

TeamWork

The people working in a software organization are its greatest assets. In successful companies and economies, the best possible return on its investment is achieved when people are respected by the organization and are assigned responsibilities that reflect their skills and experience.

Four critical factors in people management:

1. Consistency - People in a project team should all be treated in a comparable way. No one expects all rewards to be identical but people should not feel that their contribution to the organization is undervalued.
2. Respect - Different people have different skills and managers should respect these differences. All members of the team should be given an opportunity to make a contribution. In some cases, of course, you will find that people simply don't fit into a team and they cannot continue, but it is important not to jump to conclusions about this at an early stage in the project.
3. Inclusion - People contribute effectively when they feel that others listen to them and take account of their proposals. It is important to develop a working environment where all views, even those of the most junior staff, are considered.
4. Honesty - As a manager, you should always be honest about what is going well and what is going badly in the team. You should also be honest about your level of technical knowledge and willing to defer to staff with more knowledge when necessary. If you try to cover up ignorance or problems you will eventually be found out and will lose the respect of the group.

MOTIVATION

As a project manager, you need to motivate the people that work with you so that they contribute to the best of their abilities. Motivation means organizing the work and the working environment to encourage people to work as effectively as possible. If people are not motivated, they will not be interested in the work they are doing. They will work slowly, be more likely to make mistakes, and will not contribute to the broader goals of the team or the organization. There is need to satisfied people's higher-level needs.



1. To satisfy **social** needs, you need to give people time to meet their co-workers and provide places for them to meet. This is relatively easy when all of the members of a development team work in the same place but, increasingly, team members are not located in the same building or even the same town or state. They may work for different organizations or from home most of the time. Social networking systems and teleconferencing can be used to facilitate communications but my experience with electronic systems is that they are most effective once people know each other. You therefore need to arrange some face-to-face meetings early in the project so that people can directly interact with other members of the team. Through this direct interaction, people become part of a social group and accept the goals and priorities of that group.

2. To satisfy **esteem** needs, you need to show people that they are valued by the organization. Public recognition of achievements is a simple yet effective way of doing this. Obviously, people must also feel that they are paid at a level that reflects their skills and experience.

3. Finally, to satisfy **self-realization** needs, you need to give people responsibility for their work, assign them demanding (but not impossible) tasks, and provide a training programme where people can develop their skills. Training is an important motivating influence as people like to gain new knowledge and learn new skills.

People can be also motivated through helping a group achieve shared goals.

Being a member of a **cohesive** group is highly motivating for most people. People with fulfilling jobs often like to go to work because they are motivated by the people they work with and the work that they do. Therefore, as well as thinking about individual motivation you also have to think about how a group as a whole can be motivated to achieve the organization's goals.

Personality type also influences motivation. Bass and Dunteman (1963) classify professionals into three types:

1. Task-oriented people, who are motivated by the work they do. In software engineering, these are people who are motivated by the intellectual challenge of software development.

2. Self-oriented people, who are principally motivated by personal success and recognition. They are interested in software development as a means of achieving their own goals. This does not mean that these people are selfish and think only of their own concerns. Rather, they often have longer-

term goals, such as career progression, that motivate them and they wish to be successful in their work to help realize these goals.

3. Interaction-oriented people, who are motivated by the presence and actions of co-workers. As software development becomes more user-centered, interaction oriented individuals are becoming more involved in software engineering.

Interaction-oriented personalities usually like to work as part of a group, whereas task-oriented and self-oriented people usually prefer to act as individuals. Women are more likely to be interaction-oriented than men. They are often more effective communicators.

TeamWork – working in a group

Putting together a group that has the right balance of technical skills, experience, and personalities is a critical management task. However, successful groups are more than simply a collection of individuals with the right balance of skills. A good group is cohesive and has a team spirit. The people involved are motivated by the success of the group as well as by their own personal goals. In a cohesive group, members think of the group as more important than the individuals who are group members. Members of a well-led, cohesive group are loyal to the group. They identify with group goals and other group members. They attempt to protect the group, as an entity, from outside interference. This makes the group robust and able to cope with problems and unexpected situations.

The benefits of creating a cohesive group are:

1. *The group can establish its own quality standards* - Because these standards are established by consensus, they are more likely to be observed than external standards imposed on the group.
2. *Individuals learn from and support each other* - People in the group learn from each other. Inhibitions caused by ignorance are minimized as mutual learning is encouraged.
3. *Knowledge is shared* - Continuity can be maintained if a group member leaves. Others in the group can take over critical tasks and ensure that the project is not unduly disrupted.
4. *Refactoring and continual improvement is encouraged* - Group members work collectively to deliver high-quality results and fix problems, irrespective of the individuals who originally created the design or program.

