Prakalp प्रकल्प

May, 2006
Volume 8, Issue 1

Inside This Issue

"Have You Renewed Your Membership?"

Now circulating 1,500 copies to corporates in EPC, IT Sectors. For advertising see Pg. 8

Your key to successful projects

From the Editor's Quill

We are happy to bring out this Twenty-third issue of the PRAKALP.

2006 is proving to be a "happening year" for PMI in Asia Pacific region, with special emphasis on India.

The Chapter conducted elections on 16th Dec 2005 and elected a "new look" PMI Chapter Committee. Activity committee members have also been nominated presently to assist the Chapter committee members in their activities and prepare grounds for the future leadership of PMI-MC. The year 2006 holds a lot of promises and we hope that the new Chapter Committee will accept this challenge and make it a success.

One of the main events the PMI Mumbai Chapter organizes regularly is the PMP Club. PMP Club meets on regular basis every month. The areas of discussions are expanding and so is the attendance. We wish them all the best and encourage our other members to participate actively.

Inspite of all the progress, Software is still too buggy, takes too long to develop, and costs too much. An analysis of rework trends shows that the primary cause is the Inadequacy of rigorous unit testing and coding standards- compliance checks in projects. Many defects and non compliances are caught post unit testing and lead to 10-100 times more work to fix them in later phases of the project life cycle. This not only causes cost and schedule overruns but also generates poor client perception. Dealing with this aspect, this current edition of Prakalp is carrying a white paper on "Automating Code Quality Management" written by Ms Ruta Thakkar, a practicing Project Manager from M/s Hexaware Technologies, Mumbai.

"THE Art of War" written by Sun Tzu, is the oldest known book on military. In our present edition of Prakalp, we have presented an article on "What this book teaches us on effective traits to look in people while selecting members of management teams and assigning them to specific areas to carry out their assigned tasks".

We have also published a calendar of events for the year 2006, to help you plan your activities.

One last word, it is our sincere request that you please share your experience by submitting articles, news items, views etc to make the future PRAKALP issues, more readable and effective

R. Balaji -Editor.

editor@pmi-mumbai.org

REQUEST YOUR PARTICIPATION
- Technical Articles or other material related to Project Management for future issues of PRAKALP.
- Seminar Sponsorship from your company.
- Advertisement in PRAKALP.
- Your efforts in organizing chapter activities.
- Become a member of PMI Mumbai Chapter

PROJECT MANAGEMENT INSTITUTE (MUMBAI CHAPTER) IS PLEASED TO ANNOUNCE THE DATES FOR NEXT PMP CERTIFICATION EXAMINATION PREPARATORY COURSE BASED ON PMBOK GUIDE (THIRD EDITION)

THE NEXT COURSE IS SHEDULED ON 17, 18, 24 AND 25 JUNE, 2006
BASED ON PMBOK GUIDE (THIRD EDITION)

The venue would be Western India Instrumentation Centre, University Campus at Kalina, Santacruz (East),
The Faculty comprises of experienced Project Managers and certified PMP's (Project Management Professionals)

PMBOK sessions

Introduction

The Project Management Context

Project Management Processes

Project Integration Management

Project Scope Management

Project Time Management

Project Professional Responsibility

Project Cost Management

Project Quality Management

Project HR Management

Project Communication Management

Project Risk Management

Project Procurement Management

"HOSTEL FACILITIES AVAILABLE AT MODERATE RATES ON PRIOR REQUEST"

PMI - Mumbai Chapter welcome you to participate in the PMBOK Guide & PMP Certification - Examination Preparatory Course June 2006

COURSE CONTENTS

Introduction to the PMBOK
Project Management
Project Management Processes
Professional Responsibilities

Scope Management
- Initiation
- Scope Planning
- Scope Definition
- Scope Verification
- Scope Change Control

Time Management
- Activity Definition
- Activity Sequencing
- Activity Duration Estimating
- Schedule Development
- Schedule Control

Cost Management
- Resource Planning
- Cost Estimating
- Cost Budgeting
- Cost Control

Quality Management
- Quality Planning
- Quality Assurance
- Quality Control

Risk Management
- Risk Identification
- Risk Quantification
- Risk Response Development
- Risk Response Control

Human Resources Management
- Organizational Planning
- Staff Acquisition
- Team Development

