Building Productive Teams

by Watts S. Humphrey
Software Engineering Institute, Carnegie Mellon University

The Software Engineering Institute's (SEI) Team Software Process (TSP^SM^) is a framework and a process structure for building and guiding integrated engineering teams, which are essential in development of today's complex, increasingly sophisticated systems. This paper discusses characteristics of productive teams, how to build/launch such a team, and team member preparation.

Teams are required for most engineering projects. While some small hardware or software products can be developed by individuals, the scale and complexity of modern systems is such, and the demand for short schedules so great, that it is not practical for one person to do the entire job. Systems development is a team activity, and the effectiveness of the team largely determines the quality of the engineering.

Modern systems are becoming increasingly sophisticated. Aircraft, automobiles, computer printers, television sets, and even electric razors contain software, often lots of software, and the amounts of software have been rapidly increasing. The design of such systems is vastly more complex than it was a few years ago. While there are still many modest-sized systems, the trend is for the software content of just about every product to increase by 10 or more times every 10 years. This trend has been more or less followed for several decades, and it appears likely to continue for the foreseeable future.

While most large systems involve many technologies, it is generally the software that integrates all the pieces into a cohesive whole. The software engineers provide the glue that holds the system together. It is critical that the software professionals use disciplined methods. If they do not, the integration job and the software glue that binds the system will likely have quality problems.

While every technology is important, if the software people do not properly plan and manage their work, the project will almost certainly get into trouble. That is the reason that the Software Engineering Institute (SEI) developed a framework and a process structure for building and guiding integrated engineering teams. It is called the Team Software Process (TSP^{SM}).

The TSP is one of a family of methods that can help engineering teams more effectively develop and support software-intensive systems. The Capability Maturity Model (CMM®) provides the overall improvement framework needed for effective engineering work [1]. The Personal Software Process (PSP^{SM}) provides the engineering disciplines engineers need to consistently use a defined, planned, and measured process [2]. The TSP couples the principles of integrated product teams with the PSP and CMM® methods to produce highly productive teams.

A team is more than just a group of people who happen to work together. Teamwork takes practice and involves special skills. Teams require common processes; they need agreed-upon goals; and they need effective guidance and leadership. The methods for guiding and leading such teams are well known, but they are not obvious. The SEI has developed the TSP to guide engineers and their managers in using effective teamwork methods.

This paper describes the principles behind the TSP's development. Starting with an overview of the characteristics of productive teams, the paper describes how organizations can build teams that have these characteristics. Next, the paper describes the ways the TSP process guides team formation. It closes with a summary of the preparation required for TSP team members.

The Characteristics of Productive Teams

There are different kinds of teams. In sports, for example, a basketball team's positions are dynamic while baseball team members have more static roles. In both cases, however, the members must all work together cooperatively. Conversely, wrestling and track teams are composed of individual competitors who, while not dynamically interacting, support each other socially and emotionally.

In engineering, development teams often behave much like baseball or basketball teams. While they may have multiple specialties, all the members work toward a single objective. On systems maintenance and enhancement teams, however, the engineers often work relatively independently, much like wrestling and track teams. However, regardless of the team type, productive engineering teams have certain common characteristics.

A team is a group of people who share a common goal. They must all be committed to this goal and have a common working framework. The following definition for a team has been adapted from Jean L. Dyer [3]:

- A team consists of at least two people.
- They work toward a common goal.
- Each person has been assigned specific roles.
- Completion of the mission requires some form of dependency among the group members.

Conditions for Effective Teamwork

The four parts of this definition of a team are all important. For example, it is obvious that a team must have more than one member, and the need for common goals is also generally accepted. It is not as obvious, however, why team members must have roles. Roles are essential because they provide a sense of ownership and belonging. They help guide team members on how to do their jobs; they prevent conflicts, duplicate work, and wasted effort; and they provide the members a degree of control over their working environment. Such a sense of control is a fundamental requirement for motivated and energetic team members.

Interdependence is also important. This is where each team member depends to some degree on the performance of the other members. Interdependence improves individual performance because, with complementary skills, the members can help and support each other. For example, design teams generally produce better designs than any individual member could have produced alone. Team performance is further enhanced by the social support of membership. Human beings are social animals and few people like to work entirely by themselves, at least not for very long. Because of this social context, the members will make a special effort to meet their obligations to the rest of the team.

Through mutual support and interdependence, teams become more than just the sum of their members. As teams build a trusting and cohesive relationship, they develop a spirit and an energy that can produce extraordinary results.

Innovative Teams

Another characteristic of productive teams is their ability to innovate. Innovation has been essential in the development of modern society. Innovation is more than just thinking up bright ideas, it requires creativity and a lot of hard work. Just about every engineering task is part of an innovative endeavor. This is true for the development, enhancement, and repair of complex systems. These innovative teams must produce quality products while using unfamiliar and often unproven tools and technologies. They often start projects with only partially defined needs and they must be sensitive to the user's evolving requirements.

