Software Development Teams

Talent wins games, but teamwork and intelligence wins championships.

Michael Jordan
First Team Member Contribution Assessment

- Survey is linked on the calendar
- Complete by Fri midnight-- Required
- Your score will be posted in the gradebook (private to you). Significant comments will be summarized (no names disclosed) and mailed to you by me.

*Educational round focusing on SRS and start of SDS/ZFR*
First, what is a team?

A group is a collection of individuals with a common interest

whereas

A team is a cohesive coalition of individuals working together towards a common goal
Another definition

A team is a set of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable.

Katzenbach and Smith

There is no “I” in “TEAMWORK”
Why have teams?

- Having more people has benefits (that can be exploited)
  - Attack bigger problems in a short period of time
  - Utilize the collective experience of everyone

- Having more people has risks too
  - Personality conflicts
  - Coordination issues
  - Need to establish clear ownership or can have duplication of effort
  - People can fall in with masses instead of sharing a great idea
  - Not taking individual responsibility/accountability because it’s a group
  - Need to be careful to have the “right” number
More on communication

Do not underestimate the power of communication!

Communication requirements increase with increasing numbers of people
- everybody to everybody $\rightarrow \frac{n(n-1)}{2}$
- even just somebody to everybody $\rightarrow n-1$

Every effort at communication is a chance for miscommunication
Results-driven structure

Results-driven structures have:

1. Clear roles and responsibilities
   - Each person knows and is accountable for their work
2. Monitor individual performance, hold people accountable
   - Who is doing what, are we getting the work done?
3. Effective communication system
   - Available, credible, tracking of issues, decisions
4. Fact based decisions
   - Focus on the facts, not the politics, personalities, …
Typical SW team structures

- Include:
  - A person with project management responsibilities
  - A person with functional management responsibilities
  - Several “developers” in a broad sense: programmers, testers, integrators
  - A person with lead developer/architect responsibilities

- These could be all different team members, or there could be a large amount of overlap.
Project manager

- Role

- Responsibilities
Architect/lead developer

- Role

- Responsibilities
Developer

- Role

- Responsibilities
More on team organization

Pragmatic Programmer Tip:
“Organize Around Functionality, Not Job Functions”

• What are some benefits of organizing teams around:
  o … functionality?
  o … job functions?
Readings

Summary 4

- Pragmatic Programmer, Pragmatic Teams p. 224-230
- Summary of “The Five Dysfunctions of a Team” by Patrick Lencioni

Also

- Software Project Survival Guide, p.103-107 on team organization