3.11 STAFFING

Software project managers usually take the responsibility of choosing their team. Therefore, they need to identify good software developers for the success of the project. A common mis-

conception held by managers as evidenced in their staffing, planning and scheduling practices, is the assumption that one software engineer is as productive as another. However, experiments have revealed that there exists a large variability of productivity between the worst and the best software developers in a scale of 1 to 30. In fact, the worst developers may sometimes even reduce the overall productivity of the team, and thus in effect exhibit negative productivity. Therefore, choosing good software developers is crucial to the success of a project.

3.11.1 Who is a Good Software Engineer?

In the past, several studies concerning the traits of a good software engineer have been carried out. All these studies roughly agree on the following attributes that good software developers should possess:

1. Exposure to systematic techniques, i.e. familiarity with software engineering principles
2. Good technical knowledge of the project areas (*domain knowledge*)
3. Good programming abilities
4. Good communication skills like oral, written and interpersonal skills
5. High motivation
6. Sound knowledge of fundamentals of computer science
7. Intelligence
8. Ability to work in a team
9. Discipline

Studies show that these attributes vary as much as 1:30 for poor and bright candidates. An experiment conducted by Sackman [1968] shows that the ratio of coding hour for the worst to the best programmers is 25:1, and the ratio of debugging hours is 28:1. Also, the ability of a software engineer to arrive at the design of the software from a problem description varies greatly with respect to the parameters of quality and time.

Technical knowledge in the area of the project (*domain knowledge*) is an important factor determining the productivity of an individual for a particular project, and the quality of the product that he develops. A programmer having a thorough knowledge of database applications (e.g. MIS) may turn out to be a poor data communication developer. Lack of familiarity with the application areas can result in low productivity and poor quality of the product.

Since software development is a group activity, it is vital for a software developer to possess three main kinds of communication skills: oral, written and interpersonal. A software developer not only needs to effectively communicate with his teammates (e.g. reviews, walk throughs, and other team communications) but may also have to communicate with the customer to gather product requirements. Poor interpersonal skills hamper these vital activities and often show up as poor quality of the product and low productivity. Software developers are also required at times to make presentations to the managers and to the customers. This requires a different kind of communication skill (oral communication skill). A software developer is also expected to document his work (design, code, test, etc.) as well as write the users' manual, training manual, installation manual, maintenance manual, etc. This requires good *written communication* skill.

Motivation level of software developers is another crucial factor contributing to his work quality and productivity. Even though no systematic studies have been reported in this regard,

it is generally agreed that even bright developers may turn out to be poor performers when they lack motivation. An average developer who can work with a single mind track can outperform other developers. But motivation is a complex phenomenon requiring careful control. For majority of software developers, higher incentives and better working conditions have only limited affect on their motivation levels. Motivation is to a great extent determined by personal traits, family and social backgrounds, etc.

3.12 RISK MANAGEMENT