Innovative teams must have skilled and capable people who are highly motivated. They must be creative, flexible, and disciplined. They must strive to meet demanding schedules while adjusting to changing user needs. They must also control costs and schedules while keeping management informed of their progress. In short, innovative teams have a great deal to do.

A Trusting Environment

To be creative and productive, engineering teams must work in a trusting and supportive environment [4]. Engineering teams are composed of extremely capable people who can quickly sense a lack of trust. When managers do not trust their teams to make aggressive schedules or to strive to meet these schedules, the engineers will know it. When engineers do not feel trusted and respected, they will feel antagonized and manipulated. They will no longer feel loyal to the organization and can easily lose their commitment to the team.

Since people are generally more productive when faced with an important and meaningful challenge, it is appropriate for management to challenge their teams with aggressive goals. But when the teams respond to the challenge with a plan, management must be willing to negotiate realistic commitments the engineers believe they can meet. Few people will work diligently to meet a seemingly hopeless project schedule.

Producing Productive Teams

In summary, the basic conditions for productive teams are that the members have defined roles, their work is interdependent, they are skilled and highly motivated, and they work in a trusting environment. To achieve these conditions, there are a number of well-known methods [3, 5, 6, 7, 8, 9, 10]. The team-building principles used by TSP are:

- The team members establish common goals and defined roles.
- The team develops an agreed strategy.
- The team members define a common process for their work.
- All team members participate in producing the plan and they each know their personal roles in that plan.
- The team negotiates this plan with management.
- Management reviews and accepts this plan.
- The team members do the job the way they have planned to do it.
- The team members communicate freely and often.
- The team forms a cohesive group, members cooperate, and they are all committed to meeting the goal.
- The engineers know their status, get feedback on their work, and have leadership that sustains motivation.

Effective team formation requires that the members truly understand what they are supposed to do, agree on how to do the job, and believe that their plan is achievable. These conditions can all be established by involving the engineers in producing their own plans. Then, assuming that their plans are competently made, teams can almost always sell their plans to management.

While all these conditions are necessary for effective teamwork, the specific ways to establish these conditions are not obvious. The TSP provides the explicit training and guidance organizations need to build productive engineering teams.

Launching TSP Teams

TSP projects start with a four-day launch during which the team members make a detailed plan for their project. A trained coach leads the team through determining goals, assigning member roles, making plans, and assessing project risks. By following the PSP planning process and using any available historical data, the engineers are able to make a realistic plan and schedule for their work. Once the team members have built team and personal plans, the team leader holds a management meeting where the team presents the plan to senior management and negotiates an agreed schedule commitment.

Cooperation, Cohesion, and Commitment

While most of the steps in the TSP process involve producing specific things, one condition does not. This is:

- The team forms a cohesive group, members cooperate, and they are all committed to meeting the goal.

These team-member attitudes cannot be legislated, imposed, or established by fiat; they must be created by the team itself. If the team members do not want to cooperate, or if they do not wish to act like a close-knit group, they will not, and telling them to do so will not fix the problem. Commitment is an attitude. When people are truly committed, they behave differently than people who are merely following orders.

While it is clear that cooperation, cohesion, and commitment are necessary, it is not obvious how to produce them. The TSP approach for doing this involves all the team members in producing their own plans and selling these plans to management. People like to work together, they enjoy close-knit and cohesive groups, and they respond to challenging goals and objectives. Unless the working conditions actually block team formation, and as long as the members do not have serious personal antagonisms or emotional problems, such teams will generally become cooperative, cohesive, and committed units.

Team Member Preparation

The first and most fundamental step in the teambuilding process is to ensure that all the members are capable of doing the job. This requires that they have proper skills and abilities and a common set of processes, methods, and terminology. For example, in forming a ball team, you would not pick players from different sports. While they might all be outstanding athletes, a ball team composed of football, baseball, basketball, and soccer players would not likely win many championships. They would not have the common language, agreed rules, or supportive skills to cooperate effectively. Just as on a ball team, TSP team members need a shared process, supportive skills, and the ability to work in an interdependent team environment.

In forming a TSP team, a key requirement is that all the team members understand the principles behind the TSP methods. They must be able to plan and track their work and measure and manage the quality of their products. Training in the Personal Software Process (PSP) provides team members with such knowledge and skill. The major topics that team members need to understand are the following:

- Project planning.
- Status reporting.
- Time, size, and defect measures.
- Quality planning and management.
- Design and design verification.
- Process definition, use, and improvement.

