Measuring Quality of Software Engineering Education Processes

Saira Anwar, Wafa Tahir and Manzil-e-Maqsood
sairaanwar@gmail.com, wafatahir@yahoo.com and manzil@nu.edu.pk
National University of Computer and Emerging Sciences (NU-FAST)
Lahore – Pakistan

Abstract: *Improvements provide the basis of the quality. Ascertaining quality in education discipline is very invaluable. It is identified that models are always beneficial for the purpose of process improvement, like wise these are equally effective for improvement in the education sector. Software engineering methodologies guide in the development life cycle of software and are widely used in the software industry. The purpose of the current study is to define a framework for software engineering education processes. The model helps in evaluating the strength and weaknesses of the software engineering education program. The proposed associated measures shall determine the quality of Software engineering education processes. The proposed model shall help in software engineering program design as per industry practices. In other words, it shall enforce the maturity of software engineering education processes in institutes by collecting and analyzing the suggested metrics.*

Key Words: *Metrics, Measurement, Maturity model, education processes*

1. INTRODUCTION

Quality is required in every field and has gained paramount importance. Similarly, the desire of quality in the field of education exists.

The determination of success and failure of any product is no more subjective. The success of software is determined by the level of customer satisfaction it achieves, but success comes with quality, where quality of software is broadly measured in terms of maintainability and reusability. Now the question stands, how to build quality software? The major component is *Quality Human Resource* or *Quality Software Engineer*

To produce *Quality Software Engineers* we need *Quality Concepts* of the Engineers which are provided by *Quality Institutions* and *Quality Teachers.* Software engineering is taught at all institutions offering degrees in computer science. Courses that are offered vary in terms of number, content and teaching methodologies etc.

Thus we need software engineering education process improvement. Improvements provide the basis of the quality. It has been experienced that models are always beneficial for the purpose of process improvement. Process models play a significant role in the development of large software engineering projects. Choosing the adequate model is an important managerial decision for the success of a project. The process model determines the sequence of tasks required to accomplish the project.

Similarly, in an educational context, the pedagogical methods used are one of the more important concerns for a teacher to fulfill the main objective: teaching. The pedagogical methods used describe the sequence of tasks for teaching the topics of any discipline. To a certain extent, we can establish an analogy between software manager-model processes, and teacher-teaching methods [10].

This paper aims to identify the maturity model for the software engineering education program. The defined model shall help in evaluating the strengths and weaknesses of the software engineering education program. In order to do so, the related processes have been identified along the associated metrics.

We have based our model on the existing models of Capability Maturity Model Integration (CMMI) [1, 2, 3, 4, 5], Educational Capability Maturity Model (E-CMM) [6, 7], People Capability Maturity Model (P-CMM) [8], Portfolio, Program and Project Management Maturity Model (P3M3) [9].

CMMI integrates the different knowledge area. It is well known for evaluation of processes, emphasizing upon continuous process improvement, and publishing best practices in software development perspective.

ECMM proposes a model for quality higher education in general, P-CMM focuses on the workforce improvement in an organization, and P3M3's major target remains on the development of descriptive reference model for the organizations for providing more effective guidance of process improvement programs.

2. MODEL OBJECTIVES

The purpose of the software engineering education maturity model and its processes is to evaluate and improve the current practices of teaching through a structured study of its strengths and weakness. This shall help in prioritization of issues and impact. The degree of software engineering education satisfaction is also examined by mapping observations to practices.

3. OVERVIEW OF THE MODEL

The Software Engineering Education Maturity Model (SEEMM) can be used as the basis for improving software engineering education program. It defines five levels of maturity; their specifications are given in table 1, which are:

- Level 1 - initial process
- Level 2 - repeatable process
- Level 3 - defined process
- Level 4 - managed process
- Level 5 - optimized process

Table 1: Categorization of Structural components that comprise SEEMM

Maturity	Program / Course
Level 1 – Initial Process	Does the institute realize and recognize the importance of software engineering concepts required to build a software?
Level 2 – Repeatable Process	Does the institute ensure that each time the course is taught what basic contents must each instructor follow? (The basic contents are specified on the standard course outline)
Level 3 – Defined Process	Does the institute have its own centrally controlled software engineering education program and can each instructor flex with in these processes to suit the particular need and time?
Level 4 – Managed Process	Does the institute retain and obtain the specific measures related to software engineering education program management? Does the institution run an education quality management program to better predict the student's future performance in the industry?

For SEEMM these levels are described as follows:

3.1 Initial

At this level, software engineering education processes will be adhoc in nature. Few course contents are well defined and structured. Success depends mainly on instructor's self knowledge for teaching. No well defined criteria for instructor's selection are chosen.

