

Peopeware – The Fundamental Factor for High Performing Software

Dr. I. B. Lal¹, Dr. Shivkant Kumar², Dr. Vibhawendra Pathak³

^{1,2}Assistant Professor, ³Associate Professor, L.N. Mishra College of Business Management, Muzaffarpur, India

Abstract-- Human resource is the most important ingredient in the process of software development. Success of a software depends on the development process which is accomplished by humans – the peopeware, constituting a ‘team’. The effectiveness of the team decides the quality of the software. The internal composition of a work team is an important issue regarding team performance and the criteria used to select team members play an important role in determining team composition. However, there are only a handful of empirical studies about the use of team building criteria in the software industry. The objective of this article is to study and identify the factors that could be considered in industrial practices to build a software project team which would excel in its performance to develop high performing reliable software.

Keywords-- team, group, work team, work group, people management, capability maturity model.

I. INTRODUCTION

Software, now-a-days, has entered every walk of human life, from kitchen to war field, from machine to market, and much more. It is changing the human life style enormously throughout the world. Hence, delivery of highly reliable, fault free, quality software has become a challenging issue for the software professionals and software companies. Software organisations face the problem of rejections of software products after huge amount has been invested or they are bound to make changes in the software after the product has been delivered – commonly called maintenance (corrective or adaptive) cost, that sums to a huge total. Besides, with the increasing applicability of the software in the human life, the users frequently change their requirements and still intend to get a product that is fault free and operates as they intend. Simultaneously, the technology is changing drastically at rapid pace which compels the users and so the software developers to update the software and make it compatible to the changing needs of technology.

In the early days, software development was just a task of one or two person(s) but this is not the case today. Now, for a large project it is quite impossible for a person or two to complete such a large project.

Today the software sizes are developed in millions LOC (e.g. UNIX, Windows, etc.) while some would be needing billions LOC in the coming days as the needs of today is integration of multiple areas of operations. These days even in small projects it is normal practice to use a team to leverage the (hopefully) complementary set of skills that a group of developers may bring to a project. Another important point is that the software of today is not based on using one or two languages, rather now-a-days it requires multiple platforms, such as Java, HTML/XML, PHP, etc. Most of the times it becomes difficult to search a person with the expertise in multiple language platforms. Now it requires a best mix of skill to develop high quality reliable software. Therefore, all these situations bind us to have proper people management in the organisation. For better or worse, it is, practically, rather difficult for any software organisation to make modifications in its organisational structure for various reasons. This is also not in the purview of the Project Managers. The important thing for project managers is to select and form a best performing team with a good mix of skill. Selection of best development team is a managerial style considering various factors. Teams are a primary mechanism for accomplishing organizational work, especially for software projects (Faraj and Sproull 2000), hence team building, team size and cooperation between team members are critical factors in providing a quality software project. Team building process has been regularly investigated over a long period, and a wide range of methods have been implemented to improve team cohesion and quality (Yasar Guneri SAHIN, 2013). In this paper authors have attempted to study various considerations for building a high performing software development team.

II. PEOPLE MANAGEMENT

Out of the four resources – man, money, machine and material, man has been found to be the most important resource. Besides, amongst the 4Ps – People, Product, Process and the Project- the ‘People’ again sits at the top. Only the talented people can continually grow to improve the process and thus the product, which is the dire need in the field of software development.

The cultivation of motivated, highly skilled software people has been discussed since the 1960s (Curtis, 1994, DeMarco, 1987, Whitakar, 1994). How much “people factor” is important it is well explained by Software Engineering Institute (SEI) in its model – People Management Capability Maturity Model (PM-CMM). PM-CMM stresses on “to enhance the readiness of software organisations to undertake increasingly complex applications by helping to attract, grow, motivate, deploy, and retain the talent needed to improve their software development capability” (Curtis, 1994). The simple idea behind this model is that if we have good, skilled and innovative people, definitely the process would improve continuously, leading to the development of a good product. Hence it is the fundamental and perhaps the most important task of software organisations to adopt the people management maturity model which defines the following key practice areas for software people: recruiting, selection, performance management, training, compensation, career development, organisation and work design, and team/culture development. Organisations that achieve high levels of maturity in the people management area have a higher likelihood of implementing effective software engineering practices. (Pressman, 97). PM-CMM can be used as a framework for improving the way in which an organisation manages its human assets. Like Capability Maturity Model (CMM) of software process, PM-CMM is also a five-level model :

