Large Scale Systems Design
G52LSS

Lecture 5 – Project Management

- Elements of Project Management
- Estimate Size of the Project
- Produce a Work-Plan
- Produce a Staffing Plan
- Coordinate Project Activities

**Learning outcomes:**
- describe the elements of project management:
- apply function point approach to estimate project size:
- describe purpose of work-plan and staffing plan:
- appreciate key issues of coordinating project activities.

---

Elements of Project Management

Project management involves: Planning and Control of all activities required for developing a system within given time, cost and functionality.

Project management usually requires a compromise between costs, functionality and completion time.

---

Steps of Project Management

1. Estimate size of the project
   - Consider all phases of the SDLC
   - Amount of coding required and its complexity
2. Produce a work-plan
   - Identify tasks and time required
   - Schedule tasks with Pert and Gantt
   - Refine project work-plan
3. Produce a staffing plan
   - Estimate staff required
   - Coordinate team-work
   - Manage motivation and conflict issues
4. Coordinate project activities
   - CASE tools for increased productivity
   - Standards, documentation, risks, etc.

---

Estimate Size of the Project

1. Estimate using **planning phase time**
   - A typical project on developing an information system takes: 15% in planning, 20% in analysis, 35% in design and 30% in implementation
   - Very simple and general approach that relies on the current system matching the conditions of typical projects
2. Estimate using **function point approach**
   - Based on the number and complexity of lines of code required for the system's input, output, processes, file management, and interfaces with other systems.
   - It also considers the programming language chosen to build the system
   - Human effort and time are estimated with simple formulae or historical data if available
Example using Planning Phase Time

Suppose that planning has been completed in 9 weeks with 4 people working full-time and 2 people working half-time.

The planning phase time is expressed as 45 person-weeks.

Then, the estimation for the rest of the project is:
- Analysis: 60 person-week approx. (20%)
- Design: 105 person-week approx. (35%)
- Implementation: 90 person-week approx. (30%)

The overall size of the project is 300 person-week or the equivalent to approx. 70 person-month.

Example using Function Point Approach

Imagine that job hunting has been going so well that you need to develop a system to support your efforts. The system should allow you to input information about the companies with which you interview, the interviews and office visits that you have scheduled, and the offers that you receive. It should be able to produce reports, such as company contact list, an interview schedule, and an office visit schedule, as well as produce thank you letters to be brought into a word-processor to customise. You also need the system to answer queries, such as the number of interviews by city and you average offer amount.

(Case 3-1 from Dennis et al. 2006)

1. Estimate system size in terms of number of lines of code and complexity of code

The estimated number of components required for the system are:
- Inputs: Company Info, Interview/Office Visit Schedule, Offers Received
- Outputs: Company Contact List, Interview Schedule, Office Visit Schedule, Thank You Letters
- Queries: Number of Interviews by City, Average Offer Amount
- Files: Company Information, Interview/Office Visit Schedule, Offers
- Interfaces: Main menu which directs users to Input, Reports, Queries

Calculate the Total Unadjusted Function Points for the system:

<table>
<thead>
<tr>
<th>System Components</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Low</td>
</tr>
<tr>
<td>Inputs</td>
<td>1.3</td>
</tr>
<tr>
<td>Outputs</td>
<td>4</td>
</tr>
<tr>
<td>Queries</td>
<td>2.3</td>
</tr>
<tr>
<td>Files</td>
<td>1.3</td>
</tr>
<tr>
<td>Interfaces</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Total Unadjusted Function Points (TUFF) = 81
Adjust project complexity using Adjusted Project Complexity (APC) scores if available and calculate the Adjusted Function Points

Suppose the system will be built in Visual Basic and the APC scores for this development platform are: 0.65 for simple system, 1.0 for an average system and 1.35 for a complex system

The Total Adjusted Function Points (TAFP) score for this system (perceived as simple assuming good Visual Basic skills) is calculated:

\[ TAFP = 0.65 \times 81 = 52.65 \]

Considering the approximate number of lines of code per Function Point for Visual Basic = 30, then:

**Estimated size of the system** = 52.65 \times 30 = 1579.50 lines of VB code

That is, = 1580 lines of code in Visual Basic to build the system

2. Estimate the human effort required

\[ \text{Effort (person-months)} = 1.4 \times \text{thousands of lines of code} = 2.212 \]

3. Estimate the time required

\[ \text{Schedule time (months)} = 3.0 \times \text{person-months}^{1/3} = 3.90 \text{ months} \]

Estimated time required to build the system = 4 months

Note: the estimates obtained with the function point approach correspond to analysis, design and implementation but do not include the planning phase.