Communications Management
- Communications Planning
- Information Distribution
- Performance Reporting
- Administrative Closure

Procurement Management
- Procurement Planning
- Solicitation Planning
- Solicitation
- Source Selection
- Contract Administration
- Contract Close Out

Integration Management
- Project Plan Development
- Project Plan Execution
- Overall Change Control

Question Answer Sessions in Examination Pattern on each Knowledge Area.
Case Study and Discussions.
Course would satisfied 35hrs. Training Requirement prescribed as a pre-requisite for the PMP Certification Examination"

Time : MORNING
9:00 to 11:15; 11.30 to 13:15
AFTERNOON
14:00 to 15:45; 16:00 to 19:00

Chapter Committee Members

Sr. No.	Name	Designation
1	D. Y. Pathak	President
2	Deepak C. Pawar	V.P.- Membership & Recruiting
3	Tejas V. Sura	V.P.- Treasury & Finance
4	Rakesh Gupta	V.P.- Communications
5	Bharat Bhagat	V.P.- Education & Certification
6	R. C. Goyal	V.P. - Programs
7	Aneesh Kadyan	V.P.- Special Projects
8	R. Balaji	V.P.- Publicity
9	Vivek Prakash	V. P. - PMP Club

Activity Committee Members

Sr. No.	Name of the Member	Activity
1	Kishore Nayak	Publications
2	Dileep Garg	Education and Certification
3	Gururaj	Education and Certification
4	Kishan Sharma	Education and Certification
5	Asif Sayyad	Education and Certification
6	Kalyan Narayanswamy	PMP Club / Finance and Legal
7	Sujata Lovalekar	PMP Club

Staff

Sr. No.	Name	Designation
1	Shivani Nitin Shinde	Chapter Secretary

Automating Code Quality Management

By

Ruta Thakkar - *Hexaware Technologies*

Introduction

In today's competitive world of delivering defect free, adaptable software products and services, the race is on amongst companies to use initiatives like SEI CMMI and Six Sigma to ensure projects/services are delivered within budget, on time and with superior quality.

There has been a growing maturity in the industry over the past 30 yrs in requirements management, visual modeling, project life cycle management (Iterative/Agile Methodology), Configuration Management and automated functional and load testing.

In spite of all of this progress, the fact remains - software is still too buggy, takes too long to develop, and costs too much.

A 2003 study conducted by the U.S. National Institute of Standards and Technology (NIST) reported a $60 billion annual cost from buggy software in the U.S. alone.

A 2001 paper from CeBASE stated the following:

Current software projects spend 40-50% of their efforts in avoidable rework.

An analysis of this trend shows that the primary cause is the "Inadequacy of rigorous unit testing and coding standards compliance checks in projects".

Many defects and non compliances are caught post unit testing and lead to 10-100 times more work to fix them in later phases of the project life cycle. This not only causes cost and schedule overruns but also generates poor client perception.

The purpose of this paper is to highlight the current issues / limitations of our approach to Code Quality Management and methodologies to alleviate them.

Current Practices and Issues

Most companies have no structured enforcement for carrying out rigorous unit testing. In recent years open source tools like Junit, HTTPUnit, MockObjects and NUnit have become popular and provide a basis for companies to start their unit testing efforts. Unfortunately the manual effort in creating these unit test cases is overwhelming and teams are unable to leverage the benefit due to stringent project timelines, oversight, lack of automation and ineffective reviews. Also, unit test cases are almost absent in maintenance projects where all the team has to contend with, is the code base and minimal documentation.

Additionally, it is rarely possible for a developer to write test cases that cover every branch/outcome and expected and unexpected exceptions. Even, if he does manage to write these test cases, maintaining these test cases as code evolves becomes a time-consuming Herculean task.

Projects thus have very limited unit testing and defects that should be

Early Detection / Fixing Saves Costs

Software Development Life Cycle

Cost to correct a defect

Rules of Thumb :

✓ Reworking defects consumes 40-50% of the total cost of software development

✓ Every hour spent in defect prevention saves three to ten hours of repair time

(Boume Stewe McConnell, Rapid Development)

✓ For an application in production it costs 50 to 200 items more to fix a defect than to fix it early in the eyele

Prevent find and fix bugs here!

