Member's Dialog

Inside this Issue

Achyut Prasad, Phatak Baug, Off
Karve Road, Near Tech Mahindra,
Pune, Maharashtra, PIN - 411016,
India; Telephone: +91-9922909060;
http://www.pmipunechapter.org

Editor's Corner

*Good things happen when you join
PMI-Pune Chapter!*

June has been traditionally an exciting month for Pune and surrounding region. With the onset of monsoons, the general weather improves considerably. With the change in weather, we have also been witnessing a somewhat radical change in our usual monthly seminar sessions. Our sessions are becoming more interactive in nature and your feedback has been very encouraging and we are keen to make improve it further.

One of the many new trends worldwide gaining currency is LEAN & AGILE. The concepts of Lean and Agile are based on best practices in product development or project management and have been developing for decades. The goal is to maximize efficiency – to increase or maintain perceived customer value with less work. PMI Pune Chapter has also been emphasizing on AGILE and SCRUM. So while Agile is a general philosophy regarding software production, Scrum is an implementation of that philosophy pertaining specifically to project management.

According to ESI, When compared to traditional methods, studies show that Agile methods can reduce costs, speed time to market, and improve quality; however, in 2013, many organizations will continue to fall short in realizing the promise of Agile. Why? Because the professionals assigned to Agile projects aren't trained in Agile methods and their organizations are not culturally ready to embrace its principles. It's not sufficient to train just a handful of Scrum masters. The Scrum team, including developers, testers, and product owners, needs to know how to apply Agile practices. In particular, the organization's executives need to understand how they can help break down the cultural barriers to adoption, which is crucial. Providing training to only those who lead these efforts will undermine overall Agile adoption, resulting in poor or failed implementations.

But then, that's what PMI's Pune Chapter has been trying to do all through the years, and that's why we think you're going to like the new emhpasis on AGILE and SCRUM in the coming weeks. Send us your thoughts and ideas, and then keep checking back with us.

Rinoo Rajesh is the Vice President, Marketing & Communications at PMI's Pune Chapter and can be reached on marketing@pmipunechapter.org

PMI Article

*Article shared by **PMI Memphsis Chapter***

Sources of Conflict Between Developers and Testers in Software Development

This article is based on a research paper by the same name, presented at AMCIS 2008 and available at: http://works.bepress.com/xihui_zhang/6. Refer to the original paper for supportive research citations. The paper is coauthored by:

Xihui Zhang, *University of Memphis*
Jasbir S. Dhaliwal, *University of Memphis*
Mark L. Gillenson, *University of Memphis*
Gertrude Moeller, *University of Memphis*

This following is not a condensation, but an adaptation by one of the coauthors to provide research-supported guidance for IT project managers. Guidance is also relevant to non-IT projects.

OVERVIEW

Business-related information technology (IT) is becoming more integrated and complex, requiring higher levels of collaboration between specialists with different responsibilities, backgrounds, and skills.

IT development projects involve complex social interactions between diverse project team roles. One of the most important role interactions is between developers (including systems analysts, designers, and programmers) and software / systems testers. Research shows that this interaction frequently involves a level of interpersonal conflict that damages software and system project outcomes.

Therefore, it is critical that IT practitioners understand the sources of developer-tester conflict so that appropriate measures can be taken to mitigate its negative impact, which endangers organizational effectiveness and efficiency.

Thus far, relatively little research has focused specifically on conflict between developers and testers. This study addresses the research question: What are the sources of interpersonal conflict between developers and testers in software development?

Results indicate that conflict sources between software developers and testers fall into three major categories: process, people, and communication.

IT PROJECT PROCESS AND ROLES

Software testing is indispensable. It is the only way to ensure levels of quality needed to maintain or improve competitive market position across a broad range of industries and businesses.

Process and Role Changes. The role of testing in software development is currently undergoing a reassessment for two reasons:

(1) Traditional systems development life cycle (SDLC) approaches have proven inadequate to deliver required process and product quality for large, complex IT development projects.