1. *Initial*: Ad hoc, informal people management practices,
2. *Repeatable*: Establishment of policies for developing the capability of the staff,
3. *Defined*: Standardisation of best people management practice across the organisation,
4. *Managed*: Quantitative goals for people management,
5. *Optimizing*: Continuous focus on improving individual competence and workforce motivation.

The strategic objectives of the PM-CMM are to enhance the capability of the people workforce and thus the organisation, and to align the individuals with the organisation. (Sommerville, 2009). For a software organisation it becomes important to attract and recruit the right skilled practitioners, select, appoint and delegate the right people to the right job, form a right group/team with best mix of skills according to the job with the high level of communication and cooperation amongst the people of the team/group, and create an environment in which the people could work, learn, improve and develop themselves.

This would result in continuous improvement in the process which would ultimately benefit the organisation. One important thing is the perfect size of the group/team which may range from two to hundreds depending on the size of the project. As the software products are different from other physical products, pace of the software development cannot be catalyzed by adding more and more people, at the same time, it has been observed that adding people to a project late in the development cycle leads to further delays in the schedule (Ghezzi *et. al.*, 2005).

Unlike other physical products construction, James Over (2010) states - “*Software development is knowledge work*”. The key rule in managing knowledge work is this: “*managers can’t manage it, the workers must manage themselves.*” To manage software work, developers must

- Be motivated
- make accurate plans
- negotiate commitments
- track their plans, and
- manage quality

Watts Humphrey (1997) who has been the developer of Personal Software Process (PSP) and Team Software Process (TSP) comments: “*The PSP [and TSP] will show you how to plan and track your work and how to consistently produce high quality software. Using PSP [and TSP] will give you the data that show the effectiveness of your work and identify your strengths and weaknesses...To have a successful and rewarding career, you need to know your skills and abilities, strive to improve them, and capitalize on your unique talents in the work you do.*” (Courtesy: Pressman, 2014).

Management scientists have suggested different organisational structures in different situations that affect the productivity and effectiveness of the people/group. The nature of software distinguishes it from other human artefacts to the extent that it influences the effectiveness of different organisational structures applied to software development (Ghezzi *et. al.*, 2005).

III. GROUP

Stephen P. Robbins defines a **group** as “*two or more individuals, interacting and interdependent, who have come together to achieve particular objectives*”. It can also be defined as “*A collection of individuals who have regular contact and frequent interaction, mutual influence, common feeling of camaraderie, and who work together to achieve a common set of goals*”. A group can be formal or informal.

A **formal group** is defined by organisation's structure with designated work assignments and directed toward organisational goals; whereas, an **informal group** is a natural formation in the work environment, which are neither formally structured nor organisationally determined. A **command group** is determined by the organisation chart where subordinates directly report to their seniors/managers. **Task group** is also organisationally determined and is formed by those people who are working together to complete a job task. An **interest group** is formed people who do not have common command or task groups but they come together and affiliate to attain a specific objective.

Group Structure: Work groups are not unorganised mobs. They have a defined structure with some objectives to accomplish, each member having some behaviour integrating into group behaviour and thus the group effort. Why some groups are successful over others it depends on the group cohesiveness, ability of the members of the group, the level of conflict, level of communication, and the internal pressures to comply with the group's norms. Every work group is highly influenced by the external factors imposed from outside. The success of group effort depends on the intelligence and motivation of individual group members. The structural variables of performance of the group comprise – formal leadership, roles, norms, group status, group size, composition of the group, and the degree of group cohesiveness.

