ESTIMATION FOR SOFTWARE PROJECTS

Lecture # 41

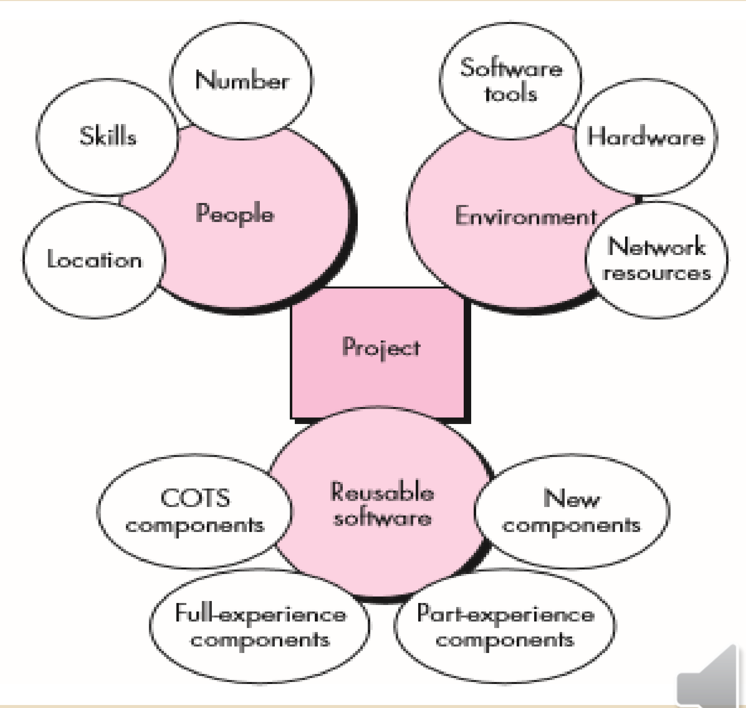
Lecture by Engr. Sidra

Software Project Planning

- The overall goal of project planning is to establish a pragmatic strategy for controlling, tracking, and monitoring a complex technical project.
- Why?
  - *So the end result gets done on time, with quality!*

Project Planning Task Set

- Establish project scope
- Determine feasibility
- Analyze risks
- Define required resources
  - *Determine require human resources*
  - *Define reusable software resources*
  - *Identify environmental resources*



Project Planning Task Set

- Estimate cost and effort
  - *Decompose the problem*
  - *Develop two or more estimates using size, function points, process tasks or use-cases*
  - *Reconcile the estimates*
- Develop a project schedule
  - *Establish a meaningful task set*
  - *Define a task network*
  - *Use scheduling tools to develop a timeline chart*
  - *Define schedule tracking mechanisms*

Estimation

- Estimation of resources, cost, and schedule for a software engineering effort requires
  - *experience*
  - *access to good historical information (metrics)*
  - *the courage to commit to quantitative predictions when qualitative information is all that exists*
- Estimation carries inherent risk, and this risk leads to uncertainty

Project Estimation

- Project scope must be understood
- Elaboration (decomposition) is necessary
- Historical metrics are very helpful
- At least two different techniques should be used
- Uncertainty is inherent in the process

Estimation Techniques

- Delay estimation until late in the project
- Past (similar) project experience
- Conventional estimation techniques
  - *task breakdown and effort estimates*
  - *size (e.g., Function Points) estimates*
- Empirical models
- Automated tools

Estimation Accuracy

- Predicated on …
  - *the degree to which the planner has properly estimated the size of the product to be built*
  - *the ability to translate the size estimate into human effort, calendar time, and cost (a function of the availability of reliable software metrics from past projects)*
  - *the degree to which the project plan reflects the abilities of the software team*
  - *the stability of product requirements and the environment that supports the software engineering effort.*

Approaches to the Sizing Problem

- "Fuzzy logic" sizing. To apply this approach, the planner must identify the type of application, establish its magnitude on a qualitative scale, and then refine the magnitude within the original range.
- Standard component sizing. Software is composed of a number of different "standard components" that are generic to a particular application area. The project planner estimates the number of occurrences of each standard component and then uses historical project data to estimate the delivered size per standard component.
- Function point sizing. The planner develops estimates of the information domain characteristics.

