Practicum in Software Project Management – An Endeavor to Effective and Pragmatic Software Project Management Education

MANZIL-E-MAQSOOD, TALHA JAVED
Computer Science Department
National University of Computer & Emerging Sciences
852-B, Faisal Town, Lahore
PAKISTAN

http://www.nu.edu.pk

Abstract: - One of the key factors influencing project success or failure is project management. Unfortunately, effective management of software projects is not in practice; what is actually being practiced varies significantly from what is advised in the available literature. In order to improve performance in the field of software project management, there is a dire need to formally educate prospective project managers in both the theoretical and practical aspects of managing software projects. This paper focuses on the formulation and execution of Practicum in Software Project Management, a graduate course that aids students in learning practical aspects of software project management. This course has been a part of the Masters in Software Project Management curriculum at National University of Computer and Emerging Sciences (NUCES), Lahore, Pakistan since 2001. We discuss the course in light of the major software project management activities recommended in literature. A comparison of the course with the Portfolio, Program and Project Management Maturity Model (P3M3) has been done to allow us to 1) assess the maturity of this course in terms of software engineering project management processes, and 2) assist us in identifying and highlighting the areas needing further improvement in terms of teaching, practice and industry needs. The comparison is based on the key process areas applicable to our course and shows that Practicum in Software Project Management is 90% capable at the Repeatable and 81% capable at the Defined levels of the P3M3.

Key-Words: - Software Project Management, Education, Software Engineering, Training, Maturity Model, Assessment.

1 Introduction

Software project management is a very well-known, yet quite undermined area in the field of software engineering. Evidence-based studies in the software industry, reported many a times by renowned practitioners and third part evaluators attribute project management malpractices to quality crisis [1]. These studies indicate that only 20% of large software systems are implemented on time and approximately 2/3rd of those experience cost overruns approaching 100% [9].

Over the past few years, much emphasis has been made on developing state of the art curriculum for teaching project management skills such as project planning, project organization management, work breakdown structures and scheduling, project staffing; project control, managing multiple projects, controlling project scope, project tracking and project close-down [1, 4, 5, 7, 11, 12]. However, as Reif and Mitri [1] put it, “successful software project management, like programming, is not a skill that students will master from a single course of instruction”. Teaching project management requires teaching students the capability to mutually explore technical skills and ‘soft’ skills. The dimension of soft skills includes people’s management, management ethics, skills, and on-the-job experience.

Managing software development is different from managing the construction of a building. It entails not only the requisite knowledge of typical project management activities, but also the expertise in transforming intellectual thought and ideas into concrete technical realm. Managing software projects involves making use of: 1) business domain knowledge (that changes from project to project, unlike construction), 2) technical knowledge that needs to be refreshed as the tools and technologies evolve, and 3) working with people from diverse backgrounds. With the new dimension of global software development, the issues in software project management have become more pronounced.

Of all the software organizations in the world, 85% are small scale, with fewer than 50 employees [20]. These organizations contribute to a major portion of the overall software economy, and unfortunately do not have the capability to improve upon their project management capacity and education. They cannot select, support and grow their project managers. Therefore, the people who rise to the level of project management are generally those having development experience but no formal training towards project management skills. Inadequate software project management practices and approaches are thus observed in a large sphere of software industry, leading to failed projects –a dilemma that has been the crux of project management reports.

In light of aforementioned, it is more important than ever to educate and grow software managers right at the student level, and to teach them how to manage the complexities of real world projects. Dennis and Cynthia [3] report that this challenge has been recognized across the computing curriculum with project management recognized as one of the core concepts. According to them, project management concepts are taught both in dedicated project management classes as well as in other courses. However, students generally realize the benefits of effective software project management when they are involved with a real time, large software project, because real project offer tremendous opportunities for learning. With the formal project management education serving as pre-requisite knowledge, students can enhance their practical skills and learn about the concepts that are typically hard to grasp in class.

In order to improve performance in the field of software project management, there are two principal targets to set: firstly, to develop better project management skills and practices, and secondly, to improve the organizational capability in project management processes. Very few educational institutes provide formal higher degree programs in software project management [14, 15]. National University of Computer & Emerging Sciences, Lahore, Pakistan, is among those very few universities that runs a Masters in Science program in Software Project Management [18].

In this paper, we present the structure of a course titled: Practicum1 in Software Project Management, a graduate2 course offered to students enrolled in MS Software Project Management degree program. The primary intent of this course is to simulate real project management environment and to make prospective managers realize the actual issues that arise while managing software projects. For this purpose, each student is required to handle an undergraduate3 final-year project as a project manager. This exercise provides students an opportunity to apply their managerial skills and learn from their mistakes. Apart from this, training sessions are conducted on MS Project to help students put theoretical concepts, acquired through related Software Project Management4 course, into practice so as to value the issues associated with real software project management.