IV. TEAM

A **Team** can be defined as

1. "a group of people with a full set of complementary skills required to complete a task, job, or project".
2. A group of people with different skills and different tasks, who work together on a common project, service, or goal, with a mesh of functions and mutual support.
3. A Team is a small group in which members have common purpose, complementary skills and interdependent roles (Gondal and Khan, 2008).
4. Katzenbach and Smith [2005] define team as follows "A team is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable".

Team members (1) operate with a high degree of interdependence, (2) share authority and responsibility for self-management, (3) are accountable for the collective performance, and (4) work toward a common goal and shared rewards(s).

A team becomes more than just a collection of people when a strong sense of mutual commitment creates synergy, thus generating performance greater than the sum of the performance of its individual members, i.e. $2+2 \neq 4$, rather >4 .

V. TEAM VS. GROUP

A work group is group that interacts primarily to share information and to make decisions to help each member perform within his or her area of responsibilities. Work groups have no need or opportunity to engage in collective work that requires joint effort. So their performance is merely the summation of each group member's individual effort. There is no positive synergy that would create an overall level of performance that is greater than the sum of the inputs. A work team generates positive synergy through coordinated effort. Their individual efforts result in a level of performance that is greater than the sum total.

VI. TEAM BUILDING IN SOFTWARE PROJECTS

Software organisations have restructured their work processes around teams with a view to looking towards positive synergy to enhance the performance. Teams are supposed to be better suitable for executing complex tasks because team members share workload, observe behaviour of other team members and contribute to the sub tasks of the complex task (Mathieu, Heffner, Goodwin, Salas and Cannon-Bowers, 2000). The extensive use of teams creates a potential to generate enhanced outputs without increasing the inputs. Technological advancements have catalyzed the usage of teams in modern software organizations. Based on the literature on teams, less is known about how to improve the team performance (Bolstad and Endsley, 2000). Researchers agree that team composition is built from the combination of individual team member characteristics and the role these individuals play in the team (Maznevski, et.al.,1994). Therefore, the process of building a work team starts with the identification of individuals with the characteristics needed to create the desired composition in a given context (Maznevski et.al, 2001). In software development, Weinberg (1971) was one of the first authors to consider programming as an individual and social activity, i.e., developed by individuals through teamwork. The "Chief Programmer Teams" (Baker,1972) and the "Surgical Teams" (Brooks,1975) were the first models that dealt explicitly with the structure and composition of software teams. Since then, software teams have been a focus of research by academics and practitioners (Boehm,1981, Curtis,2001).

Lettice and McCracken (2007) have reported that the amount of researches related to software team doubled between 1997 and 2007. The increasing interest in management of software teams stems from its reported effects on productivity (Boehm,81), product quality (McConnell, 1998), and project failure (Curtis,2001).

The first contextual study regarding team building, called “Tuckman’s model of team development” (Tuckman 1965), proposed a four-stage model, consisting of form, storm, norm and perform sequence. These stages are regarded as the best and idealized forms (Buchanan and Huczynski, 1997). Besides, various other models on team building have been proposed by many management scientists and schools of management. Understanding the importance of team, team composition in software field is very important. Size of the team is an issue. It is influenced by the characteristics of the software. If a group of modules involve high coupling, assigning the module to different people will require too much interaction among the practitioners. Thus an appropriate design must be accompanied by an appropriate assignment of tasks to individuals and appropriate team organisation that makes the assignment possible. One must have a flexible approach to choose an appropriate organisation based on the design of the system. The total number of members in a team should be exactly the same as estimated, although, it is very difficult to get absolute estimate of number of team members in a team due to number of estimation drivers. If the number of team members is less than the estimate, it has been termed as problem of capacity leading to increased cost, time and effort with poor quality of the software product whereas number of team members is more than that of estimate, it has been termed as problem of throughput; again leading to increased cost, time and effort with poor quality of the software product. The members of team should be selected from varied areas/skills to make a right mix. Team composition based on role allocation has been investigated from different perspectives in software engineering. Therefore, it is necessary to understand what constitutes effectiveness of a work team and derive a conceptual definition of team effectiveness to be operationalised.

