

Lecture-7

PROJECT PLANNING

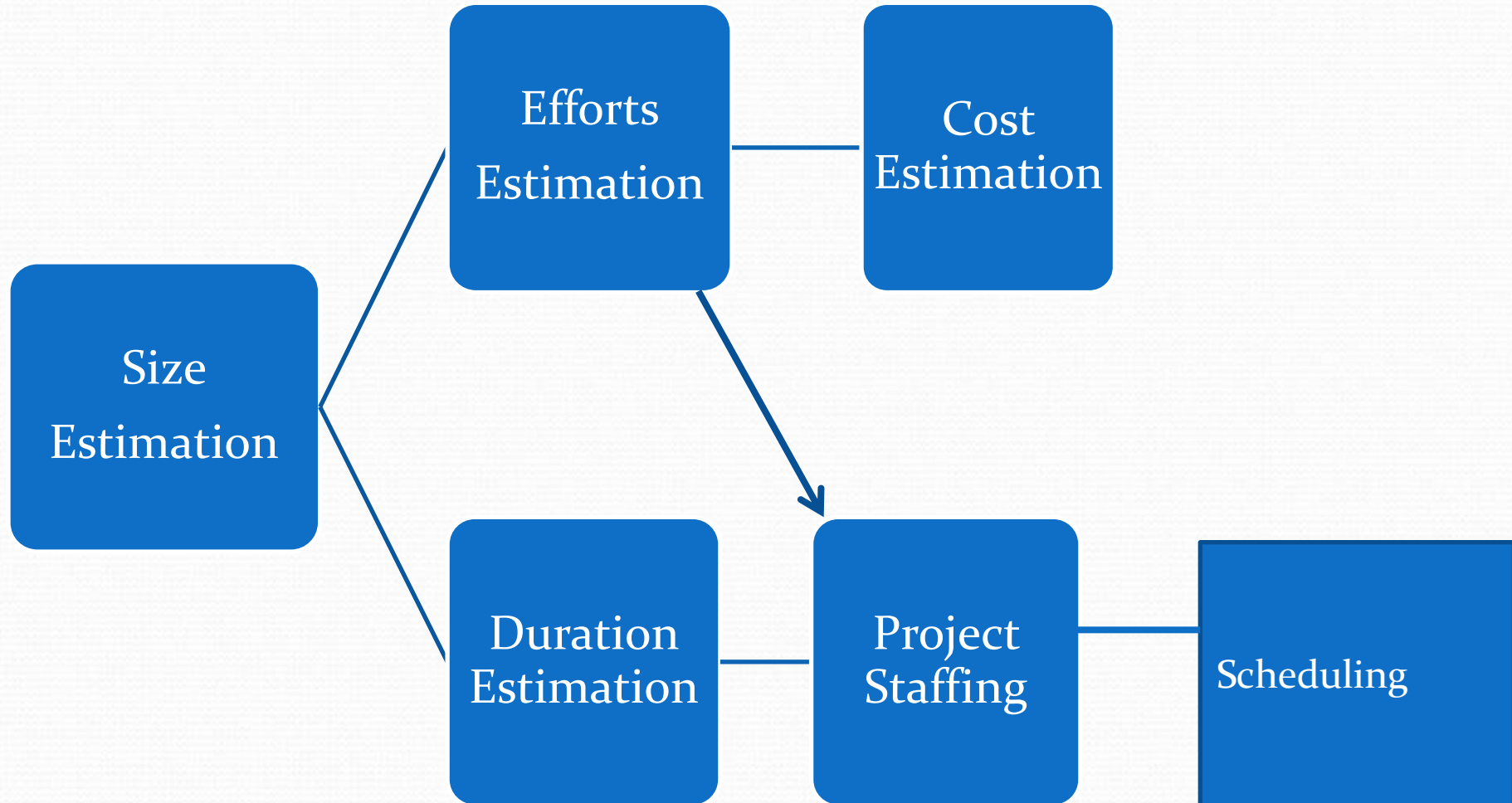
Dronacharya College of Engineering

Project Planning

Project Planning consists of the following essential activities:

- Estimating some basic attributes of the project
cost, duration, efforts
- Scheduling manpower and other resources.
- Staff organization and staffing plans
- Risk identification, analysis planning
- Miscellaneous plans

Precedence ordering among planning activities



THE SPM Document

- Once project planning is complete, project managers document their plans in a software Project Management Plan(SPMP) document.

1. **Introduction**

a) objectives

b) Major Functions

c) Performance Issues

d) Management and Technical Constraints

THE SPM Document

2. Project Estimation

- a) Historical data used
- b) Estimation Techniques Used
- c) Efforts, Recourses, Cost, and Project Duration Estimation

3. Schedule

- a) Work Breakdown Structure
- b) Task Network Representation
- c) Gantt Chart Representation
- d) PERT Chart Representation

THE SPM Document

4. Project Resources

- a) People
- b) Hardware and Software
- c) Special resources

5. Staff Organization

- a) Team Structure
- b) Management Reporting

6. Project Tracking and Control Plan

THE SPM Document

7. Risk Management Plan

- a) Risk Analysis
- b) Risk Identification
- c) Risk Estimation

8. Miscellaneous Plans

- a) Process Tailoring
- b) Quality assurance Plan
- c) Validation & Verification
- d) System Testing Plan

Software Measurement Objectives

- To understand the importance of measurement in software engineering
- To describe and compare the different metrics that can be used for measuring software
- To understand the important factors that affect the measurement of software

Why Do We Measure?