Whether or not a group is effective depends, to some extent, on the nature of the project and the organization doing the work. If an organization is in a state of turmoil with constant reorganizations and job insecurity, it is very difficult for team members to focus on software development. However, apart from project and organizational issues, there are three generic factors that affect team working:

1. **The people in the group** You need a mix of people in a project group as software development involves diverse activities such as negotiating with clients, programming, testing, and documentation.
2. **The group organization** A group should be organized so that individuals can contribute to the best of their abilities and tasks can be completed as expected.

3. Technical and managerial communications Good communications between group members, and between the software engineering team and other project stakeholders, is essential.

As with all management issues, getting the right team cannot guarantee project success. Too many other things can go wrong, including changes to the business and the business environment. However, if you don't pay attention to group composition, organization, and communications, you increase the likelihood that your project will run into difficulties.

It is sometimes **impossible to choose a group with complementary personalities**. If this is the case, the project manager has to control the group so that individual goals do not take precedence over organizational and group objectives. This control is easier to achieve if all group members participate in each stage of the project. Individual initiative is most likely when group members are given instructions without being aware of the part that their task plays in the overall project. For example, say a software engineer is given a program design for coding and notices what appears to be possible improvements that could be made to the design. If he or she implements these improvements without understanding the rationale for the original design, any changes, though well intentioned, might have adverse implications for other parts of the system. If all the members of the group are involved in the design from the start, they will understand why design decisions have been made. They may then identify with these decisions rather than oppose them.

ORGANIZATION OF A GROUP

Small programming groups are usually organized in a fairly informal way. The group leader gets involved in the software development with the other group members. In an informal group, the work to be carried out is discussed by the group as a whole, and tasks are allocated according to ability and experience. More senior group members may be responsible for the architectural design. However, detailed design and implementation is the responsibility of the team member who is allocated to a particular task.

Extreme programming groups (Beck, 2000) are always informal groups. XP enthusiasts claim that formal structure inhibits information exchange. In XP, many decisions that are usually seen as management decisions (such as decisions on schedule) are devolved to group members. Programmers work together in pairs to develop code and take joint responsibility for the programs that are developed.

Informal groups can be very successful, particularly when most group members are experienced and competent. Such a group makes decisions by consensus, which improves cohesiveness and performance. However, if a group is composed mostly of inexperienced or incompetent members, informality can be a hindrance because no definite authority exists to direct the work, causing a lack of coordination between group members and, possibly, eventual project failure.

Hierarchical groups are groups that have a hierarchical structure with the group leader at the top of the hierarchy. He or she has more formal authority than the group members and so can direct their work. There is a clear organizational structure and decisions are made towards the top of the hierarchy and implemented by people lower down the hierarchy. Communications are primarily instructions from senior staff and there is relatively little 'upward' communication from the lower levels to the upper levels in the hierarchy. This approach can work well when a well-understood problem can be easily broken into subproblems with subproblem solutions developed in different

parts of the hierarchy. In those situations, relatively little communication across the hierarchy is required.

However, such situations are relatively rare in software engineering for the following reasons:

1. Changes to the software often require changes to several parts of the system and this requires discussion and negotiation at all levels in the hierarchy.
2. Software technologies change so fast that more junior staff often know more about the technology than experienced staff. Top-down communications may mean that the project manager does not find out about the opportunities of using new technologies. More junior staff may become frustrated because of what they see as old-fashioned technologies being used for development.

COMMUNICATION

It is absolutely essential that group members communicate effectively and efficiently with each other and with other project stakeholders. Group members must exchange information on the status of their work, the design decisions that have been made, and changes to previous design decisions. They have to resolve problems that arise with other stakeholders and inform these stakeholders of changes to the system, the group, and delivery plans. Good communication also helps strengthen group cohesiveness. Group members come to understand the motivations, strengths, and weaknesses of other people in the group.

The effectiveness and efficiency of **communications is influenced by:**

1. Group size - As a group gets bigger, it gets harder for members to communicate effectively. Status differences between group members mean that communications are often one-way. Managers and experienced engineers tend to dominate communications with less experienced staff, who may be reluctant to start a conversation or make critical remarks.
2. Group structure - People in informally structured groups communicate more effectively than people in groups with a formal, hierarchical structure. In hierarchical groups, communications tend to flow up and down the hierarchy. People at the same level may not talk to each other. This is a particular problem in a large project with several development groups. If people working on different subsystems only communicate through their managers, then there are more likely to be delays and misunderstandings.
3. Group composition - People with the same personality types may clash and, as a result, communications can be inhibited. Communication is also usually better in mixed-sex groups (Marshall and Heslin, 1975) than in single-sex groups. Women are often more interaction oriented than men and may act as interaction controllers and facilitators for the group.
4. The physical work environment - The organization of the workplace is a major factor in facilitating or inhibiting communications.
5. The available communication channels - There are many different forms of communication—face-to-face, e-mail messages, formal documents, telephone, and Web 2.0 technologies such as social networking and wikis. As project teams become increasingly distributed, with team members working remotely, you need to make use of a range of technologies to facilitate communications.