDIO project. In the stage of Conceive and Design phase, Inquiry-based learning accounts for more than Co-operative education because the requirements analysis is always the innovative ideas and possibility. Gradually, in the stage of Implement and Operate phase, Co-operative education accounts for more than Inquiry-based learning because the implement and operate are always the engineering activities.

When the IPD-CMMI model is applied to an existing organization's embedded software development processes, it allows an effective approach toward improving the developer profession capability. Eventually it became clear that the model is an abstraction of an existing system which could be applied to Higher Education processes. This gave rise to a more general concept that is applied to staff training. Further, How to implement the IPD-CMMI model of embedded software engineering under IEC-CDIO Framework is still a problem to solve. We proposed a systemic methodology integrating PSP/TSP/CMMI software engineering theory and practice into the IEC-CDIO Framework. With each discipline, the IPD-CMMI/IEC-CDIO defines activities to help manage and control your development process shown in Fig. 6.

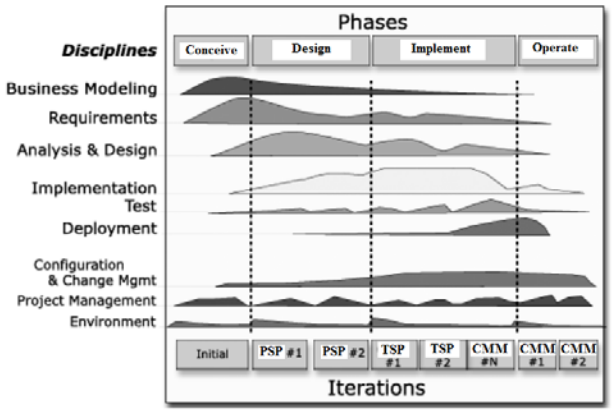

Figure 6. IEC-CDIO reference model

In study stages of IPD-CMMI, Students using the PSP to develop software follow defined processes and collect detailed metrics on the time required to complete mini project, the defects that were injected and removed at various stages in development, and the size of the finished product. These metrics are then analyzed using statistical methods, enabling students to produce highly accurate estimates based on historical data, track progress and quality of a project in progress, predict schedule impacts, and predict the quality of a finished software product. The PSP encourages students to quantitatively determine ways to improve their process [7]

In the Co-operative Studies of IPD-CMMI, students using the TSP which scales well and can be used by teams of 3 to 20 students to develop IEC-CDIO project of significant size and complexity.

In the engineering practice of, students using the CMMI for Development (CMMI-DEV), which addresses product and service development processes.

IPD-CMMI requires the collection and analysis of metrics at a very fine-grained level to IEC-CDIO project. Further, TSP requires teams to roll-up individual metrics to produce team metrics. Once data are collected at this level, statistical analyses of the data permit remarkable planning, tracking, prediction, and control of IEC-CDIO project [3].

These metrics collection and analysis processes, however, are not trivial. In any IEC-CDIO projects, tool support for the PSP and TSP become important considerations. Although studies have demonstrated that students can maintain their productivity when using the PSP without tool support, the "frustration factor" inherent with such an approach tries the patience of all but the most disciplined engineers, making PSP behaviors difficult to sustain. Ideally, IPD-CMM practitioners would like to have a support tool that:

- Allows data at the personal level to be collected quickly and easily, with minimal frustration.
- Can be integrated with existing development environments and project management tools.
- Allows individuals to collaborate on the execution of a process (even if they are geographically distributed).
- Allows data at the individual level to be rolled up to produce team-level or organizational-level metrics.
- Protects the privacy of individuals, and prevents unauthorized people from seeing or using their data.
- Supports powerful analyses of data at the individual, team, organizational, and enterprise levels, and allows existing (external) applications to access the data (while still maintaining the security mentioned above).

