

Software Engineering

SOFTWARE PROJECT PLANNING - III

PERSONNEL PLANNING

Gagan Deep

Founder & Director

Rozy Computech Services

Kurukshetra, rozygag@yahoo.com

www.rozyp.com

PERSONNEL PLANNING

- ⦿ Personnel planning basically deals with staffing. Therefore before going to personnel planning, first we discuss Staffing.
- ⦿ Staffing deals with the hiring personnel for the position that are identified by the organizational structure.
- ⦿ It involves
 - ⦿ defining requirements for personnel;
 - ⦿ recruiting (identifying, interviewing and selecting candidates);
 - ⦿ compensating,
 - ⦿ developing and promoting employees.

- ⦿ For personnel planning and scheduling, it is useful to have effort and schedule estimates for the subsystems and basic modules in the system.
- ⦿ At planning time, when the system design has not been done, the planner can only expect to know about the major subsystems in the system and perhaps the major modules in these subsystems.
- ⦿ Once the project schedule is estimated and the effort and schedule of different phases and tasks are known, staff requirements can be obtained.
- ⦿ From the cost and overall duration of the project, the *average* staff size for the project can be determined by dividing the total effort (in person-months) by the overall project duration (in months).

- ◎ Typically the staff requirement for a project is small during requirement and design, the maximum during implementation and testing, and drops again during the final phases of integration and testing.
- ◎ Using the COCOMO model, average staff requirement for the different phases can be determined as the effort and schedule for each phase are known.
- ◎ When the schedule and average staff level for each activity is known the overall personnel allocation for the project can be planned.
- ◎ This plan will specify how many people will be needed for the different activities at different times for the duration of the project.

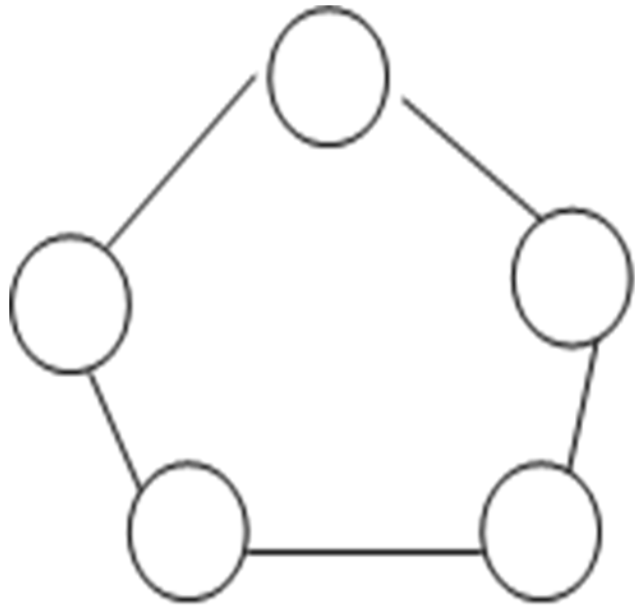
- ⦿ A method of producing the personnel plan is to make it a calendar-based representation, containing all the months in the duration of the project, by listing the months from the starting date to the ending date.
- ⦿ The total effort for each month and the total effort for each activity can easily be computed from this plan.
- ⦿ Drawing a personnel plan usually requires a few iterations to ensure that the effort requirement for the different phases and activities (and the duration of the different phases) is consistent with the estimates obtained earlier.
- ⦿ This type of plan, although it has the overall staff requirement, does not distinguish between different types of people.
- ⦿ A more detailed plan will list the requirement of people by their specialty; for example, stating how many programmers, analysts, quality assurance people, and so forth are needed at different times.