- To indicate the quality of the product
- To assess the productivity of the people who produce the product
- To assess the benefits derived from new software engineering methods and tools
- To form a baseline for estimation
- To help justify requests for new tools or additional training

Definitions

- **Measure** - quantitative indication of the extent, amount, dimension, capacity, or size of some attribute of a product or process.
- **Measurement** - the act of determining a measure
- **Metric** - a quantitative measure of the degree to which a system, component, or process given attribute (IEEE)

Project Size Estimation Metrics

Software metrics can be divided into two categories based on the type of measures used in evolving them

1) Direct Measures:

It includes cost, efforts, LOC, response time, resource usage and so on.

2) Indirect Measures :

Include functionality, quality, complexity, efficiency, dependability, ease of use ,quality of documentation.

Example: length of a pipe is a direct measure.

the quality of the pipes can only be measured indirectly by finding the ratio of the accepted pipes to the rejected.

Lines of Code Metrics

- LOC is the simplest among all metrics available to estimate Project size
- The project size is estimated by counting the number of source instructions in the developed program.
- While counting the number of source instructions, the line used for commenting the code and the header lines are ignored

Lines of Code Metrics

Shortcomings of LOC metrics:

1. LOC gives a numerical value of problem size that can vary widely with individual coding *style-different programmers lay out their code in different ways.*
2. A good problem size measure should consider the overall complexity of the problem and the effort needed to solve it *.that is should consider the total efforts needed to specify, design, code, test, etc.*

Lines of Code Metrics

Shortcomings of LOC metrics:

3. LOC measure correlates poorly with the quality and efficiency of the code. *A large code size does not necessarily imply better quality or higher efficiency.*
4. LOC metric penalizes use of higher-level programming languages, code reuse etc. *if a programmer consciously uses several library routines, then the LOC count will be lower*

Example of Project Measurements

project	effort (person-month)	cost (\$)	kLOC	Doc. (pgs)	errors	people
aaa-01	24	168,000	121	365	29	3
ccc-03	62	440,000	272	1224	86	5

Example of Size-Oriented Metrics

- **Productivity** = Size / Effort
= kLOC / person-month
- **Quality** = Errors / Size
= Errors / kLOC
- **Cost** = \$ / kLOC
- **Documentation** = pages / kLOC
- Other metrics can also be developed like: errors/KLOC, page/KLOC...etc.
- Or errors/person-month, LOC/person-month, cost/page.

Function Point Metrics

- Based on “functionality” delivered by the software as the normalization value.
- Functionality is measured indirectly
- Function points (FP) measure- derived using an empirical relationship based on countable (direct) measures of software’s information domain and assessments of software complexity

Software information domain

values

- **Number of user inputs:** user inputs.
- **Number of user outputs:** reports, screens, error messages, etc.
- **Number of user inquiries:** an on-line input that generates an immediate on-line output response.
- **Number of files:** each logical (logical grouping of data) master file is counted.
- **Number of external interfaces:** all machine readable interfaces (e.g. data files on some storage media) that are used to transmit information to another system are counted.

Steps In Calculating FP

1. Count the information domain values.
2. Assess the complexity of the values.
3. Calculate the raw FP (see next table).
4. Rate the complexity factors to produce the complexity adjustment value (CAV)

$$CAV = \sum_{i=1}^{14} F_i$$

5. Calculate the adjusted FP as follows:

$$FP = \text{raw FP} \times [0.65 + 0.01 \times CAV]$$

Function Point (FP) Metrics

- Weighting Factor

<u>Parameter</u>	<u>Count</u>		<u>Simple</u>	<u>Average</u>	<u>Complex</u>		
Inputs	<input type="text"/>	x	3	4	6	=	<input type="text"/>
Outputs	<input type="text"/>	x	4	5	7	=	<input type="text"/>
Inquiries	<input type="text"/>	x	3	4	6	=	<input type="text"/>
Files	<input type="text"/>	x	7	10	15	=	<input type="text"/>
Interfaces	<input type="text"/>	x	5	7	10	=	<input type="text"/>
Count-total (raw FP)							<input type="text"/>

Rate Complexity Factors

For each complexity adjustment factor (F), give a rating on a scale of 0 to 5

0 - No influence

1 - Incidental

2 - Moderate

3 - Average

4 - Significant

5 - Essential

Complexity Adjustment Factors

$$(F_i)$$

1. Does the system require reliable backup and recovery?
2. Are data communications required?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?

Complexity Adjustment Factors

$$(F_i)$$

6. Does the system require on-line data entry?
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated on-line?
9. Are the inputs, outputs, files, or inquiries complex?

Complexity Adjustment Factors

10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

Complexity Adjustment Value

The rating for all the factors, F_1 to F_{14} , are summed to produce the complexity adjustment value (CAV)

CAV is then used in the calculation of the function point (FP) of the software

Example of Function-Oriented Metrics

Productivity = Functionality / Effort
= FP / person-month

Quality = Errors / Functionality
= Errors / FP

Cost = \$ / FP

Documentation = pages / FP

FP Characteristics

Advantages: language independent, based on data known early in project, good for estimation

Disadvantages: calculation complexity, subjective assessments, FP has no physical meaning (just a number)

Extended Function Point Metrics

Feature points

applied to systems and engineering software.
includes assessment for complex algorithms.

3D Function points

applied to real-time systems and engineered products (by Boeing)
integrates data dimension (normal FP) with functional and control dimensions