Chapter 12 Working in teams

Further exercise pointers

1. To what extent is the Belbin approach to balanced teams compatible with having chief programmer teams?

Once again, the general aim is to provoke discussion. A general question is whether the Belbin model is appropriate to situations with fairly rigid lines of responsibility. There might be some limited application of the principles, for example, avoiding having two ‘shapers’ as the chief programmer and co-pilot.

Here is one example where someone has tried to suggest Belbin roles for each of the main roles in the chief programmer model.

Chief programmer team roles	*Belbin roles*
Chief programmer	Shaper Plant
Co-pilot	Monitor-evaluator Team worker
Program clerk	Completer-finisher Implementer
Editor	Completer-finisher Implementer
Tester	Completer-finisher Monitor-evaluator

2. If you have been involved recently in a group activity or project, try and categorize each participant according to the Belbin classification. Were there any duplications or gaps in any of the roles? Did this seem to have any impact on progress?

3. Three different mental obstacles to good decision making were identified in the text: faulty heuristics, escalation of commitment and information overload. What steps do you think can be taken to reduce the danger of each of these?

Faulty heuristics – recall that these refer to ‘rules of thumb’ that involve interpretations of the information at hand, which runs the risk of being incomplete.
One approach to, perhaps, reducing the problems with this approach would include:

- Identifying the assumptions upon which a decision is based: you could use the causal mapping technique described in Chapter 7
- Looking for evidence that confirms or refutes those assumptions

Escalation of commitment – a common symptom of this is where an organization perseveres with a project for which there is unlikely to be an economic justification because of runaway costs.

One action to reduce this is to separate the project management and project evaluation functions with the latter role being carried out by people who have no emotional investment in the project.
A general approach might be:

- Identify objective criteria which, for example, allow the current state of the business case for the project to be assessed
- Have an independent assessment of these criteria at regular intervals during the course of the project.

Information overload

- Identify key questions
- Focus on information that help answer those questions
- Ask for verbal explanations of the information provided
- Make information requirements and needs clear to those supplying the information
- Use of summary techniques such as graphs and management by exception
-

4. In Exercise 12.5, the management style most appropriate for each of three different situations was asked for. Go back and consider how you as a manager would respond to each of these three situations in terms of practical things to do or avoid.

The payroll officer

The new payroll officer would have a thorough knowledge of payroll processing, so their manager would not need to be task-oriented in that respect. However, the immediate job would be the establishment of a payroll office in what, for the payroll officer, would be a novel environment. This would require a more task oriented approach from the manager. The project manager and the payroll office might need, for example, to make quite detailed arrangements for the recruitment and training of staff. As the payroll officer would be new to the college, some people-oriented management: for example, introductions to co-workers, providing opportunities to find out about the roles, responsibilities and concerns of colleagues, and advising on local procedures.

The trainee analyst-programmer

If the trainee has had no previous relevant work experience, the need for task-oriented management would be acute.

The experienced maintenance programmer whose application is to be revised

A modest increase in task-oriented management has been suggested to deal with changes to one of the systems with which he/she is associated. One point here is whether it would be better to recruit the veteran as one of the people to drive the change through rather than be a rather passive recipient of change.

5 *Do you agree with the following statement? ``Few, if any, organization in the real world is purely functional, project, or matrix in nature." Justify your answer.*

Each type of staff organization has its own advantages and disadvantages. Depending on the problem at hand, organizations often tailor these organizations and also use a mixture of these.

6 *Explain the advantages of a functional organization over a project organization. Also explain why software development houses are preferring to use project organization over* *functional organization.*

The functional format has many advantages. The main advantages of a functional organization can be summarized as:

- Ease of staffing
- Production of good quality documents
- Job specialization
- Efficient handling of the problems associated with manpower turnover

In spite of several important advantages of the functional organization, it is not very popular in the software industry. The project format provides job rotation to the team members. That is, each team member takes on the role of the designer, coder, tester, etc during the course of the project. On the other hand, considering the present skill shortage, it would be very difficult for the functional organizations to fill slots for some roles such as the maintenance, testing, and coding groups.

Another problem with the functional organization is that if an organization handles projects requiring knowledge of specialized domain areas, then these domain experts cannot be brought in and out of the project for the different phases, unless the company handles a large number of such projects.

For obvious reasons the functional format is not suitable for small organizations handling just one or two projects.

7 *As a project manager, identify the characteristics that you would look for in a software developer while trying to select personnel for your team.*

- Exposure to systematic techniques, i.e. familiarity with software engineering principles.
- Good technical knowledge of the project areas
- Good programming abilities
- Good communication skills. These skills comprise of oral, written, and interpersonal skills.
- High motivation
- Sound knowledge of fundamentals of computer science
- Intelligence
- Ability to work in a team
- Discipline