SEMAT Position Statement

Watts S. Humphrey
The Software Engineering Institute
Carnegie Mellon University

The ultimate objective of the software engineering discipline is to help engineering teams work efficiently and effectively. Therefore, the final criterion for judging the suitability of any tool, method, or process must be its ability to improve engineering team performance. This, in turn, can only be determined objectively by measuring team performance and analyzing the resulting data. To further examine this topic, the first step is to describe the characteristics of high-performing teams.

The Characteristics of High-Performing Teams

The characteristics of high-performing teams are pretty obvious for sports. First, these teams start with highly skilled members, and then they have the following characteristics that are universal for all kinds of teams, whether for development or for sports.

1. **Goal** – Everybody on the team knows the team's goal and what it takes to reach it.
2. **Roles** – All members know their personal roles on the team as well as the roles of all the other team members.
3. **Strategy** – All team members know and agree with the overall team strategy and their role in supporting it.
4. **Process** – Everybody knows how to do their own job and how everybody else does their jobs.
5. **Plan** – Everybody knows what to do at all times and nobody stands around waiting to be told their next assignment.
6. **Support** – Everybody is aware of team workload and is prepared to pitch in and help whenever somebody needs a hand.
7. **Status** – Everybody knows precisely where the team stands at all times and is prepared to make an extra effort whenever needed to achieve overall team success.

High-Performing Development Teams

High-performing teams don't just happen; they are built. This is done by performing a well-known set of activities in preparing for the projects, in doing the development work, in managing that work, and in assessing the results when the job is completed. These activities are as follows.

Preparation Tasks

1. Define the project's goals. What is it that the team is to do?
2. Define the team, its members, its roles, and its scope. What development functions are represented on the team such as testing, software development, hardware development, or systems engineering, and what responsibilities will the team and its members have?
3. Establish the development strategy. How does the team intend to do the job, are prototypes needed, how many releases are required, what cycles are planned, and what is cycle scope and duration?

4. Produce a list the products to be produced and their essential characteristics, like size, function, and principal specifications.
5. Define the development process. How does the team plan to do the work, what methods and practices will it use, what are the entry and exit criteria for each process step, and what data are gathered and used with each activity?
6. Produce the team plan. What are the tasks to be performed for each process step and what effort will be required for each task and product element?
7. Allocate the work among the team members. For the next few months, which team members will be assigned which of the project's tasks and what adjustments are needed to balance team workload?
8. Obtain management agreement to the team plan. Does management agree with the team's plan, are revisions needed, and does the team agree with the revisions?

Development and Development Management Tasks

9. The team performs the development work.
10. As the work proceeds, the team adjusts the plan and work assignments to conform to project status and the team members' current understanding of the work.
11. The team regularly reports its progress to management.
12. The team monitors risks and issues and obtains management assistance in resolving problems that it cannot handle.
13. The team dynamically replans the work as requirements, team membership, product knowledge, and development status change.

Assessment Activities

14. Following completion of each major project milestone, the team analyzes its performance, identifies areas for improvement, gathers data on project results, and documents lessons learned.

Team Guidance

Since development teams are typically under severe schedule pressure and are in a hurry, they are likely to start doing whatever they know how to do. This is typically the development work itself. Since few teams know how to perform any of the project preparation, team organization, or development management tasks, they either skip or defer these tasks or do them incompletely. As a result, these are the areas where most project teams get into trouble. This is the principal reason that project failures are almost universally due to management problems.

The problem is not that the teams have poor managers, but because the teams have not properly prepared for the job and because they cannot provide their managers with precise status information on their work. Without precise status information, managers cannot anticipate problems in time to prevent or correct them. By the time management knows that there are problems, their projects are already in serious trouble and it is generally too late to recover. One approach to this problem has been to devise standard development processes that include all of the required tasks. While these canned processes can be helpful in some cases, this is not a general solution for several reasons.

First, all but the smallest development projects are unique, so that they each need strategies, processes, and plans that are tailored to their specific challenges.

Second, each development team has a particular mix of skills, capabilities, and experiences so that a process that was appropriate for one team would not likely be appropriate for another.

Third, teams learn and grow so that a process that was appropriate for a team on one project at one time would not likely be proper for that same team on a different project or at a different time.

This means that the only practical way for a team to obtain a development strategy and process that truly fits its own specific needs would be for that team to define its own strategy and process. That, in fact, is the role of steps 3 and 5 in the list of project preparation tasks given above.

Operational Process Support

To help teams perform their planning, project management, and technical tasks, they need operational guidance. That is, they need some thing or some person that tells them precisely what to do and when. That is the role of an operational process. An operational process provides a simple and brief description of the steps required to do a specific job. It is in a format that is easy to read and to use, and it is typically on a one-page script or short set of electronic cues. The key is that the guidance be brief and clear and not buried in paragraphs of opaque text in some procedures books.

To be most helpful, operational processes must be brief, be action oriented, be precise, be concise, and be prescriptive. They must also provide explicit guidance on how to perform the project tasks. This means that the work must involve very standardized tasks or the process must be customized to the specific needs of the job and the team. Since every team and every job is different, this further means that every team must customize its own process.

With modest preparation and guidance, most teams can quickly be guided through the steps of establishing their own customized strategies and defining their own customized operational processes. For trained and experienced teams, this entire planning and process definition work can be done in only a few days.

Conclusion

To establish a sound and firmly based software engineering discipline, we must do the following.

1. Start measuring software work and using the resulting data to plan and manage the work and to assess the effectiveness of the tools and methods that were used.
2. To do this consistently and effectively, software engineers must use operational processes to guide their work.
3. For these processes to be consistently and properly used, the development teams themselves must define the operational process that they use.
4. To do this consistently, software engineers must be properly trained and coached.