(2) More effective alternate methodologies (such as agile methods or prototyping) require significant changes to project processes and team roles, including the process of software testing and the role of software tester.

Tester Integration. Research has shown that engaging testers early and maintaining their involvement throughout the software development process significantly improves overall project performance. This may seem counter-intuitive, since testing starts only after code development is complete. However, 60-75% of tester effort begins before a single line of code is written. Fully engaging testers early in the project, with open access to information and influence, drives the planning and preparatory excellence needed to deliver a quality product.

Growing awareness of these facts has led to more frequent and sophisticated interaction between developers and testers. This in turn, inevitably increases the potential for interpersonal conflict between these two very different IT project team roles.

CONFLICT FACTORS

Interpersonal conflict results when interdependent parties have different work goals, backgrounds, experience, mindsets, and values. Research has identified factors that encourage conflict between IT developers and testers:

(1) **Responsibilities.** A tester's primary responsibility is to identify and report problems in the code written by the developer. Thus, the roles of software developer and software tester are inherently adversarial.

(2) **Work methods.** To perform effectively, developers and testers must think very differently about the work to be done. Developers "build" code, and seek to maximize "efficiency" by completing code as quickly as possible. Testers "break" code to ensure quality and maximize product "effectiveness." Testers focus on compliance with user requirements, while developers look for ways to exploit technical possibilities, sometimes violating user specifications in the process.

(3) **Role valuation.** The two roles are differently valued by teammates, management, and other stakeholders. Typically, developers are perceived, and perceive themselves, as having more importance to the project and higher status on the project team compared to testers. Testers often feel they must work continually to gain a level of respect similar to that granted to developers.

(4) **Time pressure.** Because critical activities for both roles occur toward the end of the IT project life cycle and are dependent on upstream deliverables, both developers and testers are under greater time pressure than other team members. Thus, the project schedule itself contributes to the adversarial nature of the developer-tester relationship. Test planning is dependent on the quality and timeliness of developer deliverables; late delivery or incompleteness compromises test plan quality and tester preparation. A key testing output (code defects) pressures developers to fix defects with minimal turnaround time; the fixes may then be found wanting by the tester necessitating rework.

(5) **Interdependence.** Developer and tester must work together while simultaneously competing for scarce resources including project personnel and time. Higher levels of interaction, under competitive, task-related pressure, increase the potential for interpersonal conflict.

CONFLICT IMPACT

For the reasons described above, conflict between developers and testers is inevitable and pervasive, and impacts both their working relationship and the quality of the software they produce. Poor quality software and systems lead to disgruntled end users.

Developer-tester conflict is also associated with lower job satisfaction, which research has shown to be associated with absenteeism, intention to leave, and actual workforce turnover.

Increased understanding of the sources of developer-tester conflict is the key to developing strategies and tactics for managing it. Measures taken to enable positive, effective, and mutually satisfactory working relationships will improve overall software development effectiveness and efficiency, thus supporting product quality.

STUDY METHOD

Critical incident technique (CIT) was used to answer the research question:

> *What are the sources of interpersonal conflict between developers and testers in software development?*

CIT is an appropriate method to gain understanding of the sources of developer-tester conflict because it focuses on the perspective of the individuals involved in problematic interactions, based on their reported thoughts, emotions, and behaviors.

At a high level, the research approach consisted of gathering written conflict scenarios from developers, testers, and IT managers, and then analyzing the data using a five-step qualitative evaluation approach.

The study was performed at a large, globally branded organization employing over 4,000 software developers and 700 testers. Written conflict scenarios were collected from participants in a custom two-day "Conflict and Conflict Management" training module. All course participants were actively engaged in software development, software testing, or both.

Participants were asked to provide a written description of a software development conflict experience, including information about the issues involved and reactions of both parties. Fifty of these developer-tester conflict scenarios were collected, 19 from developers, and 31 from testers.