3.2 Repeatable

Basic generalized set of contents are defined and well established. These contents are based on previous experiences. The basic schema of study is in place.

3.3 Defined

Proper course outlines are documented and practices are identified for enhanced level of understanding among students, and are incorporated. Methodologies of teaching are standardized.

3.4 Managed

Detailed measures of course outlines quality and education quality are collected and made possible to well understand the requirements of the course.

3.5 Optimized

Continuous improvement in structure of software engineering education is supported from the process and from piloting new ideas and structures with new and advanced technologies.

4. WHAT IS SEEMM ALL ABOUT?

The Software Engineering Maturity Model is a documented set of practices that enables evaluation and improvement of students' competencies, Instructor's skill set and Software Engineering Department (SED) in general. The SEEMM is a model for all institutes for their software engineering department. Its practices help to retain, grow, and nurture the software engineering department. As an inherent model of People CMM, SEEMM attracts, develops, motivates, organizes and retains talented people both in the form of instructors and students.

At the core of SEEMM lies a framework to define the software engineering department capabilities and practices, which helps continuous improvement. The whole department cannot get better overnight, but it can start getting better, piece by piece right away. Each operation can substantially and even dramatically improve. The SEEMM is structured as a staged model wherein such improvements happen through progressive, evolutionary steps.

5. THE IDEAL MODEL

SEEMM following the guidelines of its mappers, uses the approach of IDEAL model to define the process of improvement. IDEAL (Initiating, Diagnosing, Establishing, Acting, Learning) is based on Shewart-Deming cycle of Plan-Do-Check-Act or the PDCA Cycle.

6. FOCUS OF THE LEVELS

6.1 Level 1 – Initial

This level is referred as the initial process level and is the default way of teaching. The default key process areas are expected in this level of software engineering quality education. At this level the processes is highly dependent upon the individuals who perform the hit – or – miss type of practices, and quality software engineering is highly dependent on luck. The following process areas must exist:

- Course definition
- Course awareness

6.1.1 Aims and Objectives

At this stage the aim remains to make students aware of the benefits that they will achieve as practitioners after studying the software engineering course. The purpose is to make them familiar with the basic guidelines and principles that they should follow while developing the software. The concept is to make students aware of the dual role of software engineering i.e. it is a product and at the same time a vehicle for delivering the product.

6.1.2 Activities

The aim is to make the students acquainted with the key practices of software development.

- Students must be given the experience of working in a team.
- Students must be assigned case studies reflecting problem scenarios from industry in order to apply the software engineering approach.
- Students should be able to develop the basic set of work products as defined by the SE approach.

6.2 Level 2 – Repeatable

By institutionalizing basic software engineering practices and building the culture based on commitment, a certain orientation towards a disciplined execution is enabled. The following process areas are identified:

- Software Engineering Course Definition
- Software Engineering Education Program Organization
- Course planning, monitoring and Control
- Course configuration Management
- Software Engineering Education Program Management
- Course Risk Management
- Research Initiative
- Course Establishment
- Emerging software engineering department
- Hiring instructors
- Training instructors

6.2.1 Aims and Objectives

At level 2 of the software engineering maturity model the aim is to extend the understanding of software engineering practices and knowledge. The instructor should be capable of delivering the crux of the software development practices. The emphasis of the course should be on the technical activities that are required for software maintenance and enhancement.

6.2.2 Key Activities

The aim is to make the students acquainted with enhanced practices of software development and thus introduce two courses at best:

1. SE Course 1
2. SE Course II for advanced contents

The key activities are:

- The concept is to make student capable of planning and estimation.

- The students must be made familiar with the statistical quality control techniques of the software.
- The students are made aware of writing reusable and maintainable code.
- Students should be able to know the importance of research.
- Students should be given understanding of less ambiguous modes of representation.
- Students should be taught the techniques which reduce rework.

6.3 Level 3 – Managed

It is needed to determine and separate professionalism from the influence of commitment. The level 3 KPA enforces professionalism within the instructor and students. The aim of this KPA is to remove the inconsistencies in the software engineering education among all the teachers of the institute and inter institute. The guidelines are tailored according to the needs of the course and individual competencies.

The following process areas are defined

- Instructor's knowledge and skills analysis
- Detailed planning of the course
- Competency development of students
- Career oriented studies
- Research encouragement
- Discussion oriented course design
- Concept analysis

6.3.1 Aims and Objectives

At level 3 of SEEMM, it is required by the instructors to indulge the thought maintenance issues and problems identification among the students. The instructor should define its courses in such a manner that students get the experience of common problems they encounter and should be able to solve their problems. Thus it is needed to develop the career oriented environment among the students and their concepts should be analyzed.