PM Solutions [30] has developed a Program and Project Management Model (P3M3) [16] based on is based on the five maturity level framework of SEI's Capability Maturity Model Integrated (CMMI) [17]. The purpose of this model is to provide guidelines for establishing organization-wide project management programs, and to enable organizations to assess and improve their project management maturity against the key process areas described in the model.

In this paper, we compare the processes and detailed activities of Practicum course with P3M3 Model so as to assess the capability of this course for effective software project management education. We have focused on the process goals, approaches and deployment activities advised by the model. The results of the comparison show that this course enables project management students to practice 90% of the key process areas recommended in Repeatable and 81% of those in Defined levels of P3M3.

The structure of the paper is as follows: Section 1 begins with background literature review on software project management education. Section 2 outlines the Practicum course in light of the recommended software project management practices. Section 3 provides a detailed comparison of Practicum with Portfolio, Program and Project Management Model (P3M3), followed by limitations of the course, future directions and conclusions.

2 Literature Review and Problem Elaboration

The industry and academia have not yet reached a consensus on what constitutes formal software project management education and what should be left to real life and on-the-job training [5].

In order to produce successful future software project managers who can survive and compete in software industry's stringent professional environment, the academia must design curriculum that address both the theoretical as well as practical and close to real life aspects of managing software projects. Software industry has been repeatedly providing feedback on the quality of graduates produced by the academia. In terms of software project management, one of the most noticeable problems reported by the IT industry is that the students graduated with higher degree in project management are either management savvies, lacking technology management skills, or are techies, equipped with only robust technical skills [5, 21]. A blend of both qualities in one graduate is hard to find, and if it is found, it is mostly attributed to the individual's personality.

[1] For the sake of readability, we refer to Practicum in Software Project Management as Practicum throughout the rest of the paper.

[2] We use the term 'graduate' for Masters level program of study (18 years of education) at NUCES. It can be used interchangeably with the term 'postgraduate' used widely in UK education system. We offer our MS Software Project Management program in two modes: full-time or part-time. It is mandatory for all graduate students enrolled in the program to have at least a prior industry experience of two years.

[3] We use the term 'undergraduate' for Bachelors level program of study (16 years of education) at NUCES. A typical undergraduate semester offered at NUCES spans about 20 weeks.

[4] Software Project Management is a 3 credit hour graduate course offered in Masters in Software Project Management degree program at NUCES, Lahore. It is a pre-requisite to Practicum.

For software managers to add value, steps must be taken to enable them to [5, 12, 21, 22, 24, 25, 26]:

- Understand fundamental practices involved in the development, deployment, and retirement of software systems.
- Identify life-cycle models, understand their differences, and know when to use the myriad tools in their software toolbox.
- Grasp software development concepts such as quality management, requirements analysis, configuration management, information integration, and software metrics.
- Distinguish sound development practices from ad-hoc coding, negotiate solid contracts with clients, and make sound business decisions about software assets.

Universities around the world are keen in determining the appropriate methods to teach pragmatic software project management. According to Reif and Mitri [1], integrating project management instruction in relevant non-technical and technical courses would enable us to produce effective project managers who are able to reduce project implementation delays and cost overruns.

Training professional resources, especially project managers, is an activity that the software industry is least willing to contribute to. When students graduate with higher degrees in project management, they are not placed directly at project management positions. Their project management expertise is scrutinized carefully before they could be assigned a significant leadership role. Normally, organizations prefer to hire candidates who already possess the skills and knowledge to succeed [5]. To align the software project management coursework to industry requirements, we need to have a clear understanding of:

a. The project management malpractices that lead to project failures in real project environments,

b. The project management best practices that have been recommended by renowned software engineering practitioners, industry professionals and project management standards

A basic and necessary set of activities which if taught and practiced in the university environment can assist freshly hired project managers in performing according to industry requirements.

2.1 Software Engineering Management (SEM)

Theoretically a host of different standards and guidelines are available that prescribes various methodologies for development and improvement of software development life cycle processes. While few would specifically target software development for industry projects, guidelines provided by SWEBOK [19] were found congenial for reference purpose in order to develop processes for execution of academic projects especially for the Practicum course.

The Software Engineering Body of Knowledge (SWEBOK) defines guidelines pertaining to ten different areas of software development life cycle. Six further sub areas defined by SEM are [20]:

- **Initiation and scope definition**, which deals with the decision to initiate a software engineering project
- **Software project planning,** which addresses the activities undertaken to prepare for successful software engineering from a management perspective
- **Software project enactment,** which deals with generally accepted software engineering management activities that occur during software engineering
- **Review and evaluation,** which deal with assurance that the software is satisfactory
- **Closure,** which addresses the post-completion activities of a software engineering project
- **Software engineering measurement,** which deals with the effective development and implementation of measurement programs in software engineering organizations

2.2 Software Project Management Best Practices Recommended in Literature

Evidences of project management malpractices are reported throughout the software engineering literature. Some of the major activities that jeopardize a software project are:

- Excessive schedule pressure
- Changing user needs
- Lack of technical specifications
- Lack of a documented project plan
- Inadequate change control
- Requirements creep
- Inaccurate metrics
- Inadequate measurement
- Inaccurate cost estimation
- Insufficient senior staff on projects
- No project management methodology
- Reliance on new technology without testing
- Quality mismanagement
- Lack of user involvement
- Inadequate understanding of customer requirements
- Miscommunication among project staff
- Lack of assimilation of lessons learned from past mistakes

- Ill-defined non functional requirements
- No risk assessment and management
- Inadequate software development methodology

Table-1 displays a minimum set of software project management practices extracted from various up-to-date papers, articles and surveys in literature. Educating students on these basic set of practices can enable the academia to produce better project managers, and provide trained project management resources to the industry.