SCENARIO ANALYSIS

A five-step process was used to analyze and categorize the scenarios by conflict source type:

(1) The handwritten scenarios were transcribed into MS Word and the transcriptions were checked for accuracy.

(2) Each of the 50 transcribed scenarios was independently analyzed, summarized, and categorized by two researchers: a) An IT doctoral candidate, and b) A project manager with a social science doctorate. Category agreement between the two was 83.3%.

(3) Each scenario was reviewed, validated, and summarized by a senior University of Memphis MIS faculty member actively involved in software testing instruction and research. Final summary versions of the scenarios were synthesized from the three summary sets, and each summary was printed on a 4x6 index card.

The summaries were further categorized using a qualitative sorting methodology (Q-Sort), described below:

(4) Q-Sort 1: The index cards were arranged in random order, and three coders (an IT manager and two IT professionals from the employing company) independently read the conflict scenario summaries printed on each card. The coders then sorted the cards into piles based on their perception of similarities in conflict source for the described interactions.

Finally, each coder supplied a descriptive name for the conflict categories represented by each pile of cards. The researchers developed category and subcategory lists based on coder output.

(5) Q-Sort 2: In a second set of coding sessions, the cards were presented in random order to three additional coders (two IT managers and one IT professional) who were provided with the category and subcategory lists developed from the first Q-Sort. Each coder placed the summary cards into a conflict category from the list, and indicated their perception of the best subcategory fit.

As a result of the steps described above, a total of 20 scenarios were removed from the study for one of two reasons:

- **Steps 1-3:** All three researchers agreed that the scenarios did not describe developer-tester conflict.

- **Steps 4-5:** All three analysts agreed that coders had not reached consensus on conflict source categorization. (For each Q-Sort, consensus required agreement by at least 2 of 3 coders.)

Based on coder classification, the thirty remaining scenarios were assigned to one of four categories:

- **Communication:** Interpersonal communication, including knowledge sharing.

- **People:** Individual aspects including knowledge, experience, emotions, attitudes, and personalities.

- **Process:** Role boundaries, task and process ownership, process documentation, and process compliance or noncompliance.

- **Organization:** Organizational structures and politics.

ONLINE SURVEY

To further validate the categories resulting from the analytical process, a quantitative approach was taken. The thirty scenarios were incorporated into an online survey, and the survey link was emailed to a list of 300 IT scholars, including university faculty and doctoral students. One hundred surveys were completed (participation rate = 33.3%). For each scenario, respondents selected the conflict source category that in their opinion, supplied the best match.

To ensure analytical quality, exclusion rules were applied to the resulting data. Of 24 validated scenarios, 10 (42%) were contributed by developers and 14 (58%) by testers. Three categories or layers of conflict were validated: Process, People, and Communication. These were subdivided for a total of five categories, described below.

Process – Documentation (3 of 24, 13%). In two of the three scenarios, testers report that developers failed to communicate information critical to testing activities (defect fixes, development changes). In the third scenario, a developer identifies inadequate requirements as a conflict source, leading to tester requests for changes after code completion.

Process – Compliance or noncompliance (2 of 24, 8%). Of the two scenarios, one is a tester report of developer failure to follow a mandated defect management process. The other is a developer report of tester criticism of the code design, based on developers' failure to comply with a coding standard.

People – Emotions / Attitude / Personality (8 of 24, 33%). Some scenarios include strongly negative interactions. They can be subdivided by content type:

- **Behavioral:** Negative behavior is described.

- **Perceptual:** Negative attributions regarding other's attitude or orientation are expressed.

- **Combination:** Elements of both.

Some scenarios explicitly report damage resulting from the interaction. For example, an emotional outburst by a developer results in abandonment of a prescribed level of testing. In another case, a developer refuses to supply information needed for test design, resulting in poor tester preparation.

Implicit damage can be inferred in other cases; for example a developer reports general reluctance to work with a tester who is perceived as gloating about critical code defects. The negative impact of this damaged working relationship is obvious.

