People Management and Team Organization

CS A401

People Management

- Software development involves teamwork
- Members must coordinate work, decisions, etc.
- As the size of the teams increase, communication between team members increases exponentially
  - Large teams usually split into smaller teams to confine coordination and communication
Types of Team Organizations

- Hierarchical
- Business
- Matrix
- Chief Programmer
- SWAT
- Open Structured
- Search and Rescue Team
- Professional Athletic Team

Hierarchical Team

- Project is divided into major subsystems
- Each subsystem is headed by a project manager
  - If a subsystem is complex, it may be further divided into smaller components
- Most of the work done at the leaves of the tree
- Peter Principle
  - In a hierarchical organization each employee in general rises until reaching a level at which he is incompetent
  - Rewards management more for rising up the tree
**Business Team**

- Most common of the team structures
- Peer group headed by a technical lead
  - Aside from the leader, team members are all equal in status and differentiated by expertise
  - Technical lead is also an active technical contributor but also responsible for final decisions and communicates externally for the team
- Streamlines communication with management, allows team members to work in their area of expertise, allows the team itself to sort out who should work on what
- Strength and weakness is its generality

---

**Matrix**

- Basic units with specializations
- Individuals with their specialty assigned to projects requiring those talents
- Each feature team is responsible for developing one feature of the product
- Good for problem resolution teams

<table>
<thead>
<tr>
<th></th>
<th>Networking</th>
<th>Graphics</th>
<th>Databases</th>
<th>QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Project B</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Project C</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Chief Programmer

- The chief programmer is the team leader. This person must be a superstar and design/code much of the project.
- Other people just provide support services for the star.
  - They act as assistants in various capacities, documentation, administration, etc.
  - Similar to a surgeon in the OR.
- Makes good use of superstars; but hard to find anyone “super” enough.
- Can have conflict with the leader who makes all decisions.

SWAT

- Skilled With Advanced Tools.
- Based on the RAD methodology.
- Typically builds incremental versions of a software system.
  - Focuses on:
    - Reusable components.
    - Use of development tools.
    - Software generators.
    - Groupware and workflow management software.
  - Members must be experienced with multiple skills.
### Open or Weakly Structured

- Combine open management style with clear structure on team roles
- Technical leader
  - Responsible for resolving issues for which no consensus can be reached
- Moderator for meetings
- Rotate among members for various tasks
- Tries to combine a focus on individual specialists as well as centralized coordination of activities

### Skunk Works

- Take a bunch of talented people, hide them from the rest of the world, and see what they can do
- Based on the “black program” model
  - Covert development of sensitive systems
- May get extremely high creativity in exchange for low visibility and high risk
Search and Rescue

- Focuses on solving a particular problem
- Needs very specialized product knowledge
- Like a “tiger team” to attack a specific problem
- Consists of software and hardware specialists with detailed business environment knowledge
- Often used for emergency maintenance

Professional Athletic Team

- The developers are the stars of this team.
- Managers and everyone else are secondary.
- Roles are very specialized
- Good for tactical execution teams, which need specialized skills
- Lots of stars may not get along
General Team Principles

• Use fewer and better people
  – Highest productivity is achieved by a relatively small group of people
  – Large groups require exponentially more communication, which has a negative effect on productivity
  – Need to keep one person in charge of the project’s overall vision (chief programmer or architect)
• Try to fit tasks to the capabilities and motivation of the people available
• Select people for a well-balanced and harmonious team
• Someone who does not fit the team should be removed

Motivation

• Researchers have found a 10:1 range of performance among people with the same levels of experience
• Key to this is their motivation
• Problem is that motivation is hard to measure, quantify, or express
Motivation

• Types of motivating factors will vary from person to person
  – **Career growth potential**
  – Management potential
  – **Personal life**: is it possible to have one?
  – Status
  – Peer recognition
  – Personal achievement...

• More motivating factors:
  – Money! (or grades in our case)
  – Company ownership
  – Job security
  – Work environment
  – Home environment (travel, entertainment, …)

• No single thing motivates everyone
Top Five Motivational Factors

- Achievement (reaching goals)
- Possibility for Growth (personal improvement)
- The Work Itself
- Personal Life
- Technical Supervision Opportunity

Achievement

- Ownership: Get developers to own (set and agree to) the schedules they are working to achieve
- Goal Setting: Set specific goals for development (speed, quality, etc.)
  - Keep it to one or two specific goals
Possibility for Growth

- Support continuing education
- Conferences (attending, maybe presenting)
- Mentor new staff
- Avoid unrealistic schedule pressure
- Expect 2-3 weeks per year for training and education

The Work Itself

- Five main aspects
  - *Skill Variety*: to avoid boredom and fatigue
  - *Task Identity*: to be able to identify your product
  - *Task Significance*: to work on something important
  - *Autonomy*: to control how your task is done
  - *Job Feedback*: to see your work in action
Personal Life