TEAM STRUCTURES

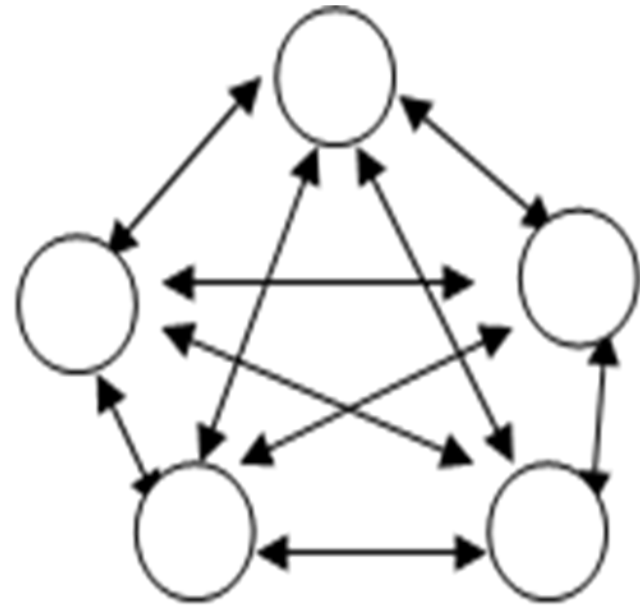
Ego-less or Democratic Teams

- ⦿ Ego-less teams consist of team of fewer programmers. The goals of the group are set by consensus, and input from every member is taken for major decisions. Group leadership rotates among the group members. Due to its nature, egoless teams are sometimes called democratic teams. The structure results in many communication paths between people, as shown in Figure
- ⦿ The structure allows input from all members, which can lead to better decisions in difficult problems. This suggests that this structure is well suited for long-term research-type projects that do not have time constraints.

Ego-less Programming Team structure and Communication paths



(a) Structure



(b) Communication path

Advantages

- ⦿ The opportunity for each team member to contribute to decisions, the opportunity for team members to learn from one another.
- ⦿ The increased job satisfaction that accrues from good communication in an open, non-threatening work environment.
- ⦿ Democratic team is well suited to difficult, long term research and development projects.