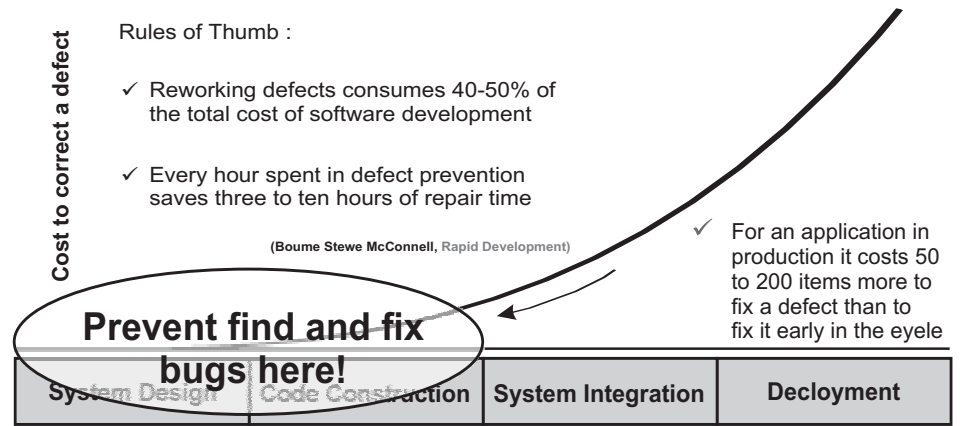| System Design | Code Construction | System Integration | Decloyment |

Figure 1 (See References)

identified during the coding phase are generally detected only during Integration/System Testing or User Acceptance Testing. Listed below is a typical pie chart for defect analysis of development projects in new technologies.
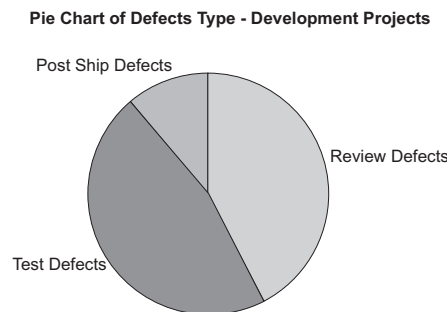
Pie Chart of Defects Type - Development Projects

Post Ship Defects

Review Defects

Test Defects

Figure 1

The chart depicts that the System Test Defects and Post shipment defects comprise more than 50-60% of the total defects detected in a software product.

This increases the value of the Defect Age Factor and undermines the quality that we deliver to clients

Compounding the issue is the lack of compliance of code to coding standards, guidelines and best practices identified for a particular

language, framework or design. Projects do have guidelines and checklists, however failure of developers to include these goes undetected due to ineffective and partial reviews

If an organization generates a Pareto chart for causal analysis of defects, it will be able to identify those causes that generate 80% of defects. Generally these causes belong to one or more of the following:

1) Error in Coding

2) Non-adherence to standards, guidelines & procedures

3) Inadequate communication or documentation

4) Inadequate standards, guidelines and procedures

We have also found that more than 80-90% of defects are attributed the coding/construction phase of projects in our industry.

All these metrics iterate the need of better unit testing and code compliance checks. However, traditional methodologies of checklists,

peer reviews. Manual generation of unit test cases etc have been ineffective in helping us achieve significant improvement.

The Solution

The area of "Unit Testing" has been one of the least explored areas of the project lifecycle phases. However, companies are increasingly realizing the impacts of ignoring this piece. The nature of unit testing dictates that it cannot be automated 100%. However, there are several areas that can be simplified and made effective by use of automation tools.

We studied the various aspects that would help organizations and teams reap the true benefits of unit testing and the findings are listed below.

We have categorized these findings relative to project team roles starting from the Developer and ending with the Project Manager.

The Developer's Wish List:

1   An automated check that highlights where run time exceptions can be thrown, so that I can do better error handling

2   An automated check that identifies performance related issues and memory leaks e.g. not releasing resources appropriately

3   An automated check that highlights violation of coding standards and helps automate the correction of the same. The automation should ensure that changes are made in all files e.g. a naming change of a method should automate the method name change in all files that call that method

4   An automatic notification when I check my code in a version control system and it breaks the compilation and build process. This indicates that I need to interact with team members and ensure that all calling code correctly handles my latest changes

5   Auto generation of unit test cases so that I have to just specify what assertions are to be made on various inputs. The test cases should cover all branches and boundary conditions. I should be able to extend these test cases if needed.