IV. SUPPORTING SYSTEM

IPD-CMMI Supporting System (IPDSS) has been designed from the ground up to support IPD-CMM processes. IPDSS process definition framework provides a great deal of flexibility and power for process automation by scripts and forms which are not hard-coded into the tool. Instead, they are dynamically loaded from simple XML files. Thus, it is possible to create your own custom process scripts, forms, and data that will be dynamically integrated into the IPDSS, even if your process definition is still immature and evolving. IPDSS is designed for supporting the following functions shown in Fig.7:

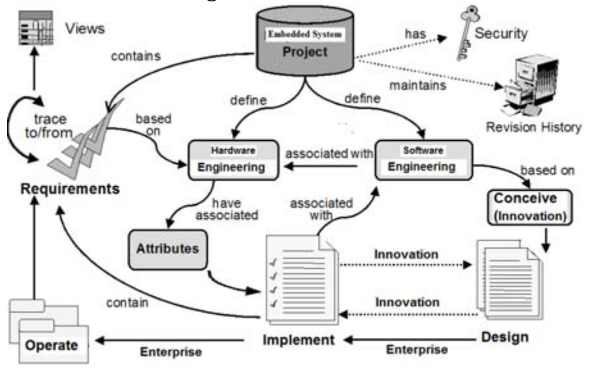

Figure 7. IPDSS system design

- Planning - Integrated scripts, templates, forms, and summaries, PROBE, earned value. A powerful, streamlined tool is provided to facilitate project launch planning sessions
- Data Collection - Time, defects, size; plan vs. actual data

- Data Analysis - Charts and reports aid in the analysis of historical data trends. Roll up time, defect, size, and earned value data from any number of individuals, team projects.
- Tracking - Powerful earned value support and defect repair.

IPDSS operates well in both Inquiry-based learning and irregular Co-operative studies owing to the RRS (reliable, reconfigurable, and scalable) design and implement architecture.

As to Planning, we adopted the Microsoft Project2010 as the external tools for the team which can support ordinary plan of IEC-CDIO project shown in Fig.8, such as V-Campus, V-Meter and so on, and support information exchange with the help of share point technology. [8, 9]

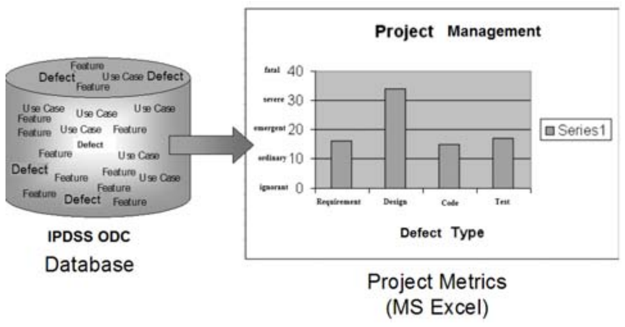

Figure 8. IPDSS integrating Microsoft project

As to Data Collection and Data Analysis, we adopt a set of Orthogonal Defect Classification (ODC) technology which creates a powerful software engineering measurement in embedded software development process. It works by leveraging information contained in software bugs often abundantly available in any software development process. [10, 11] ODC follows principles that categories never overlap each other and categories cover the whole defect space. Every defect can belong to one and only one category in the same dimension. Since the software defect distribution of different trades or organizations are very different, the defect classification can only be made, adjusted and perfected by senior experts in the corresponding trade. An ODC methodology was first tried as way to more objectively characterize and quantify defects so that priority could be given to addressing the larger problem areas. Teachers could do their work more quickly using ODC, form a coherent explanation of the facts, and make more convincing recommendations for software engineering process improvements. In order to increase the feasibility and effectiveness of the ODC application in collage software engineering, we give a software defect classification reference model which has carried on Omni-directional analysis and description of the reason and consequence of defects though the stages of verification and development respectively shown in Fig. 9. [6, 12]

