

SOFTWARE PROJECT MANAGEMENT

Lecture # 35



TEAM LEADER

➤ The MOI Model

- **Motivation.** The ability to encourage (by “push or pull”) technical people to produce to their best ability.
- **Organization.** The ability to mold existing processes (or invent new ones) that will enable the initial concept to be translated into a final product.
- **Ideas or innovation.** The ability to encourage people to create and feel creative even when they must work within bounds established for a particular software product or application.



TEAM LEADER

- Another view of the characteristics that define an effective project manager emphasizes four key traits:
 - **Problem solving.** An effective software project manager can diagnose the technical and organizational issues that are most relevant, systematically structure a solution or properly motivate other practitioners to develop the solution, apply lessons learned from past projects to new situations, and remain flexible enough to change direction if initial attempts at problem solution are fruitless.
 - **Managerial identity.** A good project manager must take charge of the project. He must have the confidence to assume control when necessary and the assurance to allow good technical people to follow their instincts.

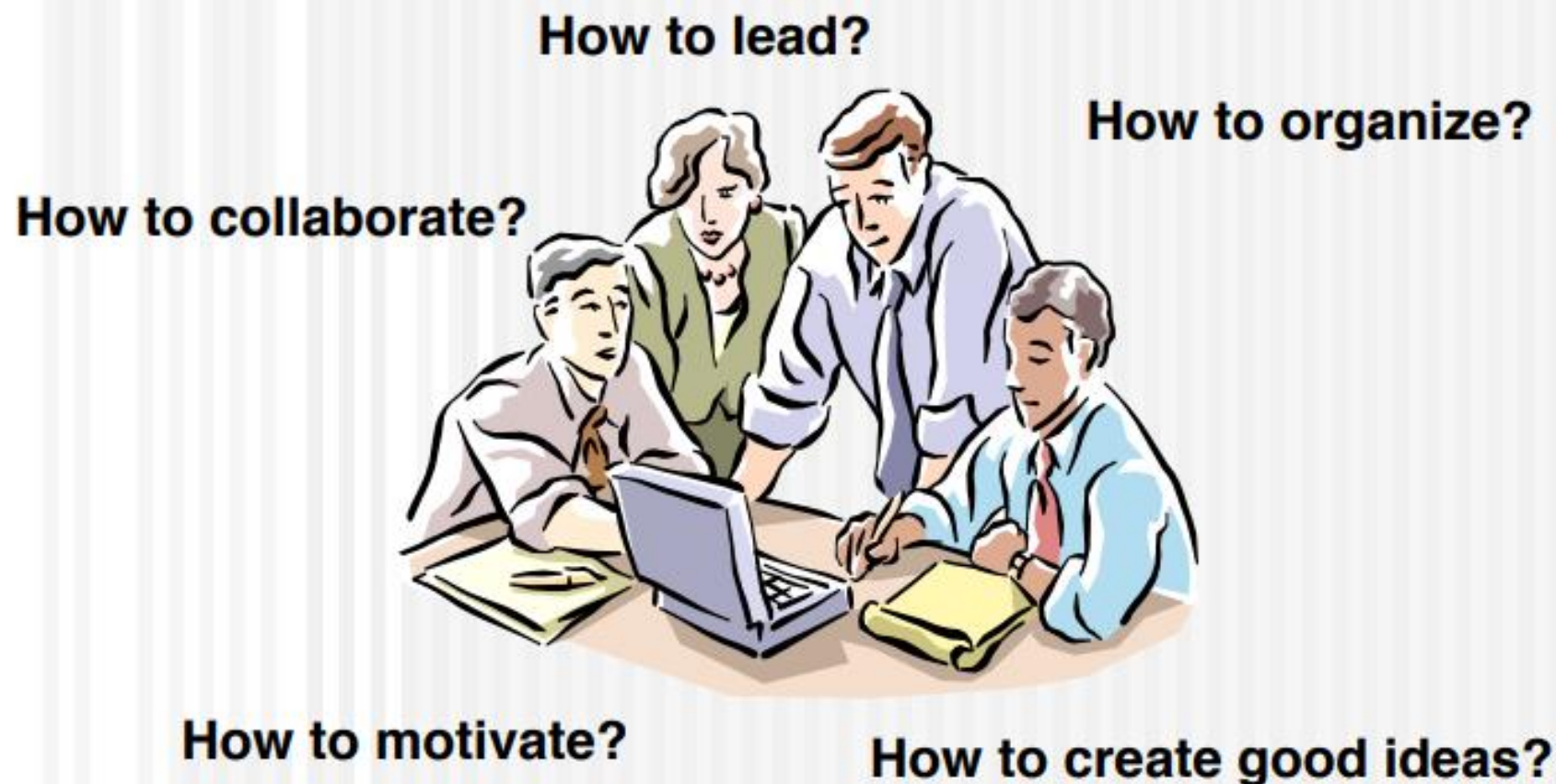


TEAM LEADER

- **Achievement.** A competent manager must reward initiative and accomplishment to optimize the productivity of a project team. He must demonstrate through his own actions that controlled risk taking will not be punished.
- **Influence and team building.** An effective project manager must be able to “read” people; He must be able to understand verbal and nonverbal signals and react to the needs of the people sending these signals. The manager must remain under control in high-stress situations.



SOFTWARE TEAMS



SOFTWARE TEAMS

- The following factors must be considered when selecting a software project team structure ...
 - the difficulty of the problem to be solved
 - the size of the resultant program(s) in lines of code or function points
 - the time that the team will stay together (team lifetime)
 - the degree to which the problem can be modularized
 - the required quality and reliability of the system to be built
 - the rigidity of the delivery date
 - the degree of sociability (communication) required for the project



ORGANIZATIONAL PARADIGMS

- **Closed paradigm**—structures a team along a traditional hierarchy of authority
- **Random paradigm**—structures a team loosely and depends on individual initiative of the team members
- **Open paradigm**—attempts to structure a team in a manner that achieves some of the controls associated with the closed paradigm but also much of the innovation that occurs when using the random paradigm
- **Synchronous paradigm**—relies on the natural compartmentalization of a problem and organizes team members to work on pieces of the problem with little active communication among themselves



AVOID TEAM “TOXICITY”

- A frenzied work atmosphere in which team members waste energy and lose focus on the objectives of the work to be performed.
- High frustration caused by personal, business, or technological factors that cause friction among team members.
- “Fragmented or poorly coordinated procedures” or a poorly defined or improperly chosen process model that becomes a roadblock to accomplishment.
- Unclear definition of roles resulting in a lack of accountability and resultant finger-pointing.
- “Continuous and repeated exposure to failure” that leads to a loss of confidence and a lowering of morale



AGILE TEAMS

- Team members must have trust in one another.
- The distribution of skills must be appropriate to the problem.
- Mavericks may have to be excluded from the team, if team cohesiveness is to be maintained.
- Team is “self-organizing”
 - An adaptive team structure
 - Uses elements of Constantine’s random, open, and synchronous paradigms
 - Significant autonomy