Approaches to the Sizing Problem

- Change sizing. This approach is used when a project encompasses the use of existing software that must be modified in some way as part of a project. The planner estimates the number and type of modifications that must be accomplished.
- The results of each of these sizing approaches be combined statistically to create a three-point or expected-value estimate. This is accomplished by developing optimistic (low), most likely, and pessimistic (high) values for size and combining them using Equation

$$S = \frac{S_{\text{opt}} + 4S_m + S_{\text{pess}}}{6}$$

Conventional Methods: LOC/FP Approach

- Compute lines of code (LOC) using estimates of information domain values
- Computes function points (FP), information domain values
- Use historical data to build estimates for the project
- Example: Computer-Aided Design application for mechanical components
  - *Apply LOC and FP to this software*

Example: LOC Approach

Function	Estimated LOC
User interface and control facilities (UICF)	2,300
Two-dimensional geometric analysis (2DGA)	5,300
Three-dimensional geometric analysis (3DGA)	6,800
Database management (DBM)	3,350
Computer graphics display facilities (CGDF)	4,950
Peripheral control function (PCF)	2,100
Design analysis modules (DAM)	8,400
Estimated lines of code	*33,200*

- Average productivity for systems of this type = 620 LOC/pm.
- Burdened labor rate =$8000 per month, the cost per line of code is approximately $13.
- Based on the LOC estimate and the historical productivity data, the total estimated project cost is 33,200*620 = $431,000 and the estimated effort is 54 person-months.

Example: FP Approach

Information domain value	Opt.	Likely	Pess.	Est. count	Weight	FP count
Number of external inputs	20	24	30	24	4	97
Number of external outputs	12	15	22	16	5	78
Number of external inquiries	16	22	28	22	5	88
Number of internal logical files	4	4	5	4	10	42
Number of external interface files	2	2	3	2	7	15
Count total						*320*

- The estimated number of FP is derived:

  FPestimated = count-total * [0.65 + 0.01 * S (Fi)]

  FPestimated = 375

organizational average productivity = 6.5 FP/pm. burdened labor rate = $8000 per month, approximately $1230/FP.

Based on the FP estimate and the historical productivity data, total estimated project cost is $461,000 and estimated effort is 58 person-months.

Example: Process-Based Estimation

Activity →	CC	Planning	Risk analysis	Engineering		Construction release		CE	Totals
Task →				Analysis	Design	Code	Test		
Function									
↓									
UICF				0.50	2.50	0.40	5.00	n/a	8.40
2DGA				0.75	4.00	0.60	2.00	n/a	7.35
3DGA				0.50	4.00	1.00	3.00	n/a	8.50
CGDF				0.50	3.00	1.00	1.50	n/a	6.00
DBM				0.50	3.00	0.75	1.50	n/a	5.75
PCF				0.25	2.00	0.50	1.50	n/a	4.25
DAM				0.50	2.00	0.50	2.00	n/a	5.00
Totals	0.25	0.25	0.25	3.50	20.50	4.50	16.50		46.00
% effort	1%	1%	1%	8%	45%	10%	36%		

- Based on an average burdened labor rate of \$8,000 per month, the total estimated project cost is \$368,000 and the estimated effort is 46 person-months.

Estimation with Use-Cases

	use cases	scenarios	pages	scenarios	pages	LOC	LOC estimate
User interface subsystem	6	10	6	12	5	560	3,366
Engineering subsystem group	10	20	8	16	8	3100	31,233
Infrastructure subsystem group	5	6	5	10	6	1650	7,970
Total LOC estimate							42,568

- Using 620 LOC/pm as the average productivity for systems of this type and a burdened labor rate of $8000 per month, the cost per line of code is approximately $13. Based on the use-case estimate and the historical productivity data, the total estimated project cost is $42,568*13 = $552,000 and the estimated effort is 68 person-months.

Empirical Estimation Models

$$E = A + B \times (e_v)^C$$

- where A, B, and C are empirically derived constants,
- E is effort in person-months, and ev is the estimation variable (either LOC or FP).
- E to be adjusted by other project characteristics (e.g., problem complexity, staff experience, development environment).
- Among the many LOC-oriented estimation models proposed in the literature are

$E = 5.2 \times (\text{KLOC})^{0.91}$	Walston-Felix model
$E = 5.5 + 0.73 \times (\text{KLOC})^{1.16}$	Bailey-Basili model
$E = 3.2 \times (\text{KLOC})^{1.05}$	Boehm simple model
$E = 5.288 \times (\text{KLOC})^{1.047}$	Doty model for KLOC > 9