Table 1: Software Project Management Best Practices

Necessary Software Project Management Practices Recommended in Literature [Ref: 2, 12, 13 22, 23, 24, 25, 26]
Realistic project planning and estimation
Understanding of client's problems
Clear requirements specifications
Scope management and control
Team leadership and decision making
Sound software development methodology
Good communication / relations with project team and client
Morale boosting rewards / performance appraisals
Project status reporting
Automated tool usage
Continuous risk management
Change control and baseline management
Assimilating and acting upon the lessons learned from past mistakes
Measuring process and project metrics
Use of mature technology

It is not however possible to educate students on the aforementioned best practices solely through a theoretical approach. A practical approach, combined with the theoretical one can give better insights into identifying and controlling project management problems while managing a software project. In the next section we describe how we have developed our graduate course that incorporates both the theoretical and practical aspects of managing software projects.

3 Practicum in Software Project Management Course

Practicum in Software Project Management is offered as a core course in the university's Masters in Software Project Management program. It is designed as a twin semester graduate course-work split into a one credit-hour Practicum-I and a following two credit-hour Practicum-II. The first half of the course, i.e., Practicum-I is offered in fall semester and the other half in spring every year. Together these are envisioned as a practical implementation of the tools and techniques of software project management taught in related pre-requisite graduate course on Software Project Management [18].

The course has been designed keeping into consideration the guidelines provided in Software Engineering Body of Knowledge (SWEBOK) as well as software project management best practices, described in section 2 above. Detailed descriptions of the processes for the course are placed at [18].

3.1 Process in a nutshell

Each graduate student registering in Practicum course is required to assume the role of a project manager and manage the execution of a year-long undergraduate final project. The intent is to simulate a real project management environment and to make these prospective managers realize the actual issues that arise while managing software projects. Practicum course instructors ensure that the enrolled graduate students are provided proper training on project planning, resource allocation, estimation and project progress monitoring and control so that they could efficiently manage the assigned projects.

When this course was offered for the first time in the fall 2001, graduate students were required to manage two year-long projects at the same time in order to experience the flavor of program management. However, this exercise did not prove fruitful because the graduate students were loaded with immense pressure both due to the Practicum course and the other course load. Therefore, we cut down the project size to one in 2005. Since then the graduate students are assigned only one project to manage in each Practicum phase. In addition to providing project managers an opportunity to apply their managerial skills and learn from their mistakes, Practicum also confronts the undergraduate students to counter schedule pressures, respect the team hierarchy, learn communication ethics and abandon ad-hoc work practices according to the management approach directed by their acting project managers. This helps them in experiencing working in an environment similar to what they will face as they step into the local software industry.

We have developed a complete set of processes for the execution of Practicum course [27]. Some of these are:

1. Stakeholders roles and responsibilities
2. Quality assurance in software development lifecycle
3. Final year projects initiation process
4. Process implementation
5. Project initiation process
6. Deliverables submission process
7. Project status reporting process
8. Performance reviews process
9. Project closeout process

Appendix A presents 'Quality assurance in software development lifecycle' process.

3.2 Key Stakeholders

The key personnel simulating project stakeholder roles are:

- Graduate student enrolled in Practicum course – acting as software project manager
- Undergraduate students, in a group of up to 6 students working on their final year projects – acting as software project team members
- Faculty members of the Computer Science department, who are actually the project advisors of final year projects – acting as project client. There may be more than one faculty advisor for a project, causing a challenging situation in overall project communication infrastructure.

3.3 Roles and Responsibilities

The project managers are held responsible for the successful execution of final projects from initiation through close out. They carry out these tasks in collaboration with the project advisors and Practicum instructor. Figure-1 depicts an overall configuration of the Practicum course, where each stakeholder is associated with some activities, and related activities are marked by arrows. Details of the roles and responsibilities of each of the project stakeholders are provided in [27].

3.4 Undergraduate final year projects

Undergraduate students take up their final projects during the fourth and final year of their Bachelors in Computer Science program. Each final year project spans one complete year of execution, and the project activities are divided in two semesters. Final year projects are usually of the following three types:

- Application based projects – that usually include typical software engineering activities such as project inception, requirements specification, design, testing, complete system documentation and delivery.
- Research based projects – that are concerned with the development and/or refinement of a new idea in computer science and related fields, and typically involve thorough literature review and analysis of new techniques.
- Research and development (R&D) projects – that involve a significant amount of research prior to developing a prototype system based on that research.