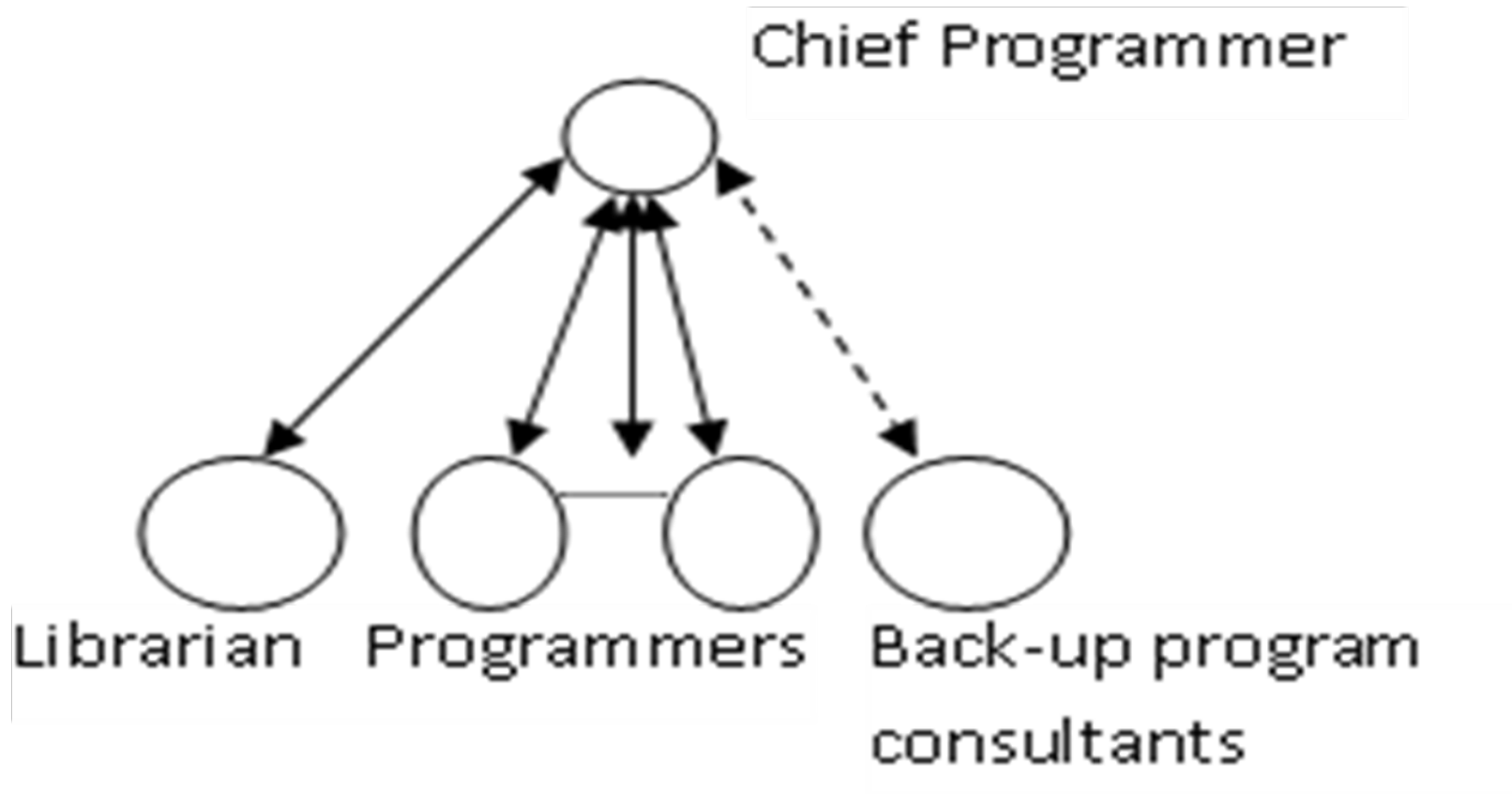
Disadvantage

- ⦿ All members have same status that's why there is problem in making decision.
- ⦿ Lack of individual responsibility & authority, that results in less initiative and less personal attention

Chief Programmer Team

- ⦿ A chief programmer team, in contrast to ego-less team, has a hierarchy.
- ⦿ It consists of a chief programmer, who has a backup programmer, a program librarian, and some programmers.
- ⦿ The chief programmer is responsible for all major technical decisions of the project.
- ⦿ He does most of the design and he assigns coding of the different parts of the design to the programmers.
- ⦿ The backup programmer helps the chief programmer make technical decisions, and takes over as the chief programmer if the chief programmer falls sick or leaves.
- ⦿ The program librarian is responsible for maintaining the documentation and other communication-related work.
- ⦿ This structure considerably reduces interpersonal communication. The communication paths are shown in Figure.

Chief programmer team Structure & Communication paths



Advantages

- ⦿ It has advantages of centralized decision making and reduced communication paths; however, the effectiveness of a chief programmer team is quite sensitive to the chief programmer's technical and managerial abilities.
- ⦿ Two situations, where chief programmer teams are effective are, first, in data processing applications where the chief programmer has responsibility for sensitive financial software packages and the packages can be written by relatively unskilled programmers; and
- ⦿ second, in situations where one senior programmer and several junior programmers are assigned to a project.

