Lecture 3: Teams

Administrative
- If interested in an external project, submit proposal ASAP
- Internal project descriptions are on website
- Upon forming a (partial) team, submit survey
- Upon forming a complete team, schedule meeting with client, start work on project plan (see website) – due Fri, Feb 11

Objective
Work effectively on a development team

Necessity of teams, organization
- Large projects require many people
  - Need to deliver while still relevant
  - Diversity of ideas, peer review improve quality
  - Utilize many different skillsets
- Multi-person development requires collaboration

Team size
- Number of pairwise communication paths is quadratic in team size
- Cannot understand full system, keep pace with all updates
- Need efficient meetings, decisions
- Quality benefits from consistency

Effective teams have 4-10 members; sweet spot is around 7

Myth of the man-month
- Development effort often measured in person-months
  - How much does one developer accomplish in one month?
- Suggests that work could be sped up by adding more developers
- Brooks’s law: “adding manpower to a late software project makes it later”
Which plot shows how project duration varies with manpower for a partitionable task without complex interdependencies that requires onboarding?

Team member roles

- No “team leader”
  - Project is a shared responsibility
  - Team decisions by consensus
- Team point-of-contact
Communicates with client (scheduling, reports)
Communicates with course staff
• Project manager
  o Maintains schedule
  o Monitors risks
  o Does not dictate tasking
  o May rotate
• Toolsmith
  o Infrastructure expert (DevOps)

Roles during meetings:
• Moderator
• Recorder

Developer strengths and motivations

Developer growth
• Mentorship
• Training
• Self-learning
• Documentation
• Reviews
• Goals (w/ manager assistance)
• Osmosis (slow, shallow)

• At a point in time, some devs may be much more productive than others
• But all devs have the potential to become more productive
  o Maximize benefit by "leveling up" early
  o "Teach a man to fish" parable
• Will get most out of this course by leveling up your whole team

Team disfunctions
  1.
  2.
Examples of disfunctional behavior
- **Yak shaving**
  - Balance quality/growth with short-term productivity
  - Prioritize simplicity
- **Bike-shedding**
  - Prioritize addressing the largest technical risks

Setting expectations
- **Avoid misunderstandings, disappointment by setting clear expectations**
  - Capture in writing: team charter, iteration plan
- **Be consistent, firm when expectations are not met**
  - Address acute issues ASAP
  - Address persistent, lingering issues at retrospectives
- **Communicate often, ask for help if stuck**

Meetings
- **High-bandwidth, high-visibility synchronous communication**
- **Expensive in terms of person-hours**
- **See John Cleese’s Meetings, Bloody Meetings**

Tips
- **Have an agenda**
- **Circulate agenda + docs ahead of time**
- **Record action items**
- **Review previous action items**
- **Spin off detailed discussions if not relevant to all or not prepared for**
- **Avoid status reports**
- **Beware slides**

Client communication
- **Respect client's time**
 Unless client requests chat-like communications, try to batch questions
  - Come prepared, stick to agenda, respect end time

• Prefer synchronous meetings
  - Higher bandwidth
  - Lower error rate
• Messages should go through point-of-contact
• Meetings should include most of development team
  - No one-on-one meetings
  - Need multiple viewpoints
  - Separate moderator from recorder

Retrospectives
• Process improvement
• Retrospectives are regular
• Like milestones, phased development – gates for evaluation, corrective action
• Postmortems come after an incident
  - Discussed in Google reading

Breakout
• What makes a good team player?

Peer review
Rubric will evaluate the following:

• Professionalism
  - Attendance/promptness
  - Preparation for meetings
  - Respectful interactions
• Quality of contributions
• Effective communication
  - Reports problems early
  - Communicates clearly
• Initiative
Contributes to brainstorming
- Volunteers for tasks
- Healthy team dynamics
  - Raises and resolves conflicts
  - Trusts other members, helps them grow

Collaboration tools
- Make sure entire team and client are onboard if using external services
  - Remember confidentiality obligation