Session 1: Preliminary notes

COMP 477 / 377 in the curriculum

Relationship to COMP 320

- There is some overlap between:
  a. COMP 320 -- Systems Analysis, and
  b. COMP 477 -- Project Management

- That's unavoidable because:
  - A project manager needs to understand the essential role of systems analysts.
  - Some organizations (*not recommended*) assign the senior systems analyst to act as project leader, too.

- Don't worry:
  - If you've already taken COMP 320 (or an equivalent) we won't bore you with a lot of repetition.
  - If you haven't, we won't assume that background.

Session 1-A: Introduction and background

- What is a project?
- What is a successful project?
- What is project management?
- Historical background?

Common I.T. activities: Which ones are projects?

- Developing and installing a new application
- Moving the data center to a new location
- Operating the computer network
- Converting servers to a different O.S.
- Supporting a user-help desk
- Entering data into a production application

*Why?
Common I.T. activities: Which ones are projects?

- Developing and installing a new application
- Moving the data center to a new location
- Operating the computer network
- Converting servers to a different O.S.
- Supporting a user-help desk
- Entering data into a production application

A project has a definite END

Some noteworthy projects (non-I.T.) in history

- Building the pyramids
- D-Day invasion of Europe, June, 1944
- Moon landings (1968 - )
- Others? (They don't have to be good)

Project management scope

- Project management = project planning + project control + team leadership + communication

- Project planning occurs
  - mainly at the beginning of a project or major project phase,
  - but changes and replanning can occur throughout a project

Information Systems Projects

- Usually concerned with implementing a new or modified application system to support a set of related business or functional activities
  - Software components of the system may be:
    - a. Custom developed or assembled by in-house professional staff
    - b. Custom developed under contract to a software firm
    - c. Purchased from a packaged application software vendor

- Other kinds of I.S. Projects:
  - Developing a software tool
  - Installing/converting an operating platform
  - . . . etc.
A successful project is one that
1. Meets agreed-upon objectives (end result)
2. within an agreed-upon budget
3. at an agreed-upon target-date
4. while complying with applicable standards and constraints

Which of those is most important?
Which is most visible?
What's all that agreed-upon stuff?
Who did the agreeing and when?

Successes and failures in application system development

- In the half century that we've been developing application software, only a minority of projects, by that definition, have been successful!
  - Mostly due to schedule slippage and cost overrun
- Although we keep learning lessons and devising "breakthrough" methods, the ratio of successful projects today is not much better than it was in 1975!
  - Why is that?

Common reasons for cost overrun and schedule slippage

- **Estimation** failures:
  - Lack of management discipline
  - Wishful thinking
  - Irrational pressures
  - Excessive trust in fad "breakthrough" methodology
  - Unrealistic assumptions about the environment
- **Performance** failures:
  - Lack of rigorous change control
  - Lack of essential skills
  - Poor communication among key people

Out-of-control project:

- Schedule slippage and/or cost overrun—usually discovered late in the project, often after a long succession of optimistic status reports
- Incremental and repeated: once you've invested the initially estimated effort, you're forced to invest the overrun, too (the "Vietnam War" syndrome)
- There's often no rational way of stopping a project once it's out of control.
- Two common causes account for most out-of-control projects.
  - We'll see what they are later.
What can be done about that?

- That's what this course is about.
- Some managers in responsible positions claim that it's hopeless!
  - "Uncertainty is just the nature of I.T."
- Some keep looking for the "silver-bullet" breakthrough.

Session 1-B: Project roles and staffing

What are the major project roles?
How are they specified?
How do we fill them?

Some Project Roles
- Project manager
- Project sponsor (user, customer)
- Systems analyst
- Designer / Chief programmer
- Programmer
- Database manager
- Quality reviewer

Which are the project team members?
Whom does the P.M. have control over?
Who are the "developers"?

Roles versus jobs

<table>
<thead>
<tr>
<th>Job (or position)</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>relatively permanent</td>
<td>exists for duration of an activity (Project roles are always temporary.)</td>
</tr>
<tr>
<td>individual occupies exactly one at a time</td>
<td>Individual may take on multiple roles</td>
</tr>
<tr>
<td>recognized and administered by H.R.; tied to salary, rank, status, etc.,</td>
<td>needn't be official -- a management tool at any level</td>
</tr>
</tbody>
</table>

Which is Project Manager?
Project Role Categories

- Project team members
  - Project manager
  - Systems analyst
  - System designer
  - Programmer
- Project consultants / advisors
  - Database manager
  - Quality reviewer
  - Methodology administrator
  - Systems integrator
- Project sponsors
  - Manager / funding authority
  - Prospective user

Defining a role

- To successfully manage people in the various project roles, the project manager must understand exactly what each role does.
- Each staff member should understand his or her own role in exactly the same way as the project manager does.
- The role definition, then, is like a contract between the incumbent and his or her boss.

Therefore . . .