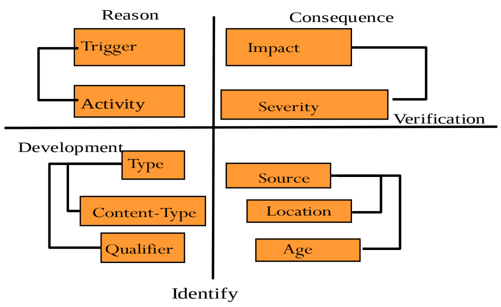

Figure 9. ODC attributes

TABLE I.
ACTIVITY ATTRIBUTE

Type	Activities	Triggering Factors
1	Requirements Review, Information Review, Test Plan Review	Accuracy, Completeness, Coherence, Organization, Feasibility, Style, Task Orientation, Graph Design/ Esthetics etc.
2	Design Review, Code Inspection	Design Coherence, Program Criterion, Compatibility, Synchronization, Internal Security, Language Particularity, Side Effect, Rare Situation etc.
3	GUI Review and Test	GUI Design Coherence, Windows Acquit, Screen Words/Characters, Input Facility, Navigation, GUI Behavior etc.
4	Unit Test	Single Path, Combination Path etc.
5	Function Test	Single Function Covered Verify, Single Function Boundary/ Valid Assurance, Multiply Function Serial Execution, Multiply Function Interaction etc.
6	System Test	Working load/Stress Test, Restore from Invalidation, Start/Restart, Hardware Configuration, Software Configuration etc.

Activity attribute describe the student's activities in the IPD-CMMI. Triggering factor and activities is not the same concept here. The contrasts between the activities and triggering factors are shown in Table I.

Trigger attribute pays attention to mechanism to detect the defect. The main triggering factor is shown in Table II.

Impact attribute analyzes the effect that defects had or would have on the customer. The following criteria should be followed to evaluate the influence of the defect during development, versification and maintenance. [13]

1) As for defect that is found in the course of product development, its impact should be judged as the level of consequence that would take place in the future.

2) As for the question happened when running, choose the actual impact of the deficiency caused by the defect.

3) When deficiency has the impact on customer in many aspects, choose the most serious impact category among them.

Impact attribute includes reliability, standardization, information security, function, performance, extension /Compatibility, maintainability, maneuverability/usability, installable, local language support etc.

Severity attribute is of the impact from the customer's perspective and when the fix is known, and is divided into

TABLE III.
TRIGGER ATTRIBUTE

Type	Trigger attribute		
	Trigger	Weight	Phase
1	The necessary environment or conditions for the exposure of defects.	15	Requirement
2	Thought of discovering defects in evaluation and inspection.	20	Design
3	Environment or conditions in other unexpected activities causing the system deficiency.	10	Test
4	Actual running environment in maintenance stage.	10	Implementation
5	Test use-cases.	40	Test
6	If there is no way of knowing the triggering factor, the defect is classified according to reappearance condition in the laboratory.	5	Test

five grades including fatal, emergent, severe, ordinary and ignorant according to relevant standard. Alarms can be Critical, Major, or Minor grade to signify the different levels of severity of a situation. When software deficiency happens, alarms will be sent.

Origin, content and type of defect are shown in Table III. The origin objects causing the question are the names of the entities including design specification or other earliest matters resulting in deficiencies, which need to be revised [6]. The defect type signifies the classification of technical key elements including the technological characteristics to revise the defect. The defect content type signifies the content nature of the defect. After the revising project is finished, origin object causing the question, defect type and defect content type can be classified according to the actual revising result, so as to ensure the uniqueness of the classification.

IPDSS can support arbitrary processes configuration and integrate arbitrary new process management tools, the default software process embedded according to CDIO standard. Every process is divided by a serial of activities and the customized well-configured IPD-CMMI process for IEC-CDIO project. The IPDSS also makes many graphical charts of various kinds of available analysis for IEC-CDIO project shown as Fig.10: [6, 14]

- Defect analysis - rates of injection and removal of defects in the different process phases, defect injection and removal percentages in the different process phases, and pie charts showing the distribution of defect types.
- Plan analysis - comparisons of planned time and size with actual planned time and size, planning error of both time and size, and breakdowns of the amount of time spent in the different development phases.
- Process analysis - comparisons of productivity rates, defect removal yields, and appraisal to failure cost ratios.
- Quality analysis - various looks at the quality metrics of defect removal yield, failure and appraisal cost of quality, and LOC reviewed per hour rates.