Managers also need to understand these items so they can lead and guide their teams. If the managers are not PSP trained or if the engineers do not know how and why to plan, there is no point in trying to build the group into a cohesive team. The TSP teambuilding process will not work. When the team members do not know how to make plans, they cannot participate in the planning. The team leaders and project managers must then produce the plans. While these may be very good plans, the engineers will not have been involved and they will not be personally committed to the plans.

Conclusions

While the PSP and TSP are relatively new, the experience to date has been promising [11, 12, 13, 14]. The SEI and a growing number of organizations are now qualified to assist industrial and government groups in introducing these methods. There are also an increasing number of universities that teach PSP and TSP courses [2, 15, 16]. For further material on these methods, see the other articles in this issue.

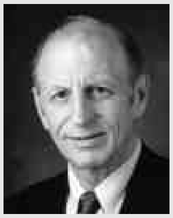
Acknowledgements

The work reported in this paper and several of the other papers in this issue has been developed at the SEI with the support of the Department of Defense. Many people have contributed to this work, but I particularly thank those who reviewed and commented on this paper: Eileen Forrester, Don McAndrews, Jim McHale, Julia Mullaney, Mark Paulk, and Bill Peterson.

References

1. Paulk, Mark C. et al, *The Capability Maturity Model: Guidelines for Improving the Software Process.* Reading, Mass. Addison Wesley, 1995.
2. Humphrey, Watts, *A Discipline for Software Engineering.* Reading, Mass., Addison-Wesley, 1995.
3. Dyer, Jean L, Team Research and Team Training: a state-of-the-art review, *Human Factors Review*, 1984, The Human Factors Society Inc., pp. 286, 309.
4. Shellenbarger, Sue, To Win the Loyalty of Your Employees, Try a Softer Touch, *The Wall Street Journal*, Jan. 26, 2000, page B1.
5. Cummings, Thomas G., Self-Regulating Work Groups: A Socio-Technical Synthesis, *Academy of Management*, vol. 3, no. 3, July 1978, p. 627.
6. DeMarco, Tom and Lister, Tim, *Peopleware, Productive Projects and Teams*. New York: Dorset House Publishing, 2nd. Ed. 1999.
7. Katzenbach, Jon R., and Douglas K. Smith, *The Wisdom of Teams.* Boston, Mass. Harvard Business School Press, 1993, p. 3.
8. Mohrman, Susan Albers, *Designing Team-Based Organizations, New Forms for Knowledge Work*. San Francisco: Jossey-Bass Publishers, 1995, pp. 52, 176, 279.
9. Shaw, Marvin E., Group Dynamics, *The Psychology of Small Group Behavior.* New York: McGraw-Hill, 1981.
10. Stevens, Michael J. and Michael A. Campion, The Knowledge, Skill, and Ability Requirements for Teamwork: Implications for Human Resource Management, *Journal of Management*, Vol. 20, No. 2, 1994.
11. Ferguson, Pat, Watts S. Humphrey, Soheil Khajenoori, Susan Macke, and Annette Matvya, Results of Applying the Personal Software Process, *IEEE Computer*, Vol. 30, No. 5, pp 24-31, May 1997.
12. Hayes, Will, *The Personal Software Process: An Empirical Study of the Impact of PSP on Individual Engineers,* CMU/SEI-97-TR-001.
13. Humphrey, Watts, Using a Defined and Measured Personal Software Process, *IEEE Software*, May 1996, pp. 77-88.
14. Webb, Dave and Humphrey, Watts, Using the TSP on the TaskView Project, CrossTalk, Vol. 12, No. 2, February 1999.
15. Humphrey, Watts, *Introduction to the Personal Software Process*, Addison-Wesley, Reading, Mass., 1997
16. Humphrey, Watts, *Introduction to the Team Software Process,* Addison-Wesley, Reading, Mass., 2000.

About the Author



Watts S. Humphrey joined the Software Engineering Institute (SEI) of Carnegie Mellon University after his retirement from IBM in 1986. While at the SEI, he established the Process Program, led initial development of the Software Capability Maturity Model®, and introduced the concepts of Software Process Assessment and Software Capability Evaluation. Prior to joining the SEI, he spent 27 years with IBM in various technical executive positions such as management of IBM commercial software development, including the first 19 releases of OS/360. He holds graduate degrees in physics from the Illinois Institute of Technology and business administration from the University of Chicago. He is an SEI Fellow, an ACM member, an IEEE Fellow, and a past member of the Malcolm Baldrige National Quality Award Board of Examiners. He has published several books and articles, holds five patents, and received an award for SPI leadership and innovation from the Boing Corp.

SEI, Carnegie Mellon University
4500 5th Ave.
Pittsburgh, Pa. 15213-2612
E-mail: watts@sei.cmu.edu