投票 最多可选1项

设置

Are you ready?

A Yes

B No

提交

Review – Product/Process/Project Metrics

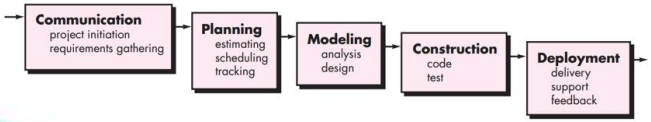
- Product: size(LOC/KLOC , FP)
- Process:
  - defects per FP/KLOC
  - pages of documentation per FP/KLOC
- Project:
  - Effort per SE task
  - Errors uncovered per review hour
- Quality:
  - Quality model
  - DRE $\mathrm{DRE} = \frac{E}{E + D}$ $\mathrm{DRE}_i = \frac{E_i}{E_i + E_{i+1}}$

Software Engineering

Part 4
Project Management

Chapter 33
Estimation for Software Projects

Review-1



most important

People

Process

Product

Project

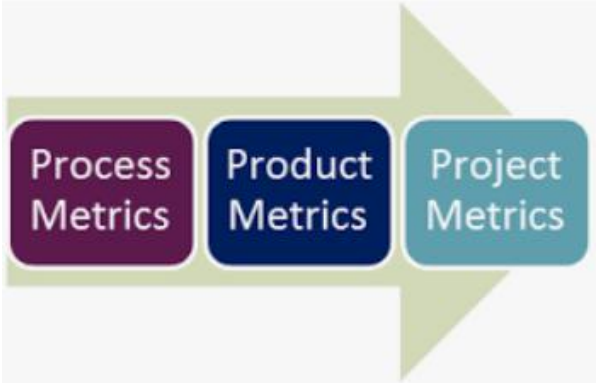
Activity & Action & Task

the software to be built

all work required to make the product a reality

Review-2

- **process metrics**
  Quality-related, Productivity-related, Statistical SQA data, DRE, Reuse
- **product metrics**
  LOC, Function-Based Metrics, Specification Quality System complexity, CK Metrics, Halstead, Software Maturity Index
- **project metrics**
  Effort/time per software engineering task, Errors uncovered per review hour, Distribution of effort on SE tasks, Scheduled vs. actual milestone dates



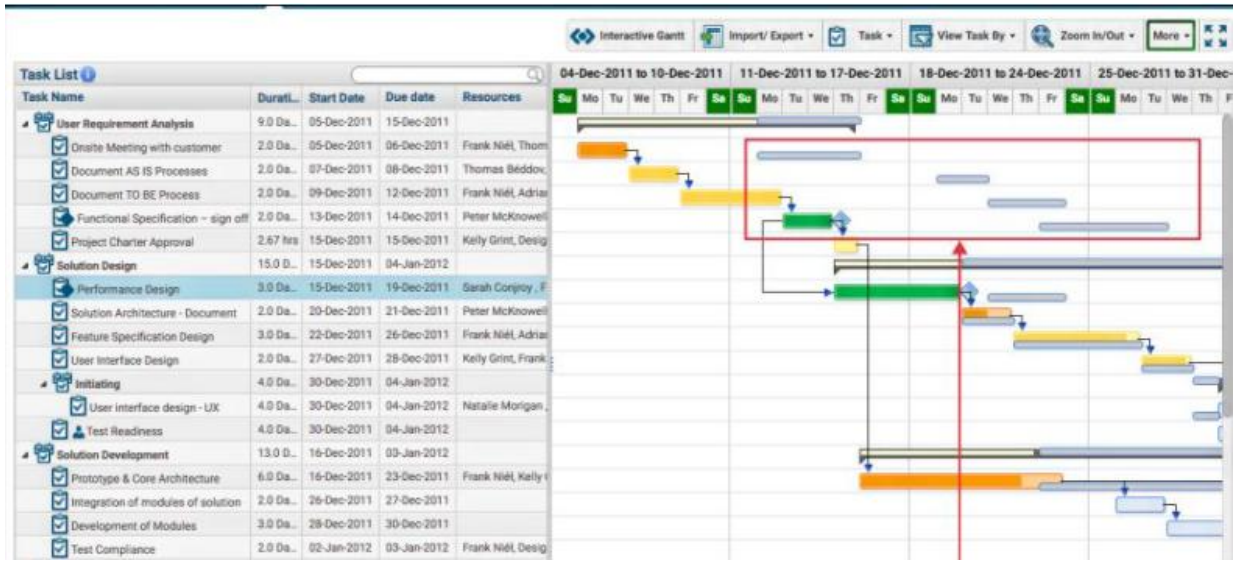
Many of the same metrics are used in both the process and project.

Contents

Contents

Question

- If you are going to make the plan for a software project, what points will you consider?



