

Why Teams Need Operational Processes

Watts S. Humphrey

The Software Engineering Institute

Carnegie Mellon University

The ultimate objective of development tools, methods, and processes is to help teams do their work more efficiently and effectively. Therefore, the final criterion for judging the suitability of any tool, method, or process must be its ability to improve team performance. This, in turn, can be determined subjectively by asking team members for their opinions or by measuring team performance and analyzing the resulting data. Clearly, gathering and analyzing objective data is preferable whenever it is possible. In examining this topic, the first step is to describe the characteristics of high-performing teams.

The Characteristics of High-Performing Teams

While the characteristics of high-performing development teams are not always clear, these characteristics are generally pretty obvious for sports teams. The characteristics are universal, however, whether for development or sports teams, and they are as follows.

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 Tasks

9. The team performs the development work.

Development Management Tasks

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 project 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 specific 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. It provides a simple and brief description of the steps required to do this 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 meet the following requirements.

Be Brief. Operational scripts should be limited to one or at most two pages.

Action Oriented. The script statements are in active voice, that is, they start with a verb.

Precise. The statements are clear: who does what, when, and with what result.

Concise. Operational scripts are designed for repetitive use by knowledgeable practitioners and contain no tutorial information. Where tutorials are needed, they are provided before the job starts and are not included in the scripts.

Prescriptive. The scripts define their own entry and exit criteria so that users can quickly tell if they have the required prerequisites and if they have completed all of the specified tasks.

General Purpose vs. Custom Scripts

Process scripts are only useful if they provide explicit guidance on how to perform a task. This means that they must be either used for very standardized tasks, or they must be customized. As noted above, however, that means that, except in very special circumstances, a team's development work must be guided by customized process scripts. This further means that the team must customize that process itself since no one else can know precisely how its members

intend to do the job. Finally, unless the team knows how to define a customized operational process, it will almost certainly end up working without the guidance such a process could provide.

While this may not be a serious problem for small teams that have worked together on many prior projects, most teams are newly formed for each development project. In these cases, the team members will each have their own views of how the job should be done, and, without some process definition step, they will all typically follow different development processes. Under these conditions, the team members would then not know what their team mates are doing at any time, and be unable to provide support in case there were problems. The consequences of such random development processes are even more severe when it comes to determining project status, managing product quality, incorporating new team members, or adjusting the team's plan when the requirements change.

Clearly, one essential requirement for a high-performing team is that the members all agree on how to do the job, and that means that they must have actually taken the time to define the process they intend to follow. Unless they do, the team will be a random collection of separate developers and not a truly cohesive, interactive, and supportive development team. This means that, to achieve high performance, development teams must be supported by customized development processes.

The situation is much more hopeful for the other 13 activities typically performed by high-performing teams. All of these tasks can be performed in a relatively standard way, even over a wide range of team sizes and types. For example, standard operational scripts have been defined to cover all of these 13 activities, and they have been successfully used by both large and small teams developing software, hardware, and even entire systems. The identical process scripts have also been used by process-improvement teams in managing their work as well as by one management team in running a corporation [Humphrey 2002, Humphrey 2006]. This means that, with modest preparation and guidance, most teams can be guided through the steps of establishing their own customized strategies and defining their own customized operational processes.

References

[Humphrey 2002] Watts S. Humphrey, *Winning with Software*. Boston: Addison Wesley, 2002.

[Humphrey 2006] Watts S. Humphrey, *TSP: Coaching Development Teams*. Boston: Addison Wesley, 2006.