

COSC345

Software Engineering

Team Organization

Assignment

- Are you in a group?
- When is assignment-1 due?
- Do you have a schedule between now and then?
- When does your team meet?
- Do you have a github archive?

Outline

- Teams
 - Effective teams
 - Ineffective teams
 - Team composition
 - Task oriented
 - Talent oriented
 - Cohesiveness
 - Communications
 - Motivation
- Individuals
 - Skills to have
 - Getting staff

Teams

- A team is:
 - At least two people
 - Each with specific tasks
 - Working together for a common goal
 - And who have dependencies on each other
- A good team is:
 - Greater than the sum of its parts
- A bad team is:
 - Less than the weakest member
- Most significant software projects built by a team

Team Size

- Keep teams small (10 or fewer)
- Small teams often informal and emergent
 - The leader often emerges
 - Designs done by those with most skill
 - Tasks allocated by ability
 - Make decisions by consensus
- Large teams often appointed
 - Require more time in communication
 - Can be incohesive
 - Hard to manage

Effective Teams

- To be effective a team needs:
 - Direction
 - Agreed common goals
 - A supportive environment
 - Good communications
 - Mutual respect
 - Commitment
 - To the team
 - To the project
 - Individual roles

Effective Team Managers

- Ensure:
 - Realism
 - All deadlines must be realistic
 - Consistency
 - Everyone must feel valued for their contribution
 - Respect
 - Take into consideration people's strengths and weaknesses
 - All team members must be given the chance to contribute
 - Inclusion
 - People contribute when they feel heard
 - All people (junior and senior) must be comfortable speaking
 - Honesty
 - Tell the truth about what's going well and what's not
 - Defer to those of greater skill or knowledge

Ineffective Teams

- Bad leadership
 - Failure to compromise or cooperate
 - Lack of participation
 - Procrastination
 - Missed milestones
 - Late deliverables
 - Failed projects
-
- Bad management is one of the most significant contributors to project failure

Team Composition

- Teams must have complementary:
 - Personalities
 - Experience
 - Technical skills
- Leadership need not mean management
- No simple single answer
 - Task oriented
 - Team software process (Humphrey, 2000)
 - Talent oriented
 - The surgical team (Mills, 1971)
 - Mesh Management

Task Oriented

- Team leader
- Development manager
- Planning manager
- Quality / process manager
- Support manager

Talent Oriented

- The surgeon (chief programmer)
- Co-pilot
- Administrator
- Editor
- Secretaries
- Program clerk
- Tool-smith
- Tester
- Language expert

Team Cohesiveness

- Team is more important than the members
- If the team is cohesive:
 - Team standards will emerge
 - Members will learn from each other
 - Members will be familiar with each other's work
 - Programs will become team property
 - Team goals become individual goals
- But...
 - Preference of democracy over leadership
 - Resistant to changes in leadership
- Try to increase cohesiveness
 - Take ownership of project
 - Run team building events (socials)

Team Communication

- Vital for effective team work
- But...
 - As size increases communications takes longer
 - For complete communications
 - $n*(n-1)$ messages
 - Communications must become “partial”
 - Managers often act as buffers
- Look out for
 - Personality clashes
 - Workspace inhibitors
 - Senior member domination
- Flat teams work better than structured teams

Team Motivation

- Your staff have basic human needs
 - Social
 - Provide places to meet and time to communicate
 - Esteem
 - Show people that they are valued
 - Self realization
 - Give responsibility of tasks to individuals
 - Provide mechanisms for people to develop their skills
 - Mental stimulation
 - Don't make people do things they don't want to (for too long)
 - Eating, drinking, sleeping
 - You are even responsible for this

Getting Staff

- Often you won't have a choice
- Some skills are hard to find (and so cost more)
 - DBA, GUI designers
- The “best” person may not have the “best” skills
 - Staff must interact with
 - Other staff
 - Managers
 - Clients
 - Suppliers and third parties

Individual's Skills to Consider

- Application domain experience
- Platform experience
- Programming experience
- Problem solving experience
- Educational background
- Communication skills
- Adaptability
- Attitude
- Personality
- Nature of the task and the individual

Negative Productivity

- “The best (top 5% or so) are ten times better than average, not better than bad. So an average ‘best’ programmer is worth 10 average programmers (at least). An average ‘best’ programmer is worth 100 average ‘bad’ programmers (or infinitely more if the bad ones have negative productivity which is the usual case.)“ (Scott)
- “It has been widely demonstrated that productivity of software developers may vary radically from person to person, and on average about one in five programmers has a negative productivity, i.e. she [*sic*] slows down others’ work, reducing the quantity/quality of the final outcome” (Pianon 2004 citing Feller & Fitzgerald 2002).

Pre-Hire Knowledge Building

- Standard methods
 - CVs are factual
 - Interviews are personable
 - Recommendations are often unreliable
- Testing
 - Such as:
 - Programming tests
 - Psychometric profiling
 - Aptitude tests
 - Usefulness often debated
- It's easier to spot bad than to spot good

References

- F. Brooks, *Mythical Man Month*, Chapter 3
- W. S. Humphrey, *Introduction to the team software process*.
- I. Sommerville, *Software Engineering*, Chapter 25