33.1 Observations on Estimation

- Estimation:
  - Resource, cost and schedule

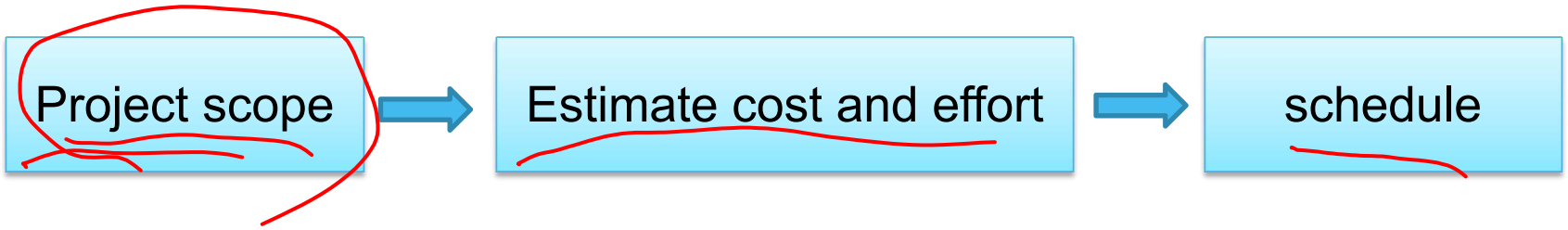
- Estimation risk:
  - ✓Project complexity
  - ✓Project size
  - ✓The degree of structural uncertainty

33.2 The Project Planning Process

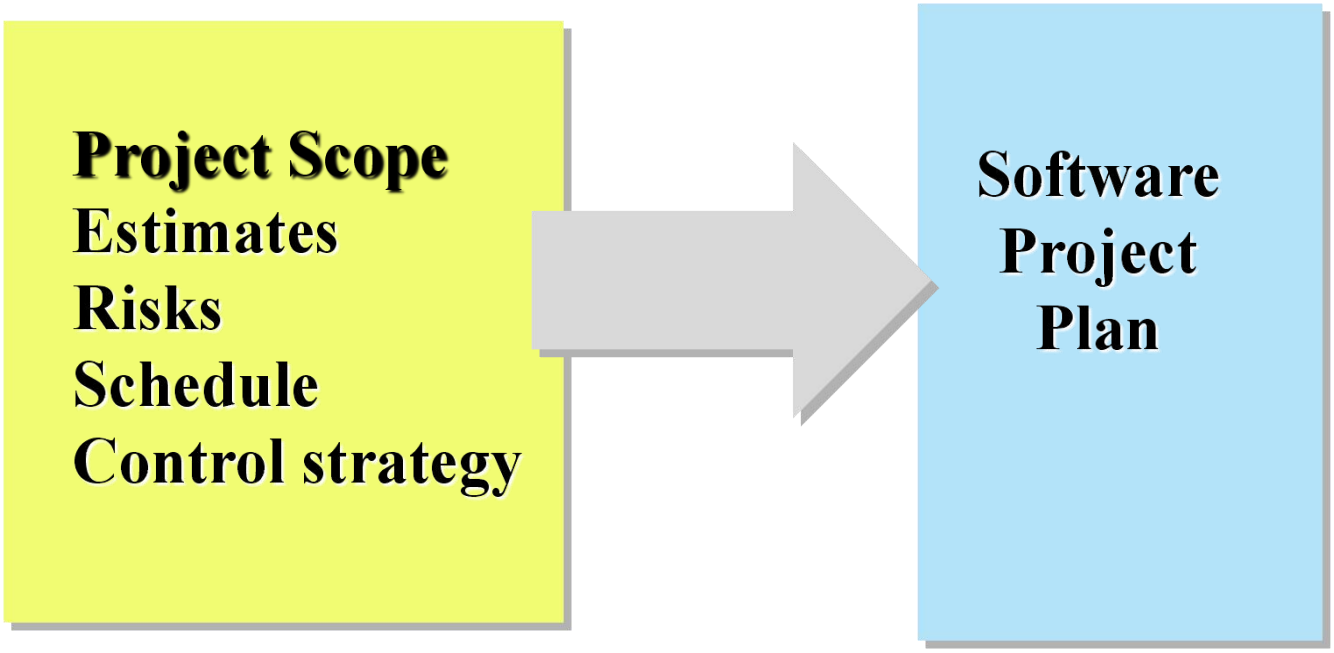
TASK SET

Task Set for Project Planning

1. Establish project scope.
2. Determine feasibility.
3. Analyze risks (Chapter 35).
4. Define required resources.
   a. Determine required human resources.
   b. Define reusable software resources.
   c. Identify environmental resources.
5. Estimate cost and effort.
   a. Decompose the problem.
   b. Develop two or more estimates using size, function points, process tasks, or use cases.
   c. Reconcile the estimates.
6. Develop a project schedule (Chapter 34).
   a. Establish a meaningful task set.
   b. Define a task network.
   c. Use scheduling tools to develop a time-line chart.
   d. Define schedule tracking mechanisms.



33.2 Write it Down!



33.3 What is Scope?

- *Software scope* describes
  - the functions and features that are to be delivered to end-users
  - the data that are input and output
  - the "content" that is presented to users as a consequence of using the software
  - the performance, constraints, interfaces, and reliability that *bound* the system.

33.3 What is Scope?

- *Software scope* sample

(*a short description of the software*)

1.2. Scope

The Jacksonville State University Computing and Information Sciences Web Accessible Alumni Database (CISWAAD) is designed to run on the departmental server and to allow alums to fill out a survey form, create a new database entry, update an existing database entry, or contact another alum. The data will be held in an Access database on the departmental server.

Web Based Integrated Development Environment

