Practical SW Project Management Issues

What is it like, to manage a team effort, building a complicated system?

What is the work environment?

Based on…

- Steve McConnell (Code Complete), Fred Mooey (I Sing the Body Electronic), Tom Demarco (PeopleWare, The Deadline), Tracy Kidder (The Soul of the New Machine), Fred Brooks (The Mythical Man-Month)
- IBM, Kodak, Motorola, NATO, USAF, Lockheed, Banamex, more, projects in excess of $5 million.
- The IEEE index on salaries and job performance
  - Where EE and CSE graduates are working

SW Engineering Subculture

- SW Engineers:
  - Rarely work alone
  - Rarely understand the entire system
  - Stop when the money stops (is this a new concept?)
- SW Teams:
  - 4-12 people (not everybody codes)
  - greater than 1 year schedule
  - more than 500,000 LOC  (Lines Of Code)
SW Engineering Subculture

- Work through gates: assess the performance of the team and system before moving on to the next step
- SW budget exceeds HW
- SW Engineers: work on many projects at once
  - some new
  - maintenance or enhancements on old
- SW Project Managers largely come from a CSE or EE background – not MBA, not Civil, not Mechanical

SW Employment

- Embedded Systems – standalone, event- and error-driven systems – factory, medical, command & control
- Information Technology – MRP and databasing as decision aids
- Development Vehicles – tools that aid in the production of software and systems, including operating systems
- Test
- Research
- Communications / DSP / RF
- many, many more

Examples

- Digital image processing, laser photodeveloping, and packaging of satellite images.
- Air situation display for the Persian Gulf war.
- Precision high speed manufacturing of wax lips.
- Radio metro-area network design for lottery terminals in Moscow.
- Clean-room manufacturing of Huber MediPort needles.
- NASCAR – fuel consumption estimation during races.
- 802.16 MAC & PHY Layer standards for the IEEE/FCC.
- Lights and effects for Cirque de Soleil, Phantom of the Opera, Genesis
Customers may not be users.
Designers may not be coders.
The SW Project Manager has technical, budget, and schedule concerns, and is the “people coordinator”

Responsibilities of the SW Project Manager
- Often the Technical Lead
  - Domain Expert
- Central to Architecture and Design
  - Structured, OO, Info-Flow, Event-Driven or Error-Driven State Machine, etc.
- Determines the Development Environment
  - OS
  - Language
  - Compilers and Tools

Responsibilities (continued)
- Determines the Execution Environment
  - OS e.g. VxWorks, WindowsCE, rtLinux, etc.
  - Cooperative Technology Assessment – Client/Server topologies, pt-pt, pt-multip, networking, etc.
  - Works with the HW developers on platform choice - e.g. DSP, PLC, when not in the spec.
- Decides on the Development Methodology
  - Waterfall, Incremental, Chaos, Hero-based, etc.
  - enforces company policy
Responsibilities (continued)

- Arbitrates Technical Disagreements
- Provides Scheduling and Budget Guidance
- Buffers Management from Implementers
  - Estimation
  - Status Reporting
- Assembles the Optimum Team
- Handles People / Social / Interpersonal Issues
- Uses Team Guidance on all of the above

Software Failure

- Success is not always assured.
- Software Project Managers use policy, procedures, methodology, engineering, and people skills to control risk and prevent failure.
- Coding is not foremost on a Project Manager’s mind.

How Do Software-based Projects Fail?

- Unhappy Customer – do not get what they expect or expect what they get
- Out of time and $$$
- Unhappy user
- Safety compromised, including death
- Technically inadequate
- Does not contribute to the company business case
- Menace to Society
side note

- Life-long Learning
  - Read books, including novels, on Engineering
    - I Sing the Body Electronic, by Fred Moodey
    - American Genesis, by Thomas Hughes
  - Apollo 13, Office Space, October Sky
  - Watch for newspaper articles

A Manager’s View

1. You will be assigned to a long project that you don’t want, or a project that you do want, but has changed considerably in budget or function. But you’ll assume that with a fresh start and talented people, you can succeed - what is your first task?

2. You won’t get the people you want. Most likely, they’ll look to take other jobs in other companies, because they can. This will happen before the project, and during – what is the platinum rule on how you are to treat people? How do you protect the project from turnover?
3. You will be one of a few people who understand the project completely, technically and fiscally, from start to finish. 
Should you be the master of all details?

4. You will be the only person to have all sides to the project’s story: the customer, your company, and the implementation team. All sides will try to make you their own - where is your loyalty?

Side note: What was “noted” as the single greatest cause of software project failure?

5. Sometimes, it will be a personal playground: the ability to buy and try technology and equipment, solve problems, research approaches and solutions, think…

This hints at the reason on why we become Software Engineers
6. By far, most problems will be people-related, and social, and not technical or even financial.

Time spent on people-problems – 65% of SW Project Managers said greater than 90%

7. People in which you had little confidence will surprise you with talent and loyalty.

8. People on which you had counted will disappoint you with laziness and pettiness.

Appreciate team makeup, organizational behavior, the SW Engineering subculture
Read “PeopleWare” by DeMarco and Lister

9. Your best person will quit. Your ally in management will quit. Your counterpart in the customer organization will quit.
10. Life will intrude relentlessly on your team: illness, vacation, day care, divorce, lottery winners, gambling, drinking on the job, theft, tempers.

Side note: manage the whole person

11. The project will proceed along in a positive direction, you will hit milestones, things will begin to work regularly - how can you maximize this?

Incremental Development within a design methodology

12. Approaches that you counted on will fail. You will have to back-up and restart on a number of major points - what is your best strategy?

Incremental Development within a design methodology
13. The specification will change - what will be your response to the customer, to the team?

   *Side note: manage the requirements in the same way as you manage code*

14. You or your team will feel stress for bad decisions of which you were not a part.

15. You or your team will feel stress for bad decisions that you made.

16. Someone else will get major credit and reward for something you did.

17. Company rewards might not include your team.

   *note : Status Reporting - Keep everyone (your team, management, customer, user) appraised and aware.*

18. During integration, the system may not be debuggable - unless you've done a few things right. What are they?

   *Incremental Development within a design methodology (the whole subject of this course).*
19. A working system will have 5-10 bugs for every 1000 lines of source code, in the software trouble report - response?

20. The customer will deem your successful system as “unusable” - why would they do this? How do you respond?