TABLE II.
CLASS OF DEFECT CAUSE

Type	Defect Cause Class		
	Origin	Defect Type	Defect Content Type
1	Requirement Analysis	(as the mentioned impact classification)	-
2	Design Specification, Code	Evaluation, Check, Arithmetic achievement, Function model or Class Design, Relevancy, Synchronization or Concurrency, Serialization Design, Interface, Local Language Support	-
3	Information Development	Edit, Technology, Navigation, Local Language Support	Reference, Task, Example Represent Means, Conception
4	Configuration Management Standard	Process Coherence, Delivery Conformity, Maintains Dependency, Assistant Script/Package, Feature Install or Upgrade Dependency	-
5	Test Plan/Project	Improper Resource Arrangement, Resource Arrangement non-optimization, Incorrect Agenda, Agenda non optimization, Improper infrastructure or tools, Document Criterion, Single Path Test, Combination Path Test, Single Function Cover Test, Single Function Cover boundary/Invalid Input Test, Multiply Function Serial Execute Test, Multiply Function Interaction Test, Working Load/Stress Test, Restore Test, System Start/Restart Test, Hardware Configuration Test, Software Configuration Test, Local Language Support	Feature List, Requirement Trace, Object Analysis, Resource Plan, Agenda, Strategy, Model, Methods, Use-Case, Code, Infrastructure or Tools etc.

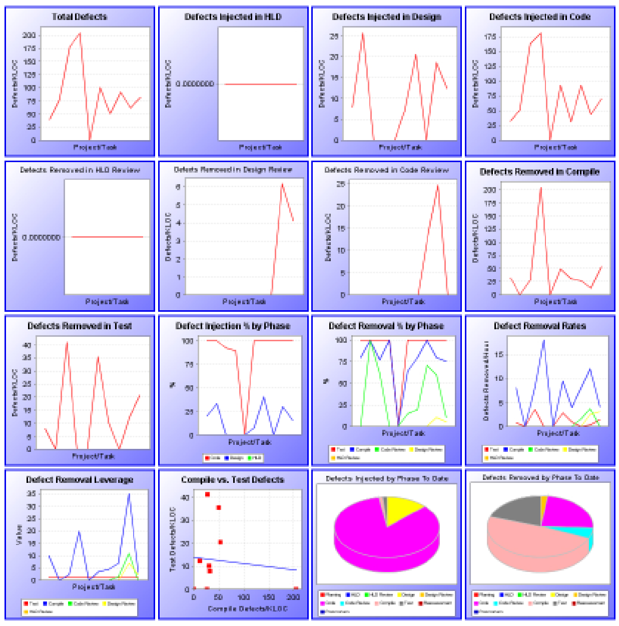

Figure 10. IPDSS project postmortem analysis

With the list of tasks and the calendar of available time in place, the IPDSS will calculate planned completion dates for each task. As students work, accurately log time against tasks should be collected. While looking at the earned value plan for a team, it might be necessary to look at the schedule for a single individual - for example, to see when their work is forecast to complete. According to the amount of data collected by IPDSS, we setup the baseline and millstone of every Co-operative education team which can be used to forecast the finish dates and ensure the success of Co-operative education projects shown as Fig.11. This is also helpful for the teachers to schedule the balance of study and work term.

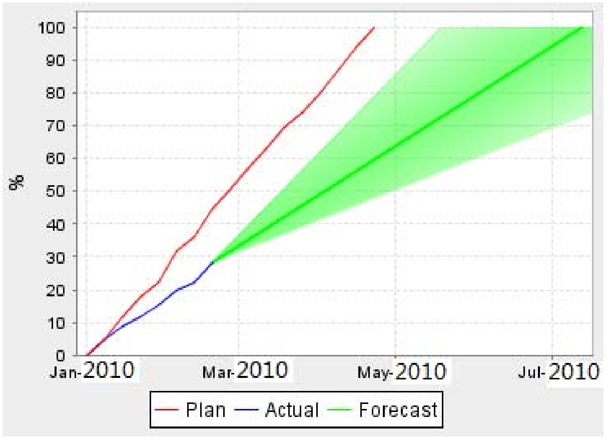

Figure 11. IPDSS project management

IV. CONCLUSION

IEC-CDIO is the core of developing the creative entrepreneurial talent in higher education of China, which is rethinking of the philosophy, aim and model of engineering education by fully combining advocating innovation of application and pursuing excellence of application with CDIO engineering education philosophy. On this basis, the hierarchy of cultivation orientation, aim, cultivation plan, course group, experiment and practice system, evaluation and assessments, teaching methods and quality assurance system of information engineering program are completely conceived and designed and an integrated talents cultivation scheme of combining knowledge, ability and quality oriented education, by which students are taught and layered naturally according to their aptitude, is developed. The proposed scheme is put into practice and the results have shown its effectiveness in Zhejiang Wanli University which is CDIO engineering education pilot universities in China.