- An organization needs rigorous written role definitions
- But what if the organization doesn't already have any role definitions?
- Then it's up to the project manager to define them for his or her project.

Formal Role Definition

- Much more concrete than the typical organization's job descriptions
- Avoids vague activity terms, such as:
  - participates in . . .
  - assists with . . .
  - contributes to . . .
  - advises on . . .
  - concurs in . . .

Why not?
Are these consistent?
- Howard comes to work early every day and stays late.
- He keeps busy most of the day doing assigned tasks that the organization wants to be done.
- He does most of those tasks well.
- When he goes home, Howard feels satisfaction in what he accomplished that day.
- His boss just rated Howard's performance poor!

What's a likely explanation?

Accountability for results
- More important than the activities an individual engages in are the results he or she is accountable for producing.
- Therefore a rigorous (and useful) role definition must specify measures of accountability.

Role Definition structure
- A successful form of role definition is the so-called RAA structure, consisting of three sections:
  - **Responsibility** -- The activities the incumbent performs
  - **Accountability** -- The results the incumbent is expected to produce (also called "measures of performance")
  - **Authority** -- What the incumbent is permitted to do without specific higher approval.

Accountability versus annual objectives
- Many organizations require a list of goals an individual commits to accomplish during the coming year.
- The accountability list is a list of results the individual commits to produce.

How are they different?

Let's look at an example:
Project Manager
Another critically important role: programmer

- Origin and evolution
- Role in a modern I.S. project
- Titles and status
- Productivity

1950s origins

- A programmer (a technically competent problem solver) would confer with a problem sponsor (potential user with funds and authority) on requirements for a new or modified application.

What happened next?

Why didn't that work for complicated business applications?

Changing role of the programmer: the 1950's

- Given a problem, implement a computer-based solution to it - a prestigious job!
- Who gave the problem to the programmer?
- How was the problem stated?

1960s crises

- Growing demand from business along with cheaper computers led to:
  - larger project teams with >4 programmers
  - hordes of poorly trained, underqualified programmers.
  - major project fiascos
- Desperate organizations "promoted" their most mature and knowledgeable programmers to systems analysts.

What did those systems analysts do?
Changing role of the programmer: the 1960's

Who created that design?

- Given a detailed design (usually flowcharts & record layouts) code and test a set of programs to implement it.
  - Such a programmer didn't actually solve a problem stated by the sponsoring user!
  - They were often (more accurately) called "coders"

Explosion of demand (why?) --
Hordes of poorly qualified trainees.

1970's maturing

- The "structured revolution" (~1975-1985) established a clear distinction between programming and systems analysis.
  - The systems analyst would rigorously define the problem to be solved
  - The programmer would solve a well-defined problem, usually using a computer.

Did that always work well?

The role of the modern programmer

- Given a well-defined problem, to design and implement a computer-based solution.
  - That solution may or may not require writing new software (programming)
  - A competent professional programmer is aware of alternatives and may even try to avoid writing programs.

Programming = design + coding + testing

The systems analyst role

- The main role of the analyst is to determine and document system specifications.
  - What the proposed application must do, not how.

- Other names for system specifications include:
  - detailed user requirements
  - external system design
  - functional specifications

- What about "analysis and design"?
Is this a common career path? a desirable one?

- Programmer
- Systems analyst
- Project manager

Not a common successful career path

- Few people are equally skilled at (or interested in) both systems analysis and programming.
- "Promoting" your best programmer to become a systems analyst often loses a good programmer and gains a bad analyst.
- Salary ranges and prestige for the two roles should be comparable.
- The rare individual who is skilled at both should
  - Do analysis some of the time
  - Do programming some of the time
  - Do a mixture of the two never.

Choosing programmers

- Studies repeatedly show a 20-to-1 range in productivity among experienced programmers.
- Their range in salary is at most 3-to-1
- Furthermore the most productive programmers usually produce higher quality programs than the least productive.

What strategy does that suggest?

Offshore programming

- Managers are often tempted to outsource a programming job to a country with low labor rates
  - That's especially tempting on huge projects with dozens or hundreds of programmers.
- That may work out well, but:
  - Communication among team members may be difficult or costly, due to time-zone spans or cultural differences.
  - It's hard to judge the competence of individual programmers you haven't screened and supervised yourself.
  - Misunderstandings and other problems may go undetected/undiagnosed for many days.

Can we outsource systems analysis, too?
Offshore programming (continued)

- Other countries are competing for that business.
  
  What are some of them?
  
  What do they offer?

- Many countries support a full range of programming skill from superb to atrocious.

The bottom line on project roles

- You can call the roles anything you like, as long as:
  1. Every result that must be produced is on the accountability list of some project role
  2. Every incumbent has both the resources and the authority to produce the results he or she is accountable for.
  3. No two roles have conflicting authority.

  Trying to conduct a major project without satisfying those criteria will almost always lead to project failure.