- Traditionally, is a significant motivation for developers, not managers
  - Respect holidays and weekends as much as possible
  - Respect family and emergency needs
  - Avoid extensive travel

Tech. Supervision Opportunity

- Motivation for developers, not managers
  - Project Leads: Can assign leads for each technical area in the project: graphics, database design, etc.
  - Functional Leads: Or assign leads for each process area: system testing, technical reviews, performance evaluation, etc.
  - Assign mentor positions to all but most junior staff
Other Motivational Factors

- Rewards and Incentives - prizes, promotions, bonuses, …
- Pilot projects - adds importance to project
  – Try something new each project
- Performance reviews - need proper reviews to ensure good performance feedback

Motivation Killers

- Lack of hygiene - a clean office is required
  – Good lighting, office ergonomics, up to date equipment
- Management manipulation - don’t pretend your staff are dumb
- Excessive schedule pressure - have we emphasized this enough yet?
- Lack of appreciation - kudos are not optional...
More Motivation Killers

- Meddling by technically inept managers
- Leaving developers out of decisions
- Productivity barriers in the environment
- Low product quality - destroys sense of achievement
- Blatant motivation campaigns - be subtle

Teamwork

- The Amish barn raising is cited as a prime example of teamwork
  - Barn raised in a single day without electricity or power tools
  - Jobs for entire community
    - Kids carry food, tools
    - Men cooperate
    - No gripes or distractions
    - Nobody leaves until the barn is complete
- Model of the perfect team?
Teamwork’s Importance

• Major factors
• Effectiveness of teams can vary by a factor of 2-6 on the same project
  – Group cohesiveness is a key factor in team effectiveness
  – Developers’ experience is also significant

Characteristics of High Performance Teams

• A shared elevating vision or goal
  – A clear objective builds trust and focus, keeps perspective
• Challenging work
  – The project must be challenging to be worth the team’s focus
• A sense of team identity (who are we?)
  – Team name, dress code (IBM’s Black Team), etc...
Characteristics of High Performance Teams

• A results-driven structure
  – Must have clear roles, monitoring, feedback, and decision making methods based on fact
• Competent team members (not expert)
  – Select people based on technical skills, desire to contribute, and teamwork skills
• Commitment to the team
  – Sometimes merely asking is enough to get it...

Characteristics of High Performance Teams

• Mutual trust
  – Honesty, openness, consistency, respect
• Interdependence among team members
  – Share decisions, balance strengths
• Effective communication
  – Need to be able to communicate bad news, while it still can be fixed or accommodated...
Characteristics of High Performance Teams

- A sense of autonomy (independence)
  - In spite of a chance of making mistakes
- A sense of empowerment (authority)
  - Can say ‘no’ to a request, or break a little rule
- Small team size
  - 3 to 5 per team, max of 10
- A high level of enjoyment (fun!)

How to Manage a Team

- Establish a vision for the team
- Create change needed to bring about the vision
- Manage the team as a team, not as individuals
  - Make the team responsible rather than individuals for their individual actions
- Delegate tasks clearly to the team
- Let team work out the details
Team Failure

• If team isn’t working, examine the motivation, organization, and information
• Other possible causes include:
  – Lack of vision
  – Lack of identity
    • Members that would rather work alone, or lack of commitment
  – Lack of recognition
  – Productivity roadblocks
    • Environment too stifling?

Team Failure

• More causes:
  – Ineffective communication
  – Lack of trust
  – Problem personnel (don’t ignore them!)
    • Members that are belligerent, cover up ignorance rather than try to learn, are territorial, grumble, don’t pitch in
      – “My code is too complicated to test”
      – “No one else can fix the bugs in my code.”
    • In a review of 32 management teams, the most consistent and intense complaint from members was their team leaders were unwilling to confront and resolve problems with poor performance by team members
    • First approach is for the entire team to coach the problem person on how to work as part of a team
    • Second resort is to fire the person
Practical Guidelines for Team Leaders

• As a team leader I will
  – Avoid compromising the team’s objective with political issues
  – Exhibit personal commitment to the team’s goal
  – Not dilute the team’s efforts with too many priorities
  – Be fair and impartial toward all team members
  – Be willing to confront and resolve issues associated with inadequate performance by team members
  – Be open to new ideas and information from team members

Practical Guidelines for Team Members

• As a team member I will:
  – Demonstrate a realistic understanding of my role and accountabilities
  – Demonstrate objective and fact-based judgments
  – Collaborate effectively with other team members
  – Make the team goal a higher priority than any personal objective
  – Demonstrate a willingness to devote whatever effort is necessary to achieve team success
  – Be willing to share information, perceptions, and feedback appropriately
Practical Guidelines for Team Members

• As a team member I will:
  – Provide help to other team members when needed and appropriate
  – Demonstrate high standards of excellence
  – Stand behind and support team decisions
  – Demonstrate courage of conviction by directly confronting important issues
  – Demonstrate leadership in ways that contribute to the team’s success
  – Respond constructively to feedback from others