People – Knowledge / Experience (2 of 24, 8%). Two developers report conflict resulting from testers' lack of knowledge, resulting in invalid defects and other problems.

Communication, Knowledge sharing (9 of 24, 38%). Included are reports of: (1) errors not detected because the test lead was left out of the code maintenance communication loop, (2) developers deliver non-test-ready hardware because testers failed to communicate expectations, (3) late or unclear developer communication of new "release to production" criteria result in disagreement about test results, (4) general lack of communication results in test delay and quality issues, and (5) developer fails to communicate code changes or code status, resulting in invalid test cases and test failures.

Process implications. Although only 5 of 24 validated scenarios were classified into the "Process" category, scenarios classified into the "People" or "Communication" categories by Q-Sort coders and survey respondents include several that strongly imply process gaps or noncompliance.

For example, both of the "People – Knowledge / Experience" scenarios involve perceptions of poor tester knowledge. The employing organization mandates that developers provide documentation that serves as input for multiple downstream processes, including test script creation and execution. A lack of tester knowledge raises the question of whether these documents were absent, inadequate, delivered to the testers behind schedule, or not used for test planning for some other reason. Process noncompliance by developers, testers, or both, may be involved.

Process issues are also implied in the nine scenarios classified into the "Communication, Knowledge sharing" category. Active process noncompliance is implied in the case of a test lead left out of the code maintenance communication loop, since the employing organization mandates that the test lead be included in core project team communication. The remaining scenarios in this category imply process gaps that should be addressed by the project manager to ensure adequate team communication.
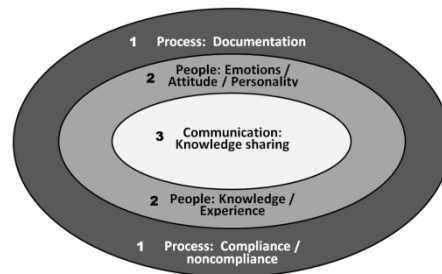
CONFLICT MODEL

A two-layer conflict model was developed based on the conflict categories discussed above. The basic conceptual structure (depicted in Figure 1) is a layered subset structure in which each layer constrains or influences the category or categories contained within it:

(1) "Process" provides an organizational context for software development and is therefore depicted as the outermost layer.

(2) "People" is contained within process because when enforced, process constrains and guides individual and group behavior.

(3) "Communication" is the innermost layer because it is a key component of human behavior and thus a function of, and contained within, the "People" layer.

This structure is intuitive and supported by practical experience. Working from the bottom up, project team communication (layer 3) is performed by people (layer 2), whose choices of when, how, and what to communicate are constrained and guided by established process (layer 1).

Reversing direction and starting with the top layer, process (layer 1) directly influences team and individual behavior (people, layer 2) and thus indirectly influences the nature, frequency, content, direction, and effectiveness of communication among the people performing the project work (layer 3).

Figure 1. Three-layer conflict model.

KEY FINDINGS

The purpose of this study was to identify sources of interpersonal conflict between software developers and testers. Results indicate that developers and testers experience conflict because:

(1) They don't document their work properly or follow standardized procedures correctly.

(2) They engage in, or are recipients of, strongly negative behaviors and consequently

develop negative perceptions of each other.

(3) They don't communicate appropriately, effectively, or efficiently.

Developer-tester conflict is rooted in both context (process or organizational characteristics) and ongoing human interaction (behavioral, attitudinal, or communication characteristics). From a practical standpoint, organizations need to take these categories into account in creating both long-term strategies and short-term tactics to manage (e.g., mediate, resolve, and prevent) interpersonal conflict between software developers and testers.

Implications for practice

Our study aligns well with advocacy for research that is clearly linked to practice, both in its methodological emphasis on practitioner-informed conflict antecedent categories and in the construction of an experientially-based conflict model.

