Imperial College London

Software Engineering Practice

Communication Structure and Scheduling

Communication Structure

Conway's Law:

"…organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations."

– Melvin Conway, 1968

Communication Structure

- In-person communication
  - Frequency and duration of meetings
- Electronic communication
  - Email, mailing lists
  - Instant messaging
  - Online collaboration tools (e.g., shared documents)
- Code management
  - Version control system
    - Can be configured to send all updates to a mailing list
  - Code documentation (e.g. Javadoc, Doxygen)
  - Bug/feature management (e.g. to-do list, bug database)

Group Projects

- Work as a team
- Each team member should contribute to the team
  - Each team member should improve his/her software engineering skills in the process
  - Document team dynamics

- If problems arise, address them **ASAP**:
  1. First try to resolve issues internally
  2. If 1 fails, talk to your supervisor
  3. If 1 and 2 fail, talk to me

Scheduling

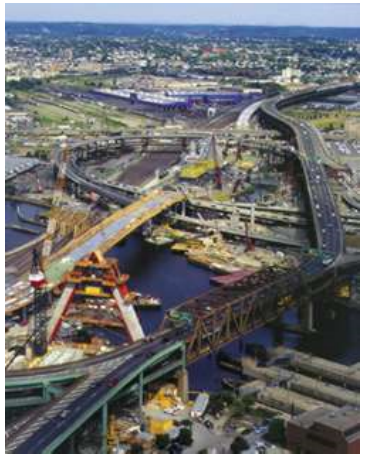
"More software projects have gone awry for lack of calendar time than for all other causes combined."

– Fred Brooks, 1975

Imperial College London

Estimates Almost Always Too Optimistic

Sydney Opera House
Started: 1958
Estimated completion & cost: **1963 ($7 million)**
Actual completion & cost: **1973 ($102 million)**



The Big Dig (Boston, USA)
Started: 1982
Estimated completion date: **1985 ($2.8 billion)**
Actual completion & cost: **2007 ($14.6 billion)**

Scottish Parliament Building
Started: 1998
Estimated completion & cost: **2001 (£50 million)**
Actual completion & cost: **2004 (£431 million)**

For a fair comparison, costs need to be adjusted for inflation, but actual costs still MUCH higher than estimated

What about student projects?

Famous 1994 study by Buehler, Griffin and Ross:

- Students asked to estimate the time needed to complete senior thesis

Estimate (avg):	34 days
Worst case (avg):	49 days
Actual (avg):	56 days

Estimates Are Almost Always Too Optimistic – Why?

- We assume everything will go according to plan
  - But it rarely does...
- We assume even effort
  - But usually more work is done toward the end
- We confuse effort with progress
  - Effort is necessary, not sufficient
- No proper response to slippage
  - More on this later

"Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law."

Any Solutions?

Setting Effective Deadlines

- Frequent
  - To make slippage visible
- Verifiable
  - That can be checked objectively by the team and outsiders
    - **NO:** 90% of all the code done
    - **YES:** 90% of Module A coded
- Realistic
  - Do not let clients push you toward unrealistic ones
  - Do not push yourself toward unrealistic ones
    - remember Hofstadter's Law

Addressing Slippage

- You must address it
  - No matter how small, it quickly adds up

- How can you address it?
  1. Add more people
  2. Add more resources
  3. Alter product
  4. Alter schedule

Adding More People

- The man-month myth
  - Progress proportional to the number of people
- Why not?
  - Ignores communication costs
  - Ignores the cost of getting new people up-to-speed

"Brooks' Law: Adding manpower to a late software project makes it later."

"*The Mythical Man-Month" book* by Fred P. Brooks
- Series of influential essays on project management

Adding More Resources

- Things you can buy:
  - More CPU power, more memory, more storage, etc.

  Always a solution?

- Works only in limited situations:
  - When resources are the bottleneck

- May be hard to integrate mid-stream

Adding More Resources

- Overall improvement to a system when only one of its parts is improved
  - Overall improvement is less than you might think

CPU: 1/3 I/O: 2/3

CPU upgrade: 10x Overall speedup = **1.43x**

CPU upgrade: 1000….0x Overall speedup = **1.5x**

Altering Product/Schedule

- Most times the only realistic options
  - Alter product
    - Drop or limit features
    - Sacrifice performance, usability, etc.
  - Alter schedule
- Will not make clients happy!