In the future, we would like to extend our work in the following directions. 1) We will try other methods to measure CDIO effects in Innovation & Enterprise Capability training for IT talent long-term development. 2) IPD-CMMI model should be further improved and extended to enterprise applications to enhance embedded software engineering performance. 3) We will extend the scale of ODC database to establish data warehouse to support further intelligent decision-making.

ACKNOWLEDGMENT

This work is funded by the Ministry of Education Humanities and Social Sciences Fund (11YJC790141), Ningbo Soft Science Fund (2011A1012) and Ningbo Natural Science Foundation (2011A610106), Zhejiang new generation mobile Internet client innovation team(2012R10009-07), Major projects of national science and technology(2011ZX03002-004-02), and National Soft Science Fund (2011GXS2D036), Zhejiang Education Reform Project (2011-3), Education Reform Zhejiang Wanli University (2010).

REFERENCES

[1] N. Sharma, Kawaljeet Singh and D. P. Goyal, “Experience Based Software Process Improvement: Have We Found the Silver Bullet?” *Communications in Computer and Information Science. London*, vol. 141, pp. 71–80, February 2011.

[2] Li Kewen, Gong Lina, Kou Jisong, Predicting software quality by fuzzy neural network based on rough set , Journal of Computational Information Systems, v 6, n 5, p1439-1448, May 2010.

[3] T.M.Khoshgoftaar, P.Rebous, N. Seliya, “Software Quality analysis by combining multiple projects and learners,” Software Quality Journal vol. 17, pp. 25-49, 2009.

[4] M. Young, Programming embedded systems: with C and GNU development tools. O'Reilly, ISBN 9780596009830, 2006.

[5] Q.S.Chen, X.W.Chen and Y.Wu, “Optimization Algorithm with Kernel PCA to Support Vector Machines for Time Series Prediction,” Journal of Computers,vol.5, pp.380-387,2010

[6] P. Tie Jun, *Research of High-quality innovative and pioneering undergraduate training model of Software Engineering*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[7] P. Baraldi, and E. Zio. “A combined Monte Carlo and possibilistic approach to uncertainty propagation in event tree analysis,” *Risk Analysis*, vol. 28, pp.1309–1326, October 2008. “doi:10.1111/j.1539-6924.2008.01085.x”

[8] Dong Jianli, Ningguo Shi, "Multilayer Matter-Element Extension Synthesis Evaluation of Software Quality," IEEE, ICBECS 2010, pp.693-697.
[9] YANG FuQing, "Thinking on the Development of Software Engineering Technology," *Journal of Software (in Chinese),* 2005, 16(1), pp.1-7.
[10] YIN Ping, "The Analyze of Software Quality Evaluation Based on ISO(in Chinese)," *Software Engineering and Standardization*, 2005, No.12, pp.37-41.
[11] D. M. Ahern, C. Aaron, R. Turner. *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*. Addison Wesley, 2003.
[12] T. Menzies, J. Greenwald, and A. Frank, "Data Mining Static Code Attributes to Learn Defect Predictors," *IEEE Transactions on Software Engineering*, vol.33, pp.2-13, 2007.
[13] Burak, "Improving the Performance of Software of Defect Predictors with Internal and External Information Sources," Bogazici University, 2008.
[14] B. C. Andréde, P. Aurora and R.V. Silvia, "A symbolic fault-prediction model based on multi objective particle swarm optimization," *The Journal of Systems and Software*, vol. 83, pp. 868-882, 2010.

Tiejun Pan (Henan,1972-) received the MS and PhD degrees in modern mechanical engineering from Zhejiang University , Hangzhou, China, in 1997 and 2001, respectively. He is currently a Master Tutor associate professor in the Department of Computer Science and Information Engineering at Zhejiang Wanli University. His research interests include software engineering, embedded system, theory and application of networked control system.

Leina Zheng (Liaoning, 1981-) received the MS degree in School of Management from Wuhan University of Technology, Wuhan, China, in 2008. She is currently a Lecturer in the Department of Business at Zhejiang Wanli University. Her research interests include management engineering, Innovation & Enterprise education, and Performance Evaluation.