Study findings have important implications for practice. Accurate identification of potential conflict sources enables IT project managers, development managers, technical leads, and test leads to proactively design project management strategies that support appropriate communication, open discussion, and mutual respect and trust, thus minimizing development of dysfunctional conflict, and providing for effective intervention before conflict can escalate and damage project outcomes, software quality, and essential working relationships.

Communication. Communication functions as a major source of conflict between developers and testers. However, as noted above, communication-related scenarios appear to describe symptoms of an underlying cause: lack of unified, appropriately designed and effectively applied project structure and process. Thus, through second

layer mediation ("People"), process gaps, flaws, or inconsistencies can negatively impact both frequency and quality of communication between team members.

Lack of effective and efficient communication between software developers and testers damages outcomes. Therefore, creation of an environment that facilitates effective, mutually respectful communication between developers and testers is critical. Well defined roles and responsibilities, supported by appropriate, consistently enforced project process decreases the opportunity for misunderstanding between these essential and highly interdependent IT project roles.

How can this be done? The solution is to define processes that open multiple communication channels between software developers and testers, and reward their use. When working relationships are first established, the emphasis is on face-to-face meetings and phone calls. To support effective collaboration, written media such as email, instant messaging, online networking, and descriptive documents are essential, but these should be regarded as supplements to, and not substitutes for, the interpersonal richness of direct, one-on-one and small group contact.

Process and Behavior. Within an organizational context, people typically act (or fail to act) in compliance with process, a key organizational feature. Rarely is formal process significantly altered if the people assigned to a project are replaced: the usual expectation is that process will remain relatively stable regardless of team composition. By its very nature, process constrains individual behavior so as to produce predictable results. Predictability enables control, and control is the entire purpose of creating, documenting, and enforcing process.

The change-inhibiting effects of prevailing IT social arrangements have been noted by other researchers. Improved administrative models and processes are needed to enable more flexible team interaction and negotiation. With reference to the antecedents of interpersonal conflict described above, this could include procedural innovations to address flaws including but not limited to:

- Poor process fit or inconsistent process compliance.

- Poorly defined team roles.

- Lack of unified cross-functional authority structures.

The findings of our study indicate a need for process tools as a means of improving developer-tester communication, by clearly delineated responsibility for reporting, status communication, decision documenting, and information distribution. Clarity and unity of vision is enabled by adopting standards and technology to achieve on-demand information access. Finally, training, mentoring, and formal policy can support more effective people management and negotiation behaviors.

Process Innovation. Despite its power to constrain individual behavior, individuals can and should function as owners of formal or informal processes that impact team interactions. When an established process does not adequately serve project team needs, or when incompatible processes interfere with team effectiveness, team members can formally or informally develop and implement process enhancements or modifications. Proactively creating and negotiating such procedural innovations, and enabling emergent modification as team needs evolve, is a key leadership function for managers, project managers, and individual developers and testers.

CONCLUSION

Interpersonal conflict between developers and testers is a pervasive phenomenon in the software development process. Although complete elimination of divergence of opinion or approach between developers and testers is neither possible nor desirable, overall software development effectiveness and efficiency can be greatly improved if emerging developer-tester conflict is managed at the source. To fulfill this goal, the first step is to clearly identify and thoroughly understand the source of conflict. Our results have provided the basis for and demonstrated the utility of such an approach.

104th MONTHLY SEMINAR!

PMI Pune Deccan-India Chapter organized its 104th Monthly Seminar on June 08th 2013. This seminar was attended for over 90 members / non-members. Seminar was conducted in three sessions.

Session 1-

'Shifting Business Dynamics to Process Consulting for customer success'

Speaker Profile-

Over 25 years of IT Industry experience – having played major roles in Consultancy, PMO, and Quality Management apart from mainstream Application development/maintenance Projects execution.

In last eight years have successfully led number of initiatives for major customers and was instrumental in design and implementation of home-grown tools.

As, Principal Quality Consultant, has responsibility to contribute to major quality initiatives organisation wide.