The activities involved in the execution and implementation of these projects significantly vary from one another. The project deliverables therefore also vary. It is the responsibility of the assigned project managers to decide about the software development methodologies suitable for the execution of these kinds of projects and modify the project deliverables accordingly.

3.5 Training

The aim of Practicum course is to provide training to project managers on a number of project management practices from a practical standpoint so that they could learn to better apply the theoretical knowledge acquired from the Software Project Management course. Some of the important training elements are:

- Automated project management tool, MS Project.
- Project management life cycle activities
- Work breakdown structures and identification of critical project paths
- Effort and task estimation and scheduling
- Project planning and tracking
- Interim baselines
- Effective communication channels among project stakeholders
- Software development methodology for projects assigned
- Gather in-process and project metrics

Training sessions commence as soon as the semester begins. Practicum classes are held every week for the first 6 weeks. Later, classes are held fortnightly so as to allow students focus on applying what they have learned on their projects.

3.6 Course Deliverables

Practicum requires graduate and undergraduate students to submit their project deliverables according to a schedule made available at the start of each semester. Table 2 presents the deliverables required by the project managers in Practicum I and II.

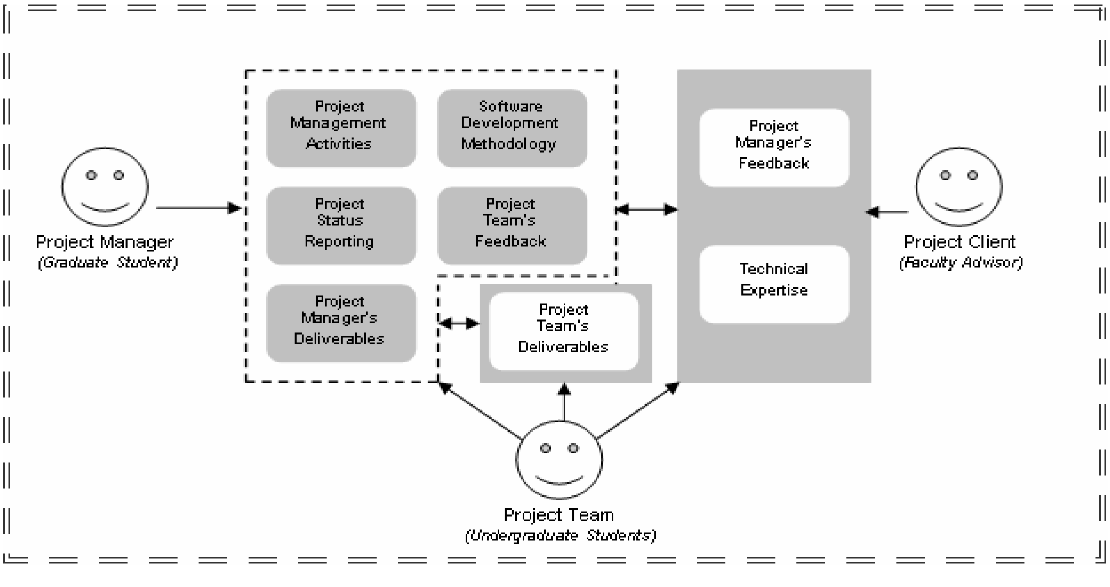


Figure 1: Practicum Course Stakeholders

Software documentation templates, typical to the waterfall development lifecycle, are provided to the undergraduate students. The basic software artifacts required of a team working on an application based project are:

- Requirements specification document
- High level and detailed design document
- Test cases document
- Bug status report
- Final project report

Project managers are advised to modify these templates according to the methodology they select for the implementation of their projects. A complete list of deliverables expected from students working on all three types of projects can be found at [18].

3.7 Course Evaluation

The Practicum instructor evaluates project managers based on pre-determined evaluation criteria (Table 3) as well as the quality of deliverables submitted (Table 2). Undergraduate students however are evaluated jointly by the project managers and project advisors. The project advisors are requested to consult student project grades with the assigned project managers each semester.

