Session 12: Odds & Ends

Documentation review
Justification review
Post-project review phase
Status reporting with MS-Project
Negotiations & compromises

The Project Workbook
- A repository for all the documentation about a project.
- It can
  - be self contained, or
  - contain references (or links) to other documents
- It can be deployed
  - over an Intranet
  - or on paper in physical binders

Authors & audiences
- The Project Workbook can (and should) be read by
  - project team members
  - sponsoring user representatives
  - management
- Sections are written by
  - the Project Manager
  - Systems Analyst
  - Database Designer
  - Software Designer
  - Individual Team Members

Who approves new or modified content?

In Particular
- Business objectives (Deliverables from ph. 2 of our sample SDLC) form the basis for understanding what the project is about
  - How it relates to the users' business
  - Justification for doing it
  - Context for detailed requirements (ESD)
- Will be read by
  - User representatives to confirm scope
  - Team members, especially new ones joining the team, to orient them to the purpose and context
- Like the later ESD they must be in terms all audiences can understand.
Impact of organization infrastructure on project workbook

- The stronger the organization's methodology infrastructure is, the less a Project Workbook must contain:
- A Project Workbook needn't duplicate explanations of organization-wide standards that apply to all projects, such as
  - SDLC
  - Data representation
  - Operational platform
  - Programming language choice
  - etc., etc.

Organization-wide Methodology: Levels

- Mandatory Standard: Projects must comply unless they secure permission in advance to deviate.
- Convention: Projects are expected to comply unless they can state a definite reason for deviating.
- Guideline: Technique that projects should find helpful.

Projects are free to add their own.

Deviating from a Mandatory Standard

1. **Project** submits request to deviate, explaining:
   - Why it would be impossible or prohibitively expensive to comply
   - Expected future and global impact of the deviation.
2. **Review authority** either
   - concurs or
   - states the negative impact of the deviation
3. If those parties can't agree, then **I.T. Management** makes final decision.
   (Another alternative is to propose a change to the organization's standard.)

Project Justification

- The simple (oversimplified) formula is:
  \[ \text{ROI} = \frac{\text{gain} - \text{cost}}{\text{cost}} \]
- But we have to account for **when**
  - We incur the cost of development **before** we realize any benefits of having the new system.
  - Therefore schedule slippage may have a huge impact on ROI.
  - Benefits are recurring over the system's life span.
- We may also want to account for the time value of money, interest, taxes, etc.
Incremental Development and project justification

- Can incremental development compensate for the slippage problem?
  - Users could start realizing benefits from a partially completed system before they've spent all the funds?

- That depends:
  When is a partial or incomplete application system useful?

Status reporting

- We've seen that MS-Project is very strong for project planning, once you get it set up right.

- How well does it work for task-level status reporting?
  - There's no Status Report form
  - But we can still enter the needed information
    - Be sure you know how to do that!

Status reporting with MS-Project

- Add another column, Finish, to the Gannt task entry form.
  - The dates in that column will originally have been computed by MS-Project, based on predecessors and estimated durations.
  - A team member can later type an actual date over the computed date, either
    - a revised estimate, or
    - the actual date it was finished, e.g. today

- Should we let every team member update our master MS-Project file?

A simple individual's task status report

<table>
<thead>
<tr>
<th>TASK</th>
<th>Hours worked</th>
<th>Rem. hours</th>
<th>Completion date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>T12</td>
<td>5</td>
<td>7/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T20</td>
<td>18</td>
<td>7/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T33</td>
<td>4</td>
<td>7/5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Late-stage troubles

What can we do if we're running out of time before a promised start-up date?

a. Apologize and negotiate the slippage

b. Compromise and take short cuts
   Such as?

c. Cover up the problem and start looking for another job

Late stage decisions

- Pressures to compromise typically increase late in a development project.

- There's no longer any vague slack in the schedule. You can no longer say "Somehow we'll make up the time."

- What compromises are available?

Possible late compromises

- Condense the system/acceptance test

  What are the likely consequences?

- Defer some features

  When is that possible?

Post Project Review Phase

- Purposes:
  - To learn from experience; do better next time
  - To evaluate performance, esp. project manager

- Many organizations skip this phase, even though their SDLC calls for it!
  - Why?
    - They need the people for other projects
    - or they just never get around to it.
**What do we review?**

**Two things!**

1. The **end product**:
   a. Does it meet agreed-upon specifications?
   b. Does it meet reliability expectations?
   c. Does it meet performance / efficiency expectations
   d. Are the users happy?

2. The **project performance**:
   a. Did we meet target-date commitments?
   b. Did we meet budget commitments?
   c. Were we responsive to change requests?

**When should we do the post-project review?**

- The **project performance evaluation** should be done immediately after installation and start up.
  - Memories are fresh
  - We have the momentum

- The **end product evaluation** should wait until the users have had some experience.
  - One month or more is common
  - But not so long that the users forget what they originally wanted.

---

**The final project report**

- Detailed review information previously listed.

- Lessons learned and recommendations for future projects.