---

**Produce a Work-Plan**

This is a dynamic document that defines the list of tasks and important information about each task. A project work-plan includes:

- Start date
- Completion date
- Estimated duration
- Actual duration
- Precedence information
- Deliverables
- Completion status
- Required resources
- Key milestones
- Staffing plan, etc.

---

**Estimate Tasks and Time Required**

- **Estimate tasks** according to objectives
- Identify tasks at various levels of detail
- The estimation of task time should be conservative and careful
- Keep track and revise estimates as the project progresses
- Estimates are rarely accurate, i.e. project can be ahead of schedule (not very common) or behind schedule (very common)
- Refine estimates as project progresses considering that margins of error are considerable even for well planned projects
Typical margins of error when estimating time and costs in large scale software projects:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Deliverable</th>
<th>Cost (%)</th>
<th>Scheduled Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Systems Request</td>
<td>400</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Project Work-Plan</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>Analysis</td>
<td>Systems Proposal</td>
<td>50</td>
<td>15</td>
</tr>
<tr>
<td>Design</td>
<td>Systems Specification</td>
<td>25</td>
<td>10</td>
</tr>
</tbody>
</table>

Source: (Dennis et al., 2006)

Suggested actions and associated risks if project is behind schedule:

- **IF next part of project is simple**
  - **IF estimates were pessimistic, THEN**
    - Lost time can be recovered
      - Do not change the schedule
  - **IF estimates were accurate, THEN**
    - Lost time cannot be recovered
      - No lost time in rest of the project
      - Increase schedule duration according to lost time
- **IF next part of project is of high complexity**
  - **IF estimates were pessimistic, THEN**
    - Lost time cannot be recovered
      - Further time can be lost in rest of the project
      - Increase schedule duration according to lost time
      - Consider compromising functionality and costs
  - **IF estimates were optimistic, THEN**
    - Lost time cannot be recovered
      - Further time can be lost in rest of the project
      - Consider compromising functionality and costs

Managing Changes To Functional Requirements

- **Scope creep** refers to adding requirements to the project after the scope had been defined.
- Likely to provoke *increase in costs and time*
- After project initiates, try to keep *changes to a minimum*
- Allow only *essential changes* to the specification
- Keep *track of implemented changes* and assess their impact on the project
- If changes are not incorporated then suggest them for *future versions* of the system
- Use *time-boxing* if appropriate

Produce a Staffing Plan

Determine **number of people required** for the project

Number of people = Effort (person-month) / optimal schedule

Adding more people to the project does not necessarily increase productivity because the management of larger teams is more difficult.

Staff must be assigned to each part of the project and a **staffing plan is produced** which describes:

- Number and kind of people required for each task
- Overall reporting structure
- Rules for team work, staff accountability, etc.
**Issues When Managing People in a Project**

- Set adequate team size
- Assign people to teams and tasks according to required technical and inter-personal skills
- Compromise between required and available staff
- Good communication and cohesiveness (resolve tensions, manage conflicts, establish norms, define roles, define accountability)
- Clear achievable goals by agreement and according to project conditions
- Clarify roles and responsibilities
- Team and individual motivation

---

**Coordinate Project Activities**

**CASE Tools**

- Use these tools to aid project management but not to drive it
- For example, MS-Project can be used to create a work-plan and staffing plan

**Standards**

- To ensure everyone follows the same procedures and rules
- For example, using the same format for reporting program bugs

**Documentation**

- Create documentation as soon as work is done to improve accuracy and good structure
- For example, document code as soon as it is tested and approved

---

**Managing Risk**

- Important to assess risks and take actions to eliminate them
- For example, ensuring requirement analysis is properly conducted to avoid scope creep

**Common Mistakes**

- Optimistic estimates
- Failing to monitor the schedule
- Failing to revise and adjust the schedule
- Add staff in response to a late project
- Poor project scoping
- Team work lacking any of the following: effective communication, interpersonal values, leadership, support form management
- Forgetting that project management involves: planning, organising, controlling, leadership and motivation.

---

**Additional Reading**

Chapter 3 of (Dennis et al., 2006)

Chapter 3 of (Kendall and Kendall, 2005)