Table 2: Practicum Deliverables

Practicum Deliverables for Project Managers (All mentioned deliverables are required in both Practicum I and II, unless explicitly stated)
Weekly Meeting Minutes For 12 weeks excluding the weeks in which midterm and final exams are held. The Project Initiation and Project Close out meeting minutes are mandatory and hold more assessment points.
Initial Project Plan Includes the baselined project schedule, work breakdown structure resource allocation, resource leveling, planned effort, cost estimation (optional), identification of critical path.
Risk Analysis Including risk identification, assessment and mitigation strategies.
Project Estimation Function point based, use case based, or activity based estimation, depending on the nature of the assigned projects.
Quality Plan Deliverable of Practicum II. Includes a plan for software testing and quality assurance activities that the project managers deems suitable for their assigned projects. Includes the testing strategy, assignment of resources to test software modules built during the coding phase, and metrics, such as defect density, for assessing quality of software developed. All the testing activities are exercised based on the quality plan.
Final Project Plan Includes the final and updated version of the project plan including interim baselines, actual versus planned work, schedule and resource allocations. Provides the progress of the actual project activities against planned ones.
Updated Risk Analysis Includes the risks that occurred during the course of the project and how the project manager assessed and mitigated them according to the planned mitigation techniques provided in the initial risk analysis document.
Monthly Status Reports At the end of each semester, project managers are required to submit the progress of their assigned projects in a status report. The report gives an overall picture of the progress of the project in terms of actual versus planned schedule, work, scope and resource allocation. The report also provides project and process metrics to assess project performance.
Quality Report Includes a detailed description of the chosen software development methodology, project management methodology, configuration management strategy, data collection strategies adopted, overall project progress and quality metrics, and quality assurance activities practiced throughout the course of the semester.
Project Close-out Report Deliverable of Practicum II. Includes the minutes of the project close out meeting, a brief discussion on how the project progressed throughout the year, team motivational strategies undertaken during the year, and lessons learned.

Table 3: Project Managers Evaluation Criteria

Project Manager Evaluation
Feedback from Project Advisor & Team (Mid Semester) Feedback from Project Advisor & Team (End Semester) Weekly Status Reports & Meeting Minutes Class Attendance/Participation
Mid Semester Evaluation Domain Understanding Project Plan (initial) Risk Analysis & Project Estimation Monthly Status Reports I and II
End Semester Evaluation Domain Understanding Final Project Plan Quality Report Updated Risk Analysis Monthly Status Report III

4. Assessing the Maturity of Practicum with Portfolio, Program and Project Management Maturity Model

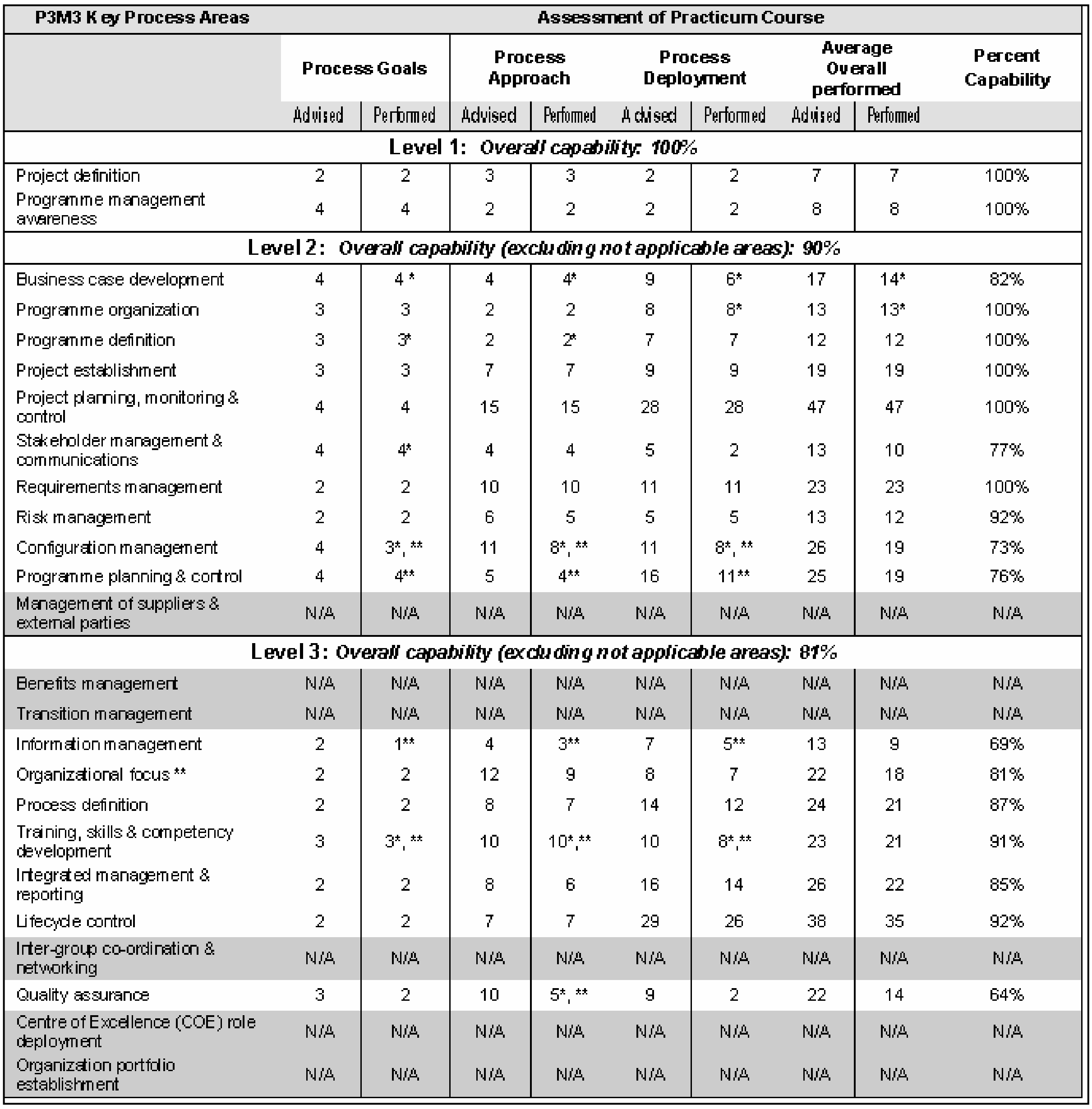
The Portfolio, Programme & Project Management Maturity Model (P3M3) is an enhanced version of the Project Management Maturity Model (PMMM) [13] developed by PM Solutions. The model is based on a five level maturity framework that is the foundation of Software Engineering Institute's (SEI) Capability Maturity Model Integrated (CMMI) [17]. These levels constitute the structural components that comprise the P3M3 [16]. In addition to PMMM and CMMI, the model integrates industry-leading standards for project and process management presented in Project Management Institute's PMBOK Guide [12]. The primary intent is to provide a comprehensive, straightforward, and easy-to-follow plan for advancing organizational project management maturity, growth and excellence.

