Software Engineering Teams

Software Requirements & Project Management
CITS3220
Some Anecdotal Observations about new SE Graduates

- Graduates have high technical competence *but*,
  - Are not good team players,
  - Have very little feel or interest for company goals,
  - Tend to be overconfident, overoptimistic as regards project effort estimation and late in project delivery,
  - Tend to adopt new technologies recklessly, with little or no risk and cost assessment,
  - Prefer to start, or restart, development efforts from scratch, especially if they had not participated in the earlier development efforts
Motivation for Groups

- Why do we need groups in software engineering?
  - most products are too large for 1 person
  - one person cannot be proficient in all the information technology options that exist today
Mythical Man/Month

- Why can’t we calculate group size with?
  - person months = time allocated * persons
  - e.g. 6 person months effort required, and 2 months allowed, means we need 3 people to do the job

- Groups don’t scale this way because
  - some tasks can be shared, others cannot
  - communication/coordination overhead
  - Brooks’ Law: “adding people to a late software project makes it later”
Project Factors for Choosing Group Type

- difficulty of problem to be solved
- size of program to be developed
- time group will stay together
- degree to which problem can be modularised
- required quality and reliability
- rigid delivery date?
- degree of sociability (communication) required
Democratic Approach

- Group makes decisions together
  - nominal team leader
- Encourages egoless programming
  - encouraged to find faults in “shared” code

**Advantages**
- positive attitude towards finding faults → find faults early
- works well for difficult problems (e.g., research)

**Disadvantages**
- hard to impose… often spring up spontaneously
- may spook management
- hard to scale?
  - need to move to hierarchical structure for implementation

- Pressman calls this Democratic Decentralised organisation
Chief Programmer Approach

- All communication through chief programmer
  - acts as manager & technical lead
  - other team members are specialists (e.g., librarian)

- Advantages
  - communication channels reduced

- Disadvantages
  - lack of communication is unrealistic
  - finding a chief programmer w/ tech & managerial skills

- Pressman Controlled Centralized

```
Librarian  Chief Programmer  Back-up Programmer
  Programmer  Programmer  Programmer
```

Technical Managerial Approach

- Split group management across two people
  - group leader responsible for technical decisions
  - manager responsible for budget & performance appraisals
  - some overlap that must be clarified (e.g., vacation for key people)
Technical Managerial Approach

- Split group management between several people
  - Group leader responsible for technical decisions
  - Manager responsible for budget & performance appraisals
  - Some overlap that must be clarified (e.g., vacation for key people)
Technical Managerial Approach

- Split group management between several people
  - As before plus
  - decentralize decision-making to get advantages of democratic teams
Roles

- **Role**: a set of responsibilities in the project assigned to a person or team
- For example, analyst, system architect, tester, developer, manager, reviewer
- One person can fill one or more roles
What is the best team size?

- 3 members
  - PRO everyone has a chance to speak
  - CON one member can dominate
  - CON too many hats for each member

- 4 members
  - PRO works even if someone drops out
  - CON ties when voting (ancient Greek prob)
- 5 or 6 members is ideal
  - PRO face to face meetings good
  - PRO diverse perspectives for creative thinking
  - PRO specialisation of one role per member
- 7 or 8 members
  - CON longer meetings
  - CON coalitions and competition can arise
Group Diversity
Identifying Individuals’ Skills

- Application domain skills
- Communication skills
  - e.g. negotiation, communication with non-experts
- Technical skills
  - knowledge of specific technology and tools, programming skills, design risk assessment skills, ability to find workarounds
- Quality skills
  - attention to detail, boundary case identification, follow procedures
- Management skills
  - assess personal skills, motivate, feedback, assess project risks
Personality: Belbin Team Roles

- action-oriented roles
  - Shaper, Implementer, and Completer Finisher
- people-oriented roles
  - Co-ordinator, Teamworker and Resource Investigator
- cerebral roles
  - Plant, Monitor Evaluator and Specialist

http://www.belbin.com/
More personality tests: Jung – Briggs-Myers Typology

- Extraversion – Introversion (I/E)
  - how a person is energised (external or internal world)
- Sensing – Intuition (S/N)
  - how a person attends (perceives information)
- Thinking – Feeling (T/F)
  - how a person decides (information processing)
- Judging – Perceiving (J/P)
  - how a person lives (implementation of processed information)

http://www.humanmetrics.com/
Group Stages

how to become an effective group
Forming

- learns about the opportunity, agrees on goals and on the resources
- Task exploration, polite discussion, exploration of each other

Storming

- different ideas compete for consideration
- Disagreement, criticism of people (vs ideas), push for influence, posturing
- Goal/task orientation, mutual co-operation
Group Stages (cont)

Norming
- adjust their behaviors to each other as they develop working habits
- Definition of roles and behaviours, agreement on procedures, value recognition

Performing
- high-performing teams are able to function as a unit
- Team members are motivated and knowledgeable
- competent, autonomous and able to handle the decision-making process without supervision
- dissent is expected and allowed
Communication in Software Projects
Communication is important

In large system development efforts, you will spend more time *communicating* than *coding*.

A software engineer needs to learn the so-called soft skills: technical writing, reading documentation, communication, collaboration, management, presentations.
Lack of Communication Leads to Accidents

“In most of the major accidents of the past 25 years, technical information on how to prevent the accident was known and often even implemented. But in each case the technical information and solutions were negated by organisational or managerial flaws.” Leveson, Safeware, p48
Lack of Communication Leads to Accidents

“The Mars Climate Orbiter Project exhibited inadequate communications between project elements during its development and operations phases. This was identified as a contributing cause to the mission failure.”

p21, Report on Project Management in NASA by the MCO Investigation Board March 2000
Definition - Mode

Communication mode

- Type of information exchange that has defined objectives and scope
- Scheduled: Planned Communication
- Event Driven: Unplanned Communication
Definition - Mechanism

Communication mechanism

- Tool or procedure that can be used to transmit information
- Synchronous: Sender and receiver are available at the same time
- Asynchronous: Sender and Receiver are not communicating at the same time.
Classification of Communication

- Communication Mode
  - Scheduled Mode
  - Event-driven Mode
- Communication Mechanism
  - Synchronous Mechanism
  - Asynchronous Mechanism

is supported by
Communication Summary

- **Modes**
  - Scheduled communication
  - Event-driven communication

- **Mechanisms**
  - Asynchronous communication mechanisms
  - Synchronous communication mechanisms

- **Examples**
  - Weekly meeting
  - Project reviews
  - Online communication (discussion forum, email, web)

- **See Bruegge and Dutoit Chapter 3 for more details**