Several authors, including Shneiderman(1980), De Marco and Lister(1999), Constantine (2001), Guinan et al.(1998) have investigated and described personal and social factors that can affect the effectiveness of software teams, including: the interaction between the personality of team members, role-related diversity, the effects of the work environment, organization and structure of teams, team processes such as communication, conflict and cohesion, among others.

Mantei (1981) describes seven project factors that should be considered when planning the structure of software engineering teams:

- Difficulty of the problem to be solved
- “Size” of the resultant program(s) in lines of code or function points
- Time that the team will stay together (team lifetime)
- Degree to which the problem can be modularised
- Required quality and reliability of the system to be built
- Rigidity of the delivery date
- Degree of sociability (communication) required for the project

To form a high performing software team:

- Team members must have trust in one another
- The distribution of skills must be appropriate to the problem
- Mavericks may have to be excluded from the team, if team cohesiveness is to be maintained. (Pressman, 2014)

VII. CHARACTERISTICS OF AN EFFECTIVE WORK TEAM

Software development is an intensive team work with high level of cohesiveness and communication among the team members. Here we can enlist some of the features of an effective work team. These can be:

1. *Clear Task*: The task or objective of the group is well understood and accepted by the group.
2. *Informality*: The “atmosphere” tends to be informal, comfortable, relaxed. There are no obvious tensions or signs of boredom.
3. *Participation*: There is a lot of discussion in which virtually everyone participates, but it remains pertinent to the task of the group.
4. *Listening*: The members listen to each other. Every idea is given a hearing.
5. *Disagreement*: There is no disagreement but the team is comfortable with this and shows no signs of avoiding conflict to keep everything on a plane of sweetness and light.
6. *Consensus*: Most decisions are reached by consensus; formal voting is kept to a minimum.
7. *Open communication*: Team members feel free to express their feelings on the task as well as the group’s operation. There is little “pussy-footing” and few hidden agendas.
8. *Clear assignment*: When action is taken clear assignments are made and accepted.

9. *Shared leadership*: While the team has a formal leader, leadership functions shift from time to time depending upon the circumstances, the needs of the group and the skills of the members.
10. *Self-assessment*: Periodically, the team stops to examine how well it is functioning and what may be interfering with its effectiveness.(Parker & Kropp, 2009).

The Project Managers decide the selection of best team. They keep in mind the personality diversification as well as skill requirement. Empirical studies investigating the effect of personality diversity in the composition of software teams have produced a wide range of evidences (Cruz et.al., 2011). For instance, Bradley and Herbert(97) proposed and empirically tested a model to relate personality to team effectiveness. In their model, team performance was affected by four factors: leadership, team cohesion, communication, and heterogeneity of individual characteristics. Leadership largely affect the performance of a team. In the field of software development another important factor is communication among the team members, whether it is vertical, horizontal, or mix of both vertical or horizontal. According to Chudoba, Lu, Watson Manheim and Wynn (2003), communication, interpersonal relationships, team member participation, team member commitment and outcomes impact the team performance. Based on their research at Intel, they identified that the three factors that affect the virtual team performance are social interactivity, knowledge networking and work predictability. Trust between team members, communication effectiveness, comfort level of team members, motivation of team members, and cohesion between team members have impact on team performance when teams are distributed geographically (Sridhar, Paul, Nath and Kapur, 2007).

DeMarco and Lister contend that members of jelled teams are significantly more productive and more motivated than average. They share a common goal, a common culture, and a sense of eliteness that makes them unique. But not all teams jell. Jackman (1998) states many teams suffer from “team toxicity”. She defines five factors that foster a potentially toxic team environment”:

1. a frenzied work atmosphere,
2. high frustration that causes friction among team members,
3. a “fragmented or poorly coordinated” software process,
4. an unclear definition of roles on the software team, and
5. continuous and repeated exposure to failure.

To avoid a frenzied work environment the project managers must ensure that they are getting all information required to accomplish the job.(Pressman, 2014).