4.1 Methodology

The basic purpose of Practicum course is to provide effective and pragmatic software project management education to graduate students aiming to become prospective project managers. With this purpose in mind, along with the motivation to improve this course, we assessed it with P3M3 model. The assessment involves comparing 1) the process as followed by all stakeholders of the Practicum course, i.e., graduate students, undergraduate students, faculty members and the Practicum instructor, and 2) management of the course at the institutional level.

At Repeatable maturity level, project management processes are retained during all times, and projects are performed and managed according to their documented plans. Project status and the delivery of services are

Table 4: Assessment of Practicum with P3M3 Model



P3M3 Key Process Areas	Assessment of Practicum Course								
	Process Goals		Process Approach		Process Deployment		Average Overall performed		Percent Capability
	Advised	Performed	Advised	Performed	Advised	Performed	Advised	Performed	
Level 1: ***Overall capability: 100%***									
Project definition	2	2	3	3	2	2	7	7	100%
Programme management awareness	4	4	2	2	2	2	8	8	100%
Level 2: ***Overall capability (excluding not applicable areas): 90%***									
Business case development	4	4 *	4	4*	9	6*	17	14*	82%
Programme organization	3	3	2	2	8	8*	13	13*	100%
Programme definition	3	3*	2	2*	7	7	12	12	100%
Project establishment	3	3	7	7	9	9	19	19	100%
Project planning, monitoring & control	4	4	15	15	28	28	47	47	100%
Stakeholder management & communications	4	4*	4	4	5	2	13	10	77%
Requirements management	2	2	10	10	11	11	23	23	100%
Risk management	2	2	6	5	5	5	13	12	92%
Configuration management	4	3*, **	11	8*, **	11	8*, **	26	19	73%
Programme planning & control	4	4**	5	4**	16	11**	25	19	76%
Management of suppliers & external parties	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Level 3: ***Overall capability (excluding not applicable areas): 81%***									
Benefits management	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Transition management	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Information management	2	1**	4	3**	7	5**	13	9	69%
Organizational focus **	2	2	12	9	8	7	22	18	81%
Process definition	2	2	8	7	14	12	24	21	87%
Training, skills & competency development	3	3*, **	10	10*,**	10	8*,**	23	21	91%
Integrated management & reporting	2	2	8	6	16	14	26	22	85%
Lifecycle control	2	2	7	7	29	26	38	35	92%
Inter-group co-ordination & networking	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Quality assurance	3	2	10	5*, **	9	2	22	14	64%
Centre of Excellence (COE) role deployment	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Organization portfolio establishment	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

* Performed in an academic context with limited resources. Some of the vast numbers of resources mentioned in the actual P3M3 model cannot be directly mapped on to our academic program structure.

** The practicum office is responsible for these activities with an organizational wide perspective visible to management at defined milestones. The minimum process discipline is in place to repeat earlier successes on projects with similar applications and scope [13, 17]. The focus of Defined level is to improve and mature the standards defined in Repeatable level over time. The standards, process descriptions, and procedures for a project are tailored from the organization's set of standard processes to suit a particular project or organizational unit [13, 17].

We believe that our course already stands past the initial or ad-hoc level of P3M3 model. Therefore, we assessed it with respect to all the key process areas at P3M3's level 2 – Repeatable, and level 3- Defined. We have not gone beyond this level because currently the more advanced process areas are not addressed by the course.