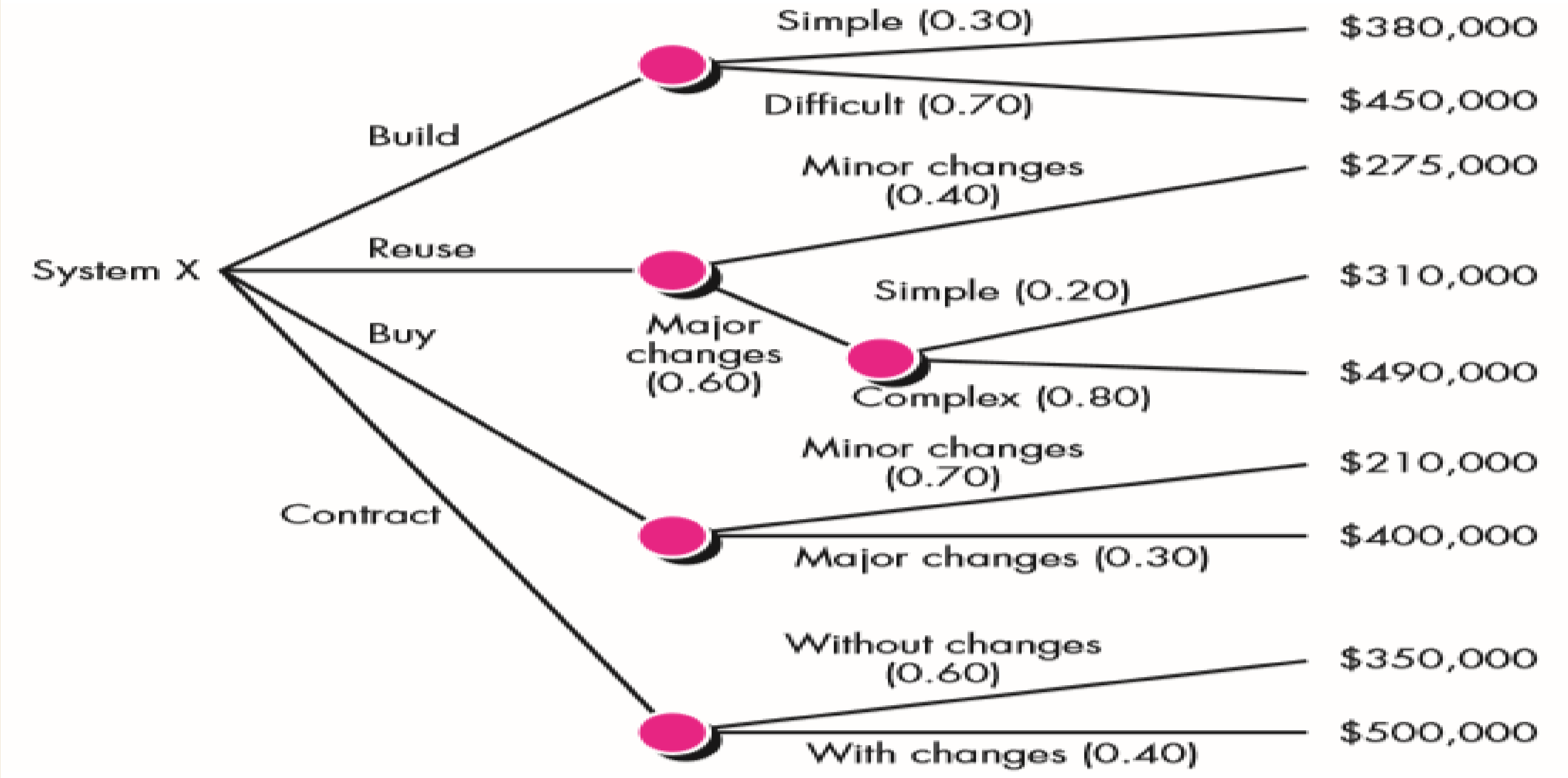
- FP-oriented models have also been proposed. These include

$E = -91.4 + 0.355\ \text{FP}$	Albrecht and Gaffney model
$E = -37 + 0.96\ \text{FP}$	Kemerer model
$E = -12.88 + 0.405\ \text{FP}$	Small project regression model

COCOMO-II

- COCOMO II is actually a hierarchy of estimation models that address the following areas:
  - *Application composition model. Used during the early stages of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount.*
  - *Early design stage model. Used once requirements have been stabilized and basic software architecture has been established.*
  - *Post-architecture-stage model. Used during the construction of the software.*
- Empirical model that relies on size information

The Make-Buy Decision



Computing Expected Cost

expected cost =

$$\sum (\text{path probability})_i \times (\text{estimated path cost})_i$$

For example, the expected cost to build is:

$$\text{expected cost}_{build} = 0.30\ (\$380K) + 0.70\ (\$450K)$$
$$= \$429\ K$$

similarly,

$$\text{expected cost}_{reuse} = \$382K$$

$$\text{expected cost}_{buy} = \$267K$$

$$\text{expected cost}_{contr} = \$410K$$