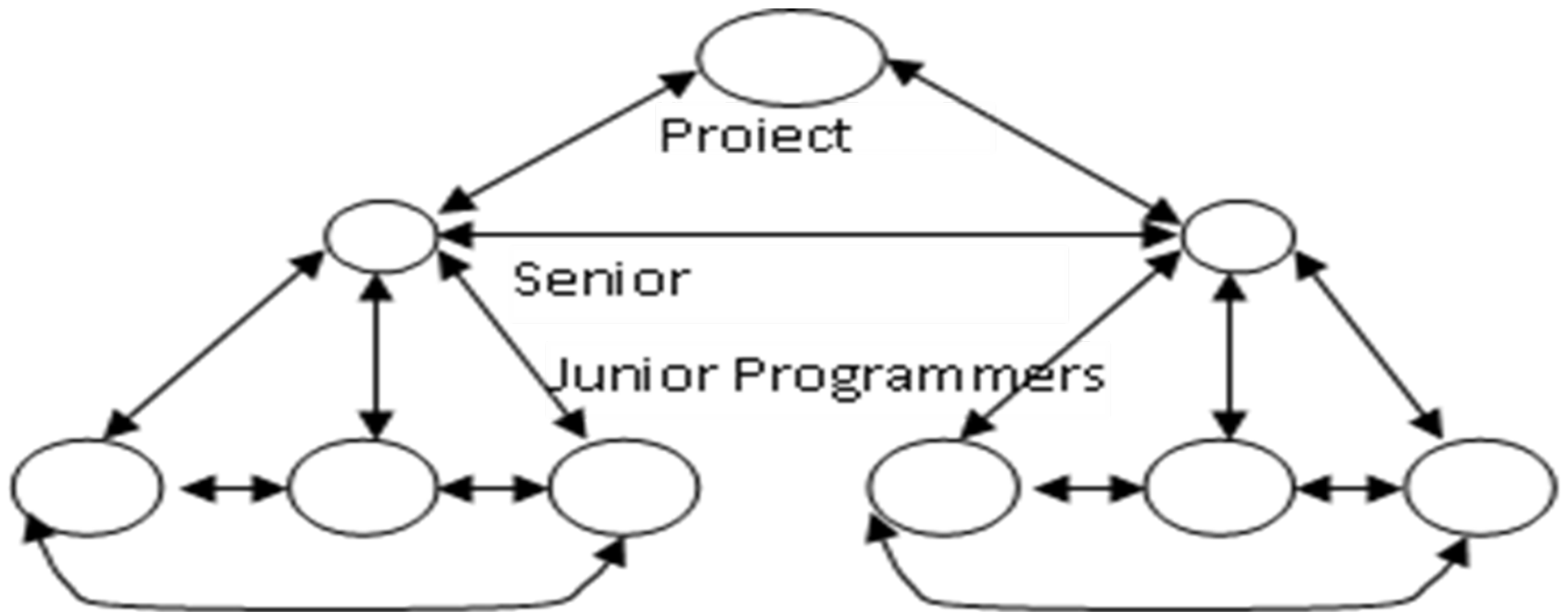
Disadvantages

- ⦿ The chief programmer structure can also result in low moral among the subordinate programmers.
- ⦿ In the long absence of chief programmer there is a hurdle between the system work, because the back-up programmer is not fully responsible.
- ⦿ Work will be slow down in the absence any team member, because every member has certain responsibilities.

Controlled Decentralized Team (Hierarchical Team Structure)

- ⦿ A third team structure, called the controlled decentralized team, tries to combine the strengths of the democratic and chief programmer teams.
- ⦿ It consists of a project leader who has a group of senior programmers under him, while under each senior programmer is a group of junior programmers.
- ⦿ The group of a senior programmer and his junior programmers behaves like an ego-less team, but communication among different groups occurs only through the senior programmers of the groups.
- ⦿ The senior programmers also communicate with the project leader.

- Such a team has fewer communication paths than a democratic team but more paths compared to a chief programmer team.
- This structure works best for large projects that are reasonably straightforward. It is not well suited for very simple projects or research-type projects.



Advantages

- ⦿ The opportunity for each team member to contribute to decisions, the opportunity for team members to learn from one another.
- ⦿ The increased job satisfaction that occurs from good communication in an open, non-threatening work environment.
- ⦿ Democratic team is well suited to difficult, long-term research and development projects.

Disadvantages

- ⦿ Since many persons involved to implement its decision, the final decision is difficult to be taken.
- ⦿ Overhead expenses are increased.
- ⦿ Since there is a democratic team, it can cause delay resulting in financial overburden and project can be delayed beyond its time.
- ⦿ One person could not take decision, hence he has to wait for the input of all members.

Thanks!

Gagan Deep

rozygag@yahoo.com

www.rozyph.com