Our assessment is based on the available course process and guideline documents [27], however due to the limitation of page numbers, we have not presented all of them here, except 'Quality assurance in software development lifecycle' process in Appendix A. We adopted the following methodology for the assessment of Practicum course:

- Within each key process area, we have focused on the major goals, approaches, and deployment activities to achieve those goals.
- We developed an assessment grid (Table 4) for each selected key process area, its constituent 'advised' goals, approaches and deployment activities. The table shows the number of goals, approaches and activities recommended by the P3M3 model.
- For each of the advised goals, approaches and deployment activities, we have analyzed the processes followed in our course. Anything that the course follows according to the advised goals, approaches or deployment activities raises the count of 'activities performed' by 1 (Table 4).
- Based on the number of actual goals, approaches and deployment activities followed by the course as opposed to the recommended or advised ones, we have provided a 'capability percentage' that determines the percentage to which the key process areas are followed in our course.
- With the assumption that a capability of more than 80% means that the elements of a key process area are stable, we have given the verdict for the capability of our course.

Since this course cannot cover all the perspectives intrinsic to organization-wide project management establishment, some processes or sub-processes are not applicable in an educational context. These are highlighted in grey in Table 4. Similarly, some processes such as, business case definition, program definition, stakeholder management and communications, configuration management, program planning and control, training skills and competency development, information management and quality assurance, are followed in a pure academic context with limited resources, therefore they are indicated with an asterisk (*). Moreover, the processes for which the Practicum instructor is responsible in terms of program management are indicated by double asterisks (**).

4.2 Assessment Results

The assessment shows that Practicum course currently implements 90% of processes at the Repeatable level and 81% of processes at the Defined level of Portfolio, Program and Project Management Model. These percentages have been calculated for only those processes are those that are applicable to the course. For example, the process titled: management of suppliers and external parties' is not applicable to the course because all projects and resources are internal to the university.

The percentage capability of two key process areas at the Repeatable level is less than 80%; these are: Stakeholder management and communications, configuration management, program planning and control. Since the stakeholders in this course are faculty advisors, who either own the projects or provide technical expertise on their projects, therefore the issue of categorizing their groups, addressing their interests and needs is not possible. Therefore, in this context, development of a stakeholder communication plan is not necessary.

At the moment, configuration management strategies are broadcast to all stakeholders at the start of each semester, but it is observed that they follow their own methods. Moreover, currently, a configuration management plan is not required at the project management level, which is the reason why some of the project teams do not follow any methodology at all. We need to improve upon this area by making it mandatory for all students to use configuration management tool and follow an action plan.

The program planning and control is basically done at the Practicum instructor level, because each of the project managers is involved in managing the execution of only one final project. Currently a resource management strategy, benefits realization plans and dependency network models are not made by the Practicum instructor, which accounts for the reduction in capability.

At Defined level, five of the key process areas are not applicable to the course. The remaining seven contribute to 81% of capability of the course, out of which 'Quality assurance' contributes the least percentage, i.e., 64%, because the process area focuses on reviews and inspection mechanisms, which are not currently addressed in the course.

5. Limitations

The assessment of Practicum with P3M3 model undoubtedly presents an encouraging picture, however there are certain limitations associated with this study. These are described as follows:

- Due to the very small size of project teams, i.e., only up to a maximum of five team members, the extent to which prospective project managers could learn about team development, conflict resolution, communication, morale boosting activities, and other team management activities is limited.

- Since the projects are done in an academic setting, it is difficult for project managers to accurately perform cost estimation and management. Most of the project managers utilize dummy cost figures to perform cost estimation.
- The final year projects are either entirely application based, purely research oriented or R&D based. Due to this variation, project managers cannot consistently apply project size estimation techniques such as function or feature point analyses. Therefore, activity based estimation is usually performed for research and R&D based projects.
- The project advisors are faculty members who assume the role of project clients in this course. Since they are the owners and in some cases initiators of the final year projects, their decisions about the project's scope are usually considered final. Because of their academic authority the project managers face problems in efficiently managing project scope and requirements changes.
- Each year, about two dozen final projects are initiated, but the number of graduate students enrolled in practicum course is very small, i.e., about a dozen or less. Therefore, some of the final projects are left without a project manager, which deprives them of the practical experience that they could gain by working in a simulated real project environment.
- P3M3 is basically for organizations that want to excel in their program, portfolio and project management capability. Academicians around this world might not agree upon assessing an academic course with such a professional model, but we have selected this model as a means to evaluating the program's maturity with what is acceptable as a standard in the industry.

Despite all these limitations, we believe that a course such as Practicum in Software Project Management can certainly help academicians in providing practical software project management education to graduate student; enabling undergraduate students to learn to work in team under the supervision of project managers; and supplying trained project managers to the software industry.

5. Future Directions

We are working on improving this course by gathering feedback from graduated Masters and Bachelors level students who had been part of this course during their studies at the university. For this purpose, we intend to conduct a survey targeting MS students, BS students, faculty advisors, industry personnel who could give feedback on the students who had been part of the Practicum course. We have developed and circulated the questionnaires to our target addresses, and hope to receive and compile the results within a month.