6.3.2 Key Activities

At this stage it is necessary to introduce software engineering not just as a theoretical subject, but instead focus should be on developing the competency of the students using the practical approach. At this stage SEEP emerges as a whole paradigm not just composed of one or two subjects, but rather it introduces software engineering at the post graduate level and helps students in gaining the complete domain knowledge with supporting industry practices. The following activities are followed:

- Detailed course segregation is done resulting in the different fields and aspects of Software Engineering Education.
- Instructor must have industry experience.
- It is mandatory at this stage to have instructors who are not only good in teaching but at the same time can impart their experience to students.
- Instructors should be able to change the class atmosphere from a typical class to a discussion forum where everyone has his/her own opinion which matters.
- The students should be indulged with concepts and working norms.
- Instructor should be eager in research and should try to add this skill among the students.
- Students are given tasks to read the material on latest researches and developments and provide their judgments on this.
- Students are also trained in project management.
- The latest research journals and magazines should be provided to the instructors and the students.
- The SE department should have membership of reputable digital libraries like ACM, IEEE etc.

6.4 Level 4 – Managed Process

Level 4 is more associated with integrating professionalism within the culture of competencies. The judgment is much based on quantitative terms. The intent is to give students the in hand experience as industry norms demand, making them capable enough to deal with the industry practices. The following process areas are defined:

- Team based practices
- Competency Management
- Performance Alignment
- Workshops Conduction
- Course Management Metrics
- Research Department Establishment
- Course Quality Management

6.4.1 Aims and Objectives

At level 4 of SEEMM, the aim remains in further enhancing the student's skills and incorporate a quality control mechanism among students. The software engineering department should take steps so that Software Engineering practices become their role model while working in industry. At this stage the department and industry has no difference at all. Students in the department work for industry and support industry with their work.

6.4.2 Key Activities

At this stage following key activities are defined:

- The details of each course is further refined.
- Each student is given experience as he/she works within the industry.
- Students are given research topics to work on and the department of research is established to support them.
- It is mandatory for instructors and students to enhance their knowledge and for this workshops on current topics are conducted.
- Instructors and students are given research incentives in terms of pay revisions and grade up-gradation respectively.

6.5 Level 5 – Optimized Process

Level 5 is associated with continuous process improvement based on continuous innovation and reinforcement. This is the stage of progressive growth and focus is on the enhancement of each individual student it ensures the motivation of work among the students. The following process areas are defined:

- Individual enhancement
- Continuous innovation
- Performance alignment
- Continuous Research based innovation and enhancement
- Proactive Problem Handling

6.5.1 Aims and Objectives

At level 5 of SEEMM, the aim remains in the continuous improvement of individuals. The level focuses on the proper risk management of all courses and improves courses by the feedback they gain, introduce new subjects and technologies as they arrive and open new horizon of knowledge for students and instructors at all levels.

6.5.2 Key Activities

At this stage the following activities are defined

- The courses are continuously improved by looking in the details of the current research.
- The proactive problem solving is done by keeping in view the problems that can be encountered.
- To keep the focus on individual enhancement, it is necessary to update the knowledge and research areas.

Table 2: Mapping of CMMI, E-CMM, PCMM and P3M3 for SEEMM

Level	Focus	CMMI	E-CMM	PCMM	P3M3	SEEMM
Level 5 (Continuous Improvement)	Continuous Process Improvement	• Defect prevention • Technology innovation • Process change mgt	• Process Change management • Technology Change Management • Total Faculty Involvement • Documented Feedback • Defect prevention	• Continuous workforce innovation • Personal competency development • Coaching	• Proactive Problem Management • Technology Management • Continuous Process improvement	• Individual enhancement • Continuous innovation • Performance alignment • Continuous Research based innovation and enhancement • Proactive Problem Handling
Level 4 (Optimized)	Engineering process: Use infrastructure	• Organizational Process Performance • Quantitative	• Institutional Process Performance • Educational	• Mentoring • Organizational Competency management	• Management Metrics • Quality Management	• Team based practices • Competency Management

