Software Engineering Teams

Managing Software Engineering Teams. Managing software engineers is more difficult; the following characteristics are more prevalent among software engineers and programmers:

- logical
- artistic
- possessive
- temperamental
- diverse productivity ratio (as much as 20:1)

Software projects organized as teams; each team responsible for a specific function

Project manager must form the teams

(For exclusive use of Professor Saiedian's students)
Team Work Goals

- Volunteers are better than draftees
- Creating and sustaining an environment that fosters cooperation; team leadership must inspire growth
- Share common goals; committed to achieving the goals
- Mutual respect and common code of conduct
- Sense of enjoyment; desire to do what is needed to succeed
- Sharing a common reward; equitable compensation; selective rewards only for extraordinary work
- Team spirit: team needs are more important than personal needs
- Recognition for team achievements
- Clear definition of team vision and objectives; consensus on goals and individual responsibilities - develop organization structure

(For exclusive use of Professor Saiedian’s students)
A Minimal Team Structure

- Project Manager
  - Project Secretary
  - Deputy Manager
  - Quality Assurance
  - System Engineer
  - Configuration Control
    - Development Team
    - Development Team
    - Development Team
    - Development Team
    - Development Team

(For exclusive use of Professor Saiedian's students)
Is it Really a Team?

Positive indicators:

- collaborative climate
- free flow of information
- positive interpersonal interaction
- high energy
- high morale; fun to come to work

Negative indicators:

- suspicious and un-trusting climate
- information is withheld
- finger-pointing prevails
- defensiveness and counterproductive cliques form
- no fun to work

(For exclusive use of Professor Saiedian’s students)
Team Organization

Chief Programmer Teams: team leader is both a technical mentor and the team’s coordinator; team coordination may take as much as 50% of team leader’s time to

- assign tasks to the team members
- provide advice and guidance
- supervise the work of team members
- coordinate activities

All team members report to the team leader; the leader must be very good at making decisions

(For exclusive use of Professor Saiedian's students)
Team Organization (continued)

Democratic Teams: no team leader per se; the team leader mostly in charge of coordination

- technical decisions made by the whole team
- ideal for experienced software developers; not so ideal for junior developers
- leads to ego-less programming: everyone is equally responsible
Developing High Performance Teams

- Small team size
- Result-driven structure
- Effective communication
- Mutual trust
- A sense of autonomy and empowerment

(For exclusive use of Professor Saiedian's students)
Physical Facilities are Important

- Work and conference rooms
- Lighting and physical comfort
- Minimal noise areas (private office vs cubicles)
- Access to hardware and equipment (e.g., printers, copy machine, fax, white boards, flip charts)
- Minimums of a productive environment (McConnell 1996)
  - 80 SF per developer; 15 SF of desk space; 15 SF of bookshelves space
  - Means of stopping phone interruption
  - Means of stopping in-persona interruption
  - Means of stopping unwanted noise
  - Convenient access to other team members

(For exclusive use of Professor Saiedian’s students)
Benefits of Workspace Improvement

A number of studies have shown that productivity will improve (as much as 11% - an IBM research) with well designed workspaces.
Effective Team and Project Meetings

Project status meetings should be held periodically and to be attended by key project members

- Well-defined agenda
- Participants must be prepared
- 2-Hour duration; Begin on time; End on time
- Document and distributes the outcome (minutes):
  - Date and name of the meeting
  - List of participants
  - Major decisions and items discussed
  - Action items and completion dates

(For exclusive use of Professor Saiedian’s students)
New Ideas

• Pair-Programming

“PairProgramming requires two engineers to participate in one development effort at one workstation. Each member performs the action the other is not currently doing: While one types in UnitTests the other thinks about the class that will satisfy the test, for example. Studies have shown that, after training for the ‘PeopleSkills’ involved two programmers are more than twice as productive as one for a given task”


(For exclusive use of Professor Saiedian's students)
• PairProgramming.com

http://www.c2.com/cgi/wiki?PairProgramming

(For exclusive use of Professor Saiedian's students)