VIII. CONCLUSION

The success of the software life depends on the quality of the software developed, which depends on the process adopted to develop a quality software, that ultimately depends on the performance of the software team. Therefore, the constitution of a right team is very essential. Today, the software organisations, project managers and practitioners exert enormous pressure for the delivery of a reliable quality software when the requirements are frequently changing the world over under high constraints of resources like time, money, and effort. The project managers can use the assessment techniques to measure their software development team performance [Pattit and Wilemon, 2005] to monitor the performance of the team and correct any disagreement from the norms. They can use software metrics (both project and process metrics) to assess the performance of the team members individually and team as a whole. The team cohesiveness, communication, motivation, reward and punish strategy also become important factors for delivery of high performance of a team.

REFERENCES

- [1] Baker,F.,1972. “Chief programmer team management of production programming”, IBM Systems Journal 11 (1),56–73.
- [2] Boehm, B.W., 1981. “Software Engineering Economics”, Prentice-Hall, Inc., Englewood Cliffs.
- [3] Bolstad, C.A. and Endsley,M.R., 2000. “Shared Displays and Team Performance”, Presented at Human Performance, Situation Awareness and Automation Conference, Savannah, GA, October.
- [4] Bradley, J.H. and Hebert, F.J.,1997. “The effect of personality type on team performance”, Journal of Management Development 16 (5), 337–353.
- [5] Brooks,F.,1975. “The Mythical Man-Month, Addison-Wesley”, ISBN: 201-00650- 2.
- [6] Buchanan, D. & Huczynski, A.,1997. “Organizational Behaviour: An Introductory Text”, 3rd ed., Prentice-Hall, London.
- [7] Chudoba, K.M., Lu,M., Watson-Manheim, M.B. and Wynn,E., 2003. “How Virtual are we? Measuring Virtuality and Understanding Its Impact in a Global Organization”, Proceeding of the International Conference on Information Systems, Dec.
- [8] Constantine,L.,2001. Peopleware Papers: The Notes on the Human Side of Software, Prentice Hall.
- [9] Cruz, S.J.O., B. da Silva,F.Q., Monteiro,C.V.F., Santos,P.C.F., M. dos Santos,I.R., 2011. “Personality in software engineering: preliminary findings from a systematic review”, in: Proceedings of the 15th Annual Conference on Evaluation and Assessment in Software Engineering (EASE 2011), Durham, 2011. doi:10.1049/ic.2011.0001.

International Journal of Emerging Technology and Advanced Engineering

Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 12, December 2016)