		Project Management	Quality Management • Quantitative Process Management. •	• Organizational performance alignment • Team building • Team based practices	• Organizational Cultural growth • Capacity Management •	• Performance Alignment • Workshops Conduction • Course Management Metrics • Research Department Establishment • Course Quality Management
Level 3 (Defined)	Product and process. quality: measure	• Requirements Development • Technical Solution • Product Integration • Verification • Validation • Organizational Process Focus • Organizational Process Definition • Organizational Training • Integrated Project Management for IPPD • Risk Management • Integrated Teaming • Decision Analysis and Resolution • Organizational Environment for Integration	• System approach of institution • Integrated Program Management. • Documented Process Management. • Intellectual Property Management. • Faculty Training • Student-Support Process.	• Knowledge and skill analysis • Workforce planning • Competency based practices • Competency development • Career development • Participatory culture	• Benefits Management • Transition Management • Information Management • Organizational Focus • Process Definition • Training, Skills and Competency Development • Integrated Management and reporting • Lifecycle Control • Inter-group coordination and networking • Quality Assurance • Center of excellence role development • Organization portfolio establishment	• Instructor's knowledge and skills analysis • Detailed planning of the course • Competency development of students • Career oriented studies • Research Encouragement • Discussion oriented course design • Concept Analysis
Level 2 (Managed)	Project management: Establish	• Requirements Management • Project Planning • Project Monitoring and Control • Supplier Agreement Management • Measurement and Analysis • Process and Product Quality	• Education Requirement Management • Degree Program and Course Planning • Degree Program and Course Monitoring and Control • Solution Provider Management	• Work environment • Communication • Compensation • Staffing • Training • Performance management	• Business Case Development • Programme organization • Programme Definition • Project establishment • Project planning, monitoring and control • Stakeholder management and	• Software Engineering Course Definition • Software Engineering Education Program Organization • Course planning, monitoring and Control • Course configuration

		Assurance • Configuration Management	• Education Quality Assurance • Faculty hiring Program		communication • Risk management • Configuration Management • Programme Planning and Control • Management suppliers and external parties	Management • Software Engineering Program Management • Course Risk Management • Research Initiative • Course Establishment • Hiring Instructor • Training Instructor •
Level 1 (Initial)					• Project Definition • Programme Management Awareness	• Course Definition • Course awareness

7. INSTITUTIONALIZATION

Institutionalization is one of the most important and necessary aspect of process improvement [1,2,3,4,5] and is an important concept within each maturity level. It implies the process is entrenched in the way the work is performed. A managed process is institutionalized by doing the following:

- Adhering to institute rules and regulations.
- Following established plans and course descriptions.
- Providing necessary and desired resources (including funding, students, instructors, and tools).
- Assigning responsibility and authority to the instructor for performing the process.
- Training the students for performing and supporting the course activities.
- Placing course contents and lecture slides in appropriate levels of shared folders.
- Identifying and involving relevant faculty members.
- Monitoring and controlling the performance of instructors and students.
- Reviewing the activities, status, and results of the course with higher faculty, and taking corrective action.

A defined process is institutionalized by doing the following:

- Addressing the details that institutionalize a managed process.
- Establishing the description of the defined process for the institute and software engineering department.
- Properly planning and collecting the improvement information.

A quantitatively managed process is institutionalized by doing the following:

- Addressing the items that institutionalize a defined process.
- Controlling the activities within software engineering department using statistical and other quantitative techniques such that product quality, service quality, and course performance attributes are measurable and controlled throughout the project.

An optimizing process is institutionalized by doing the following:

- Addressing the items that institutionalize a quantitatively managed process.

- Improving the courses contents based on an understanding of the common causes of variation inherent in the courses such that the process focuses on continually improving the range of course performance through both incremental and innovative improvements.

8. COMMON FEATURES

The software engineering education program or department in general can achieve progressive improvements in its maturity by first achieving stability at the project level and continuing to the most advanced-level, department-wide continuous process improvement using both quantitative and qualitative data to make decisions.

Four common features organize the generic practices of each process area. Common features are model components that are not rated in any way. They are only groupings that provide a way to present the generic practices [1,2,3,4,5]. Each common feature is designated by an abbreviation as shown:

- Commitment to Perform (CO)
  Groups the generic practices related to creating policies and securing sponsorship and good instructors for the course, course development and enhancement of student's skill.
- Ability to Perform (AB)
  Groups the generic practices related to ensuring that the instructors, institute infrastructure has the resources they need.
- Directing Implementation (DI)
  Groups the generic practices related to managing the performance of the students and instructors for managing the integrity of their course work and involving relevant faculty members.
- Verifying Implementation (VE)
  Groups the generic practices related to review by higher level faculty and incorporating the program of teacher's feedback and objective evaluation of conformance to course descriptions, procedures, and standards.

