

Software Engineering Project Management

Software engineers may have the right technical skills for your project, but have they got the right personality?



The major cost item for software development projects is people. Thus a key issue for any project manager is staff selection. Software engineering productivity studies show substantial differences between individual developers. Getting the right people on your project team greatly improves the chances of success. Knowing the qualifications, technical skills, and experience of team members is all well and good, but equally important is to understand the different working personalities of software developers.

People work in different ways dictated by their personalities. They develop individual problem-solving habits and ways of thinking about and understanding the requirements of a task. Different types of problems appeal to different mentalities. The secret of productivity is to match the requirements of a particular project with the personality of key team members. Just as there is no “silver bullet” methodology that can be applied to every project, in the same way there are no all-purpose system developers. The selection of project staff must be tailored to the project in the same way methods, tools, and techniques are tailored.

The general rhythm of work for people can be described as “soldiering”—working within oneself most of the time with occasional spurts of activity. A problem for project managers is to allocate people to roles and tasks that minimize the time staff spend working within themselves and maximize spurts of activity. The best way to do this is to assess team members’ development personalities and make the best use of them within the constraints of the project.

Some software engineers are good at producing systems rapidly with the first attempt. They are

capable of exploding into periods of highly focused activity, often between periods of relative inactivity. Other developers take a three-stage approach to their work. They like to plan, build, and revise. They make no attempt to perfect the system on the first draft. They prefer to see something working which can then be revised and improved over time. A third type of software developer likes to polish and perfect each system module before attempting the next. They build systems module by module, piece by piece, until the entire edifice is completed. Their focus is always on the technical excellence of their work rather than its usefulness to others. We can categorize these different work personalities as *deliverers*, *prototypers*, and *perfectors*.

Deliverers can be used effectively in short bursts on rapid application development projects. They also make good emergency software maintenance staff. Prototypers are best suited to projects where system requirements are initially unclear and where ensuring that building the right system is more important than producing something quickly. Perfectors work best on safety-critical applications where meticulous attention to detail is essential. It is futile to try to force perfectors to produce systems quickly.

THE HUGE GROWTH IN SOFTWARE DEVELOPMENT over the past 30 years and the need to regularly maintain systems has given rise to two other useful categories of software engineers: *producers* and *fixers*.

Producers are those software engineers who can produce goods in terms of programs and systems that work as long as not too many questions are asked or too much interference takes place from the project manager. They prefer to work in their own way, using techniques they have applied to advan-

tage for many years. The imposition of new methods, standards, and techniques disrupts their job satisfaction and productivity. Producers can be crucial to software development projects. On the minus side, they are often allowed to ride roughshod over documentation and development standards in order to get the job done. Companies may have to pay for this behavior at a later date when software maintenance is needed.

Fixers, on the other hand, are software engineers who have worked on particular systems for many

Just as there is no “silver bullet” methodology that can be applied to every project, in the same way there are no all-purpose system developers.

years. They know the internals of their systems so well they can envisage the system amendments needed to meet new user requirements very quickly. Fixers can make changes in hours or days that would take developers who are new to the system weeks or months. The importance of fixers to software maintenance projects rarely gets the credit it is due.

The challenge for system developers is to get the right system right, at the right time. In order to do this you need a good balance in software projects between *convergers* and *divergers*. Convergers are extremely good at analysis and at following established procedures. They tend to be drawn from scientific and engineering backgrounds. Convergers' training has taught them to focus on problems. Convergers are good at applying tools and techniques to problem-solving. If they come across a problem where no methods or procedures seem to apply they try to redefine the problem and force-fit it into a form amenable to the application of their techniques. This can sometimes lead to inappropriate and over-complex technical solutions.

Divergers, on the other hand, are good at creative thinking and seeing new patterns in situations. Their backgrounds may be less technical than that of convergers. Unlike convergers, divergers are less happy

following set rules and procedures, preferring a more unstructured approach to problem analysis. Convergers tend to quickly classify problems. Divergers leave issues and problems to germinate. Convergers have an anxiety to, “get the solution under way.” Divergers prefer to spend more time formulating the problem and exploring creative solutions. Careful analysis of the requirements of their particular project helps project managers select the right balance between convergers and divergers for their needs.

A major problem for software development projects has always been meeting deadlines. For any number of good or bad reasons, projects fail to complete on time. New tools and techniques have had only limited success in tackling this issue. A much better way for project managers to ensure a project completes on time is to staff it with *finishers*.

Finishers see only one imperative in a task and that is to finish it. They are single-mindedly task-oriented. They use whatever rules, procedures, and behavior necessary to finish. Their personal pride and job satisfaction is closely linked to meeting their commitment to deliver on time. Non-finishers tend to put things off and look for excuses. They rigorously follow rules and regulations so that these can be used as an excuse for not completing on time. They exhibit behavior that gives the illusion of progress where none actually exists.

CLEARLY, INDIVIDUALS HAVE COMPLEX PERSONALITIES and many software engineers exhibit a selection of different traits and attitudes. However, understanding developer personality characteristics can help managers put together the right team. Productivity and job satisfaction can be improved. Staff members are more likely to see the project to completion. If you get the right people on the project, you are more likely to achieve the right outcome. ■

ALAN HOWARD (a.howard@lmu.ac.uk) is a senior lecturer in Information Systems Development at Leeds Metropolitan University, Leeds, England.

© 2001 ACM 0002-0782/01/0500 \$5.00