- [10] Curtis, B., et.al., 1994. "People Management Capability Maturity Model", Software Engineering Institute, Pittsburgh, PA.
- [11] Curtis, B., Hefley, W.E., and Miller, S.A., 2001. People Capability Model (P-CMM), Version 2.0, Tech. Report CMU/SEI-2001-MM-001, Software Eng. Inst., Carnegie Mellon Univ.
- [12] DeMarco, T. and Lister, T., 1987. "Peopleware", Dorset House.
- [13] DeMarco, T. and Lister, T., 1999. "Peopleware: Productive Projects and Teams", Dorset House Publishing Co., New York, 1999
- [14] Faraj, S. & Sproull, L., 2000. "Coordinating expertise in software development teams", *Management Science*, 46, 1554-1568.
- [15] Ghezzi, Carlo, Jazayeri, M., Mandrioli, D., 2005. "Fundamentals of Software Engineering", Pearson Education, 2nd Indian Ed., N. Delhi.
- [16] Gondal, A.M. and Khan, A., 2008. "Impact of Team Empowerment on Team Performance Case of The Telecommunications Industry in Islamabad", *International Review of Business Research Papers*, Vol. 4, No.5, Oct-Nov, pp. 138-146.
- [17] Guinan, P.J., Coopridge, J.G. and Faraj, S., 1998. "Enabling software development team performance during requirements definition: a behavioral versus technical approach", *Information System Research* 9 (2), 101-125.
- [18] Hogan James M. and Thomas, Richard, 2005. "Developing the Software Engineering Team", Australian Computer Society, Inc., The Australasian Computing Education Conference, Newcastle, Australia. *Conferences in Research and Practice in Information Technology*, Vol. 42. Alison Young and Denise Tolhurst, Eds.
- [19] Horii, T., Jin, Y. and Levitt, R. E., 2005. "Modeling and Analyzing Cultural Influences on Project Team Performance", *Computational & Mathematical Organization Theory*, Volume 10, Number 4, pp. 305-321.
- [20] Humphrey, W., 2000. "Introduction to the Team Software Process", Addison Wesley.
- [21] Jackman, M., 1998. "Homeopathic Remedies for Team Toxicity", *IEEE Software*, pp. 43-45.
- [22] Katzenbach, J.R. and Smith, D.K., 2005. "The Wisdom of Teams: Creating the High Performance Organization", Collins Business Essentials, An Imprint of HarperCollins Publishers, NY.
- [23] Lettice, F., McCracken, M., 2007. "Team performance management: a review and look forward", *Team Performance Management*, 13 (5/6), 148-159.
- [24] Mantei, M., 1981. "The Effect of programming team structures on programming tasks", *CACM*, vol.24, no.3, March, pp. 106-113.
- [25] Mathieu, J.E., Heffner, T.S., Goodwin, G.F., Salas, E. and J.A. Cannon-Bowers, 2000. "The Influence of Shared Mental Models on Team Process and Performance", *Journal of Applied Psychology*, Vol. 85, No. 2, pp 273-283.
- [26] Maznevski, M.L., 1994. "Understanding our differences: performance in decision making groups with diverse members", *Human Relations* 47(5), 531-552.
- [27] McConnell, S., 1998. "Problem programmers", *IEEE Software*, 15 (2), 126-128.
- [28] Over, J., 2010. "Introduction to the Team Software Process", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- [29] Parker, G.M. and Kropp, R.P., 2009. "Team Building: Book of Activities", Viva Books Pub., London.
- [30] Pattit, J.M. and Wilemon, D., 2005. "Creating high-performing software development teams", *R&D Management*, vol. 35, No.4, pp.375-393.
- [31] Pressman, Roger S., 1997, 2014. "Software Engineering: A Practitioner's Approach", McGraw-Hill, 4th & 7th Ed.
- [32] Q.B. da Silva, Fabio, A. César C. França, Marcos Suassuna, Leila M.R. de Sousa Mariz, Isabella Rossiley, Regina C.G. de Miranda, Tatiana B. Gouveia, Cleviton V.F. Monteiro, Evisson Lucena, Elisa S.F. Cardozo, Edval Espindola, 2013. "Team building criteria in software projects: A mix-method replicated study", *Information and Software Technology*, 55, 1316-1340.
- [33] Robbins, Stephen P., 1999. "Organisational Behaviour", Prentice Hall of India, 8th ed., New Delhi.
- [34] Salgado, J.F., Viswesvaran, C. and Ones, D.S., 2001. "Predictors used for personnel selection: an overview of constructs, methods, and techniques", in: N. Anderson, D.S. Ones, H.K. Sinangil, C. Viswesvaran (Eds.), *Handbook of Industrial Work & Organizational Psychology*, vol. 1, SAGE, London.
- [35] Sommerville, Ian, 2007. "Software Engineering", 8th ed., Pearson Education Pub., New Delhi.
- [36] Sridhar, V., Paul, R. Nath, D. and Kapur, K., 2007. "Analyzing Factors that affect performance of global virtual teams", Second International Conference on Management of Globally Distributed Work, Indian Institute of Management, Bangalore, India.
- [37] Sudhakar, G. P., "Understanding Software Development Team Performance". (<http://anale.feaa.uaic.ro/anale/resurse/inf1sudhakar.pdf>)
- [38] Tuckman, B., 1965. "Developmental sequence in small groups", *Psychological Bulletin*, 63, 384-399.
- [39] Weinberg, G.M., 1971. "The Psychology of Computer Programming", silver anniversary ed., first ed., Dorset House Publishing.
- [40] Whitaker, K., 1994. "Managing Software Maniacs", Wiley.
- [41] Yasar Guneri SAHIN, 2013. "A team building model for software engineering courses term projects", *Computers & Education* 56(3):916-922, June 2013.