6. Conclusion

In this paper, we have talked about the need for providing formal software project management education to graduate students intending to assume the roles of project managers in the software industry. We have highlighted the scarcity of academic programs and courses focusing on practical and effective project management education and have described the infrastructure of Practicum in Software Project Management, a graduate course which is part of the MS in Software Project Management program at National University of Computer and Emerging Sciences, Lahore Pakistan. We have also presented the basic project management activities that renowned software engineering practitioners deem significant for effective management of software projects. In order to evaluate the effectiveness of our course, we have assessed it against PM Solution's Portfolio, Program and Project Management Model (P3M3) and have found out that the course is 90% capable at P3M3's Repeatable level and 81% capable at Defined level. We believe that with some necessary changes in areas like configuration management, we can further improve this course.

Acknowledgements

We thank the academic office and the Practicum course instructors for providing us details of the course.

References

[1] Harry L. Reif, Michel Mitri, "How university professors teach project management for information systems; determining where project management education fits into the overall curriculum" Communications of the ACM, Aug 2005, Vol 48., No. 8., pp 134-136.

[2] June M. Verner, William M. Evans, "In-house software development: what project management practices lead to success?" IEEE Software, Jan 2005, pp-86-93.

[3] Dennis J. Hood, Cynthia S. Hood, "Teaching software project management using simulations", In proceedings of ACM's Eleventh Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), June 2006, pp-289-293.

[4] Joo Tan, John Phillips, "Real world project management in the academic environment", Journal of Computing Sciences in Colleges, Volume 20 , Issue 5, May 2005, pp 200 – 213.

[5] Dale Callahan and Bob Pedigo, "Educating Experienced IT Professionals by Addressing Industry's Needs", IEEE Software, Volume 19, Issue 5, Sep 2002, pp 57 – 62.

[6] Cusing, K., "Why projects fail", Computer Weekly, Nov 21, 2002.

[7] IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, (URL: www.acm.org/education/is2002.pdf) .

[8] IS '97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, (URL: www.is-97.org/rev/Review1.asp)

[9] Capers Jones, "Patterns of Large Software Systems: Failure and Success", IEEE Computer, Vol. 28, No.3 March 1995, pp 86–87.

[10] Robert B. Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice Hall, NJ, 1992.

[11] R. Glass, "Project Retrospectives, and Why They Never Happen," IEEE Software, Vol. 19, No. 5, Sep 2002, pp. 112, 111.

[12] Project Management Institute, (URL: www.pmi.org)

[13] Project Management Institute, "A Guide to the Project Management Body Of Knowledge (PMBOK)", Third Edition, 2004.

[14] PM Solutions, Project Management Maturity Model (PMMM), 2002.

[15] Project Management Institute's Accredited Universities list, URL: (http://www.pmi.org/prod/groups/public/documents/info/pdc_gac_accredited.asp)

[16] MS in Software Management, Carnegie Mellon University West Coast Campus, URL: (http://west.cmu.edu/prospective_students/software_management)

[17] PM Solutions, Portfolio, Program and Project Management Maturity Model, Feb 2006. (URL: http://www.pmsolutions.com/maturitymodel/opm3.htm)

[18] Capability Maturity Model Integrated (CMMI), Ver 1.2, (URL: http://www.sei.cmu.edu/cmmi/)

[19] National University of Computer & Emerging Sciences, Lahore, Pakistan, "Masters in Software Project Management course outline" URL: (http://www.nu.edu.pk/program-course.aspx?prg=msspm) .

[20] Software Engineering Management, SWEBOK: A Guide to the Software Engineering Body of Knowledge, 2004 IEEE Computer Society. (URL: http://www.swebok.org/)

[21] Ita Richardson, Allen E. Milewski, Neel Mullick, Patrick Keil, "Distributed Development: An Education Perspective on The Global Studio Project", Proceeding of the 28th International Conference on Software Engineering, May 20-28, 2006, Shanghai, China.

[22] Ita Richardson, Christiane Gresse von Wangenheim, "Why are Small Organizations Different?" IEEE Software, Jan 2007.

[23] Richard E. Fairley, Mary Jane Willshire, "Why the Vasa Sank: 10 Problems and Some Antidotes for Software Projects", IEEE Software Volume 20 , Issue 2, January 2003.

[24] Arthur B Pyster, Richard H. Thayer, "Software Engineering Project Management 20 Years Later", IEEE Software, Sep 2005, Vol. 22, No. 5, pp. 18-21.

[25] Mohammad A. Rob, "Project Failures in Small Companies" IEEE Software Nov 2003, Vol 20, No. 6, pp. 94-95.

[26] Neville Holmes, "The data doughnut and the software hole", IEEE Computer, June 2006, Vol. 39, No. 6.

[27] Robert L. Glass, Software Runaways: Lessons Learned from Massive Software Project Failures. 1998 Prentice Hall PTR.

[28] Sana Gul, Kashif Shahzad, "Pilot Testing Software Engineering Management to University Academic Projects: Prototyping Processes Development and Implementation for Better Managing Academic Projects in Practicum", NUCES, Lahore, Dec 2006