9. SUGGESTED METRICS

This section describes a set of base metrics to measure a few key process areas of the SEEMM:

- INSTEXP: Instructor's years of experience in the software industry
- TCROL: Instructor's years of experience in an Instructor's role.
- TECHSKILLS: The number of training programs held to enhance the technical skills of instructors in the current year.
- STTECHSKILLS: The number of workshops on latest technology held in the SE Department in the current year.
- LVARIANCE: The number of undelivered lectures. The number will show the variance between actual and planned estimates for deliverance of lectures.
- SUCCESSRATE: Percentage of students above average at the mid or end of semester.
- TECHSUCCESS: Percentage of computer systems in a specific lab equipped with all required software.
- HWSUCCESS: Percentage of computer systems equipped with all necessary hardware.
- SRESPONSE: Percentage of students satisfied with the Instructor's teaching style. (This metric can be gathered through Instructor's evaluation forms to be filled in by students at the mid and end of the semester to make future decisions)
- RPUBS: No of research publications in the current year either by students or instructors.
- CQUAL: Calculated by management mean rating for the attributes of course quality. The attributes consists of correct delivery, concept clearing, student satisfaction, contents completion, consistent completion, traceability etc.
- CCHNG: The number of changes in course contents during the course time span.
- CCMPLT: No of contents completed in the planned time.

10. FUTURE WORK

We have defined a limited set of metrics to quantify the SEEP processes. In future we intend to extend this set to enable measurement on all the levels of SEEMM. Moreover, instead of using lots of metrics we can come up with a few derived measures that can reflect the true picture of the SEEP maturity. Currently the model detail defining is done for up to level 2, and we believe to extend this in future

11. CONCLUSION

The Software Engineering Education Maturity Model (SEEMM) can be used as the basis for improving software engineering education program. It defines five levels of maturity processes: initial, repeatable, defined, managed and optimized. At the first level of software engineering education processes are ad hoc in nature. At level 2 the aim is to extend the understanding of software engineering practices and knowledge. At level 3 it is required by the instructors to indulge in thought maintenance issues and problems identification among the students. Level 4 is more associated with integrating the professionalism within the culture of competencies. Level 5 addresses continuous process improvement based on continuous innovation and reinforcement. We have identified key process areas for each of these levels based on the mapping with existing maturity models like CMMI, PCMM, ECMM and P3M3. Moreover, we have also identified a set of metrics to measure the quality of these key process areas. We believe that this model shall help software engineering departments in evaluating and improving the quality they deliver.

ACKNOWLEDGEMENT

Whatever the shortcomings in this study, they are ours and whatever its strengths, we owe to our friends and fellows. We are thankful to all those who pointed our mistakes and guided us for proceeding in the right direction.

REFERENCES

[1]. CMMI Team, Capability Maturity Model Integration, Version 1.1, CMMI for system Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI- SE/ SW / SS, V 1.1) Staged Representation, SEI, CMU/SEI-2002-TR-012, SC-TR- 2002-012, March2002

[2]. CMMI-SE/SW v1.1, Staged Model , SEI Technical Report CMU/SEI-2002-TR-004, December 2001 available at www.sei.cmu.edu/cmmi/models/v1.1ippd-staged.doc

[3]. CMMI-SE/SW · Staged Representation, Capability Maturity Model® Integration (CMMISM), Version 1.1. CMMISM for Systems Engineering and Software Engineering available at www.sei.cmu.edu/cmmi/models/v1.1se-sw-staged.doc

[4]. Carnegie Mellon University, Software Engineering Institute: “Getting Started with CMMI® Adoption” available at http://www.sei.cmu.edu/cmmi/adoption/cmmi-start.html

[5]. Dennis M. Ahern, Jim Armstrong, Aaron Clouse, Jack R. Ferguson, Will Hayes, Kenneth E. Nidiffer: “CMMI® SCAMPISM Distilled: Appraisals for Process Improvement”

[6]. Moazzam Baig, Sidra Basharat, Manzil-e-Maqsood, “A Maturity Model for Quality Improvement in Higher Education” 1st international conference on assessing quality in higher education ICAQHE 2006.

[7]. Tayyaba Tariq, Uzma Farooq, Manzil-e-Maqsood, “Educational Capability Maturity Model for Curriculum Design”

[8]. Raghav S. Nandyal, People CMM, Interpreting People CMM for software organizations, A guide for improving people related and workforce practices 2003

[9]. Office of govt. Commerce, Portfolio, Programme & Project Management Maturity Model (p3m3) 2006

[10]. Maria Isabel Alfonso and Antonio Botía, An Iterative and Agile Process Model for Teaching Software Engineering, Proceedings of the 18th Conference on Software Engineering Education & Training (CSEET’05).