MR. Shirkant Chapekar

Topic Abstracts

Traditionally Quality Assurance team in an IT consultancy is responsible to ensure projects follow internal processes as per organization quality standards. QA was seldom responsible for implementing new practices based on changes in customer expectations. IT budget cuts for customers (industries) across the globe combined with business agility compels customers to opt for technical and economically viable solution. This in turn compels them to take decision on buying, renting cheaper, faster solutions. In order to cope up with this change IT consultancy needs to re-organise its internal functions especially delivery and QA. Hence there is need for QA team to assess team capability, upgrade competencies and play advisory role to 'delivery' function for process improvement through innovative approach, right mix of practices selected from various frameworks and radically change its contribution towards business excellence.

Mr. Shrikant did a great job in highlighting how QA function can take additional responsibility to take active role in process improvements to align business processes for optimized deliveries to achieve customer success.

Major focus areas for partner engagements would be on two aspects of first Transitioning the business processes and then Transforming business processes.

While customers are looking for value based partnerships in current business scenario, IT companies should focus on best practices from all industry proven standards like CMMi/Agile/ISO etc. He highlighted the capability development requirements for the teams to take on the emerging challenges.

Session 2 -

Felicitation to New PMP certified members

New PMPs from the month of May-2013 were felicitated by Rahul Sudame.

Session 3 -

Presentation by Luidia on EBeam

eBeam interactive technology turns static meetings into lively, visual collaboration environments where everyone can participate, whether they're in the room or not. Luidia produces portable interactive whiteboard solutions, such as the eBeam® Systems, that work with existing writing surfaces to capture meeting notes as they are created and, in conjunction with digital projection, can turn traditional whiteboards into interactive ones. Luidia creates portable products for the mobile professional, solutions for education and training, or embedded components for OEM customers.

For more info visit their website: http://www.e-beam.com/home.html

Session 4 -

Discover Invisible Power'

Speaker Profile-

Began on career making path as a helicopter pilot and then near fatal accident taught him values of life and recovering from accident gave him different perspective and meaning of ' Life '. As a matter of discovery of self , realised that training is his natural preference. Observing human psyche and the way a person responds to everything happening around , brings me a lot of experience and joy. Conducting out bound program perfectly meets my interest. He believes that all that we do actually emanates from our inner

self and so if we become reasonably aware of our inner self , then we can improve quality of our life. To help people become aware of inner self, He launched his firm titled ' Touch Inner self'. He also practices learning from Basic and Advance Labs in Human Process conducted by ISABS.

Specialties: Conducting Out Bound Programs and Carrying out training based on cognitive input to know inner self and see where are we with respect to role that we play; Customization of training to suit the needs of the organization; Professional and personal counseling; Focus on group unconscious competence

Cdr. Mukund Joglekar

Topic Abstracts

While managing the people involved in the project, we sense the impact of the unseen forces prevailing in the team. Some of these are enabling and some are disabling. There is no doubt that better project management will evolve from better team work and better leadership. However, each member while cognitively understands importance of these and might have put in efforts to develop self and team w r t soft factors, yet he may not know what to do and how to do whats relevant to him. The experiential method enables one to find answers to both questions - what to do and how to do. The simulations / activities conducted brings out truth underlying the visible behavior and that gives a picture of his inner self.

Role Play activities

Cdr. Mukund did a wonderful job in explaining why outbound training programs make sense and help in achieving better results out of training offered.
He also conducted a 20 minutes live exercise with attendees and showcased the effectiveness of the methods used.
He also declared a special offer of 80% discount to his signature one day workshop for first 20 registrations from PMI membership.

Sponsor for 104th monthly seminar:

'Luidia' represented by 'Faichi Solutions Pvt.

http://www.e-beam.com/home.html

CALL FOR ARTICLES

Do you have a story that could inspire others? Are you passionate about a project management area that you would like to share with your peers? Then you are in luck. The marketing and communications staff are looking for articles for the next month's newsletter. Please forward your submittals to marketing@pmipunechapter.org.