6   During maintenance or production support, the tool should be able to generate new unit test cases when I carry out incremental code updates.

7   All automation should be available in the popular IDEs (Integrated Development Environments) that I use for my development

8   It would be nice to have tools that automatically generate unit test cases by monitoring the run of the application especially for maintenance projects

The Technical Lead's Wish List:

1   I should be able to leverage the knowledge of my company process library and build on the experiences of prior projects while configuring coding best practices. The automation tool should not restrict to the default rules provided in the tool.

2   I should be able to control the automation checks on all developer terminals by having a centralized configuration facility

3   In addition to defining technology/language related rule checks, I should be able to define checks that are specific to my company, application framework, project or client. I should be able to export my rules configuration to projects that have similar needs

4   I should be able to continuously refine the rule repository as we gather learning in the project lifecycle and the new rules should be run on all existing code.

5   I should be able to automate the execution of unit test cases and coding standards compliance check through standard build utilities like Ant/NAnt.

6   I should be able to have clear reporting on which pieces of code have maximum defects, category of defects and developers responsible for those pieces

7   I should be able to collect all unit test cases in a regression test suite for unit testing and be able to invoke it for every intermediate release.

8   I should be able to run unit test cases that relate to selective rules. For e.g., during performance testing, I would like to run only those cases that address rules for checking memory leaks whereas during security testing, I will be more interested in running rules checks that test SQL Injection.

The Project Manager's Wish List:

1   I should be able to find out at any point of time the number of passed/failed unit test cases executed on the latest version of code in my version repository

2   I should be able to identify critical pieces of code that are lacking in successful unit testing and the

developers responsible for those pieces so that I can arrange training for them or reassign development work as appropriate

3 The tool should have a dashboard reporting facility where I can retrieve all quality related metrics to be sent to company's QMG group.

4 I should be able to measure the team's progress with CUT against set goals.

5 I should have a ready set of unit test cases and test reports that I can deliver to my client. The client is increasingly demanding this deliverable for early quality assurance.

The Tools

Any proposed solution cannot be effective until one finds the right tools and the right methodology of applying these tools to resolve the problem at hand.

Having collected the stakeholder requirements for unit testing, we explored the market for availability of commercial as well as open source tools that would aid in satisfying them.

We found that a few commercial testing tools have evolved to provide auto generation of unit test cases, auto generation of test data, facility to define custom coding standards & application specific rule checks and support for quick generation of stub objects. From the management perspective, the tools provide good dashboards to track the team's progress in unit testing as well as their performance against quality goals set for the project.

Some of the tools we explored and found promising in Java and Microsoft Technologies are listed here:

1. Parasoft's Jtest for Java/J2EE projects:

    Product Datasheet:
    http://www.parasoft.com/jsp/products/quick_facts.jsp?product=Jtest&itemId=15

2. Parasoft's .TEST for .Net projects:

    Product Datasheet:

    http://www.parasoft.com/jsp/products/quick_facts.jsp?product=TestNet&itemId=136

3. Agitar's Agitator Software and Management Dashboard for Java/J2EE projects:

    Product Datasheet:

    http://www.agitar.com/downloads/AgitatorDatasheet_3_0.pdf

    We have also found some great open source tools which are especially useful in smaller companies. These help in identifying coding standards compliance failures as well as possible bugs & unhandled exceptions in code. They integrate well with most IDEs and are a boon when companies don't have budgets to support fancy tools.

1. FindBugs: An open source tool that looks for bugs in java programs based on well researched bug patterns.

    Product Information:

    http://findbugs.sourceforge.net/

2. PMD: An open source tool that looks for dead code, duplicate code, possible bugs and suboptimal code in Java programs

    Product Information:

    http://pmd.sourceforge.net/

3. Checkstyle: An open source tool that checks for coding standards compliance using Sun's java coding standards or your own custom coding standards.

    Product Information:

    http://checkstyle.sourceforge.net/

Conclusion

Automated unit testing and code compliance checks may not be the "magic pill" to solve every quality issue in your project, but the benefits it promises makes it a deserving candidate for adoption.