Remember, you gain PDUs for writing project management related articles.

EDITORIAL TEAM

Core Volunteer

Raman Udgiri is a volunteer for various portfolios in PMI Pune Chapter and can be reached at ramanudgiri@yahoo.co.uk

Core Volunteer

Ashutosh Nadkarni is a volunteer in PMI Pune Chapter for Marketing & Communication Team and is our Social Media Manager and can be reached at ash.nadkarni@gmail.com

PMP/ RENEWALS FOR MONTH

Renewals/ Joinees for Chapter

Abhijeet Chitrao
Abhijit Patil
Ashishi Ramteke
Damodar Phatak
Harshad Bhingarkar
Khalid Ahmad Abdul Hameed Khan
Manikandan Hariharan
Prasad Jadhav
Prasad S Vesanekar
Pravin Ukey
AMIT KULKARNI
Arti Arun Kulkarni
Ashishh Khadse Jr.
ASHOK RATANRAO PATIL
Christopher John
Devendra Solanki
Fakhruddin Ali Ahmed
Ganesh Shetty
Harsha Shembekar
Jeevan Vadhavkar

Jinesh Banugaria
KETAN TAILOR
Kiran Munagekar
Madan M. Pathak
Madhavi Ramachandran
Mayuresh Vedpathak
Milind Bade
PRASAD DIXIT
Priyanka Pradeep Rokde
Rajen Kumar Tripathy
Sachin Khot
Sachin Patil
SANTOSH JAGTAP
Sarang Dhoble
shallesh tater
Shubhendra Singh Singh
Sudharsan Sundarasami
Sumit Ghosh
Tarun Jaiswal
Vinayak Kamath
Vineet Saxena Eng.D.
Vivek Kumar Boharpi
Vivek Nashine

About PMI Pune Deccan Chapter

PMI Pune Deccan Chapter is committed towards Supporting PMI's goal: "Worldwide, organizations will embrace, value and utilize project management and attribute their success to it." To promote project management discipline in the region. To provide various knowledge enriching benefits to the members of PMI Pune chapter. To promote Project Management as a functional discipline in Engineering & Management schools with active participation and support from PMI Pune Chapter

Web & Social Media:

Website: http://www.pmipunechapter.org/

Yahoo Group: http://finance.groups.yahoo.com/group/pmi-pune-chapter/

Facebook Page - http://www.facebook.com/pages/PMI-Pune-Deccan-India-Chapter/340354512676308

LinkedIn - Group: http://www.linkedin.com/groups?gid=4047678&trk=hb_side_g

LinkedIn – Follow us - http://www.linkedin.com/company/pmi-pune-deccan-india-chapter

Twitter: #pmi_punechapter | http://twitter.com/pmi_punechapter

Advertise with Us

Advertise with us and reach a community of 9000+ focused group! If you or your organization would like to give Advertisements in our Chapter's Quarterly Newsletter "On Target" or would like to Advertise on chapter's website then contact our Marketing & Communications Director: marketing@pmipunechapter.org

Sponsorship

If you or your organization would like to sponsor Chapter Activities / Events / Workshops / Gifts, or would like to know what you can sponsor then contact our Corporate Relations Director: corporate@pmipunechapter.org

Volunteers

PMI Pune chapter is a non-profit professional organization managed by volunteers. All members are encouraged to get involved in Chapter activities as a volunteer and take advantage of the valuable opportunity to learn and grow professionally/personally and at same time earn Professional Development Units (PDUs). Only with continued support from the members, Chapter is able to provide year-round variety of project management related activities. If you are interested in volunteering, please send mail to our Director Volunteer Management: at volunteer@pmipunechapter.org

Contacts

Advertisements / Media: Rinoo Rajesh, marketing@pmipunechapter.org

Memberships: membership@pmipunechapter.org

General Information: Omkar Gurjar, info@pmipunechapter.org

Good Things Happen When You Join PMI – Pune Chapter!   Connect with us!!!