❑ Code reviews no longer need to be on a sampling basis. Manual errors in reviews can be completely eliminated. Coding standards and best practices can be enforced effortlessly. Coding standards once established for a client, application or framework can be reused consistently. Review effectiveness of the organization can be significantly improved.

❑ Project managers can provide unit test cases and test results to clients with ease and foster confidence earlier in the project life cycle. This brings more transparency to the outsourcing model.

❑ Unit test cases can be kept in sync with code during maintenance or project support periods with minimal efforts.

❑ Automation tools provide ideal support for effective implementation of agile project management methodologies. Agile methodologies allow for evolving requirements and frequent builds to client to elicit early client feedback and hence pose great challenges to development teams. Project teams cannot follow them successfully without the aid of such automation.

The foremost and the most important benefit is the "means" these tools provide to transfer the responsibility of code quality to the developer, make them more accountable and foster an environment where the developer takes initiative to deliver high quality code.

References

Documentation and Technical papers available at the following links:

http://www.infoworld.com/article/04/08/27/35TCagitator_1.html

http://www.javaworld.com/channel_content/jw-testing-index.shtml

- *www.cebase.org*
- *www.junit.org*
- *www.agitar.com*

  (Figure 1 has been reproduced from an Agitar white paper)

- *www.parasoft.com*

The Art of War and Effective Team Building

THE Art of War, written by Sun Tzu, is not only the oldest known book on military tactics, but also one of the most comprehensive, and popular, volumes ever written on the subject.

Generals are the guardians of the state. If they provide effective protection, the state will be strong; if they are defective, the state will be weakened.

These are the five qualities that are dangerous in the character of a general:

1. A reckless leader may be killed.

2. A cowardly commander may be captured.

3. If he is quick-tempered, or rash, he can be made a fool of.

4. If his sense of honor is too brittle, he can be dishonored by lies and rumors.

5. If he is too compassionate, he can be easily harassed.

There are three ways in which an army can bring disaster on its nation; when the general orders an army to advance when it should retreat, or to retreat when it should advance; when men ignorant of military tactics are allowed to take part in their planning; when generals who do not understand command assume authority.

Any of these traits in a commander is a serious fault, which can lead him to death and bring his army to calamity.

The above two extracts tell us what to look out for when selecting members of management and negotiating teams. Select your teams carefully and be certain that every member of your team is up to the job assigned to them. Just because a person is good at one particular job does not necessarily mean they will be equally good at another, so base your decision on the specific job they are to fill. Perfectly competent people assigned to the wrong job can spell disaster for the project, for themselves and for you.

Generals who are capable and whose decisions are not interfered with by the king will be victorious.

When you have selected your management teams and assigned

them to specific areas, let them carry out their assigned tasks in peace. If you cannot trust them to do their job, you were wrong to place them in that particular position. Either reassign them to a place where you, and they, feel comfortable or get rid of them.

When a commander goes into battle without seeking personal fame and withdraws without orders, to protect his men, but without concern for his own punishment, he is a precious jewel of the state. Such a man regards his men as his own sons and they will die with him.

Even the best managers occasionally disobey orders, the question here is `why`. If they habitually overstep their authority, or are just trying to make themselves look important, they should not be in authority. If, on the other hand, their ultimate concern is the good of the people in their charge, they are the rarest breed of person, a caring human being.

Visit us at:
www.pmi-mumbai.org

Calender of Activities during the Year 2006

Mark Your Diary	Prakalp		PMP CLUB PM FORUM		PMBOK / PMP Review Course		Chapter Committee Meeting	
Jan.	-		8	✓	14,15,21,22		23	✓
Feb.	-		5	✓	25,26	✓	-	
Mar.	-		5	✓	4,5		-	
April	-		2	✓	15,16,22,23	✓	-	
May	15	✓	7	✓	-		-	
June	-		4	✓	17,18,24,25		11	✓
July	10		2		-		15	
Aug.	-		6		19,20,26,27		-	
Sept.	30		3		-		23	
Oct.	-		8		21,22,28,29		-	
Nov.	-		5		-		-	
Dec.	15		3		16,17,23,24		21	

NOTE : AGM & Elections to the PMI MC Managing Committee to be held on Friday, December 21st, 2006 evening

BOOK POST

ADDRESS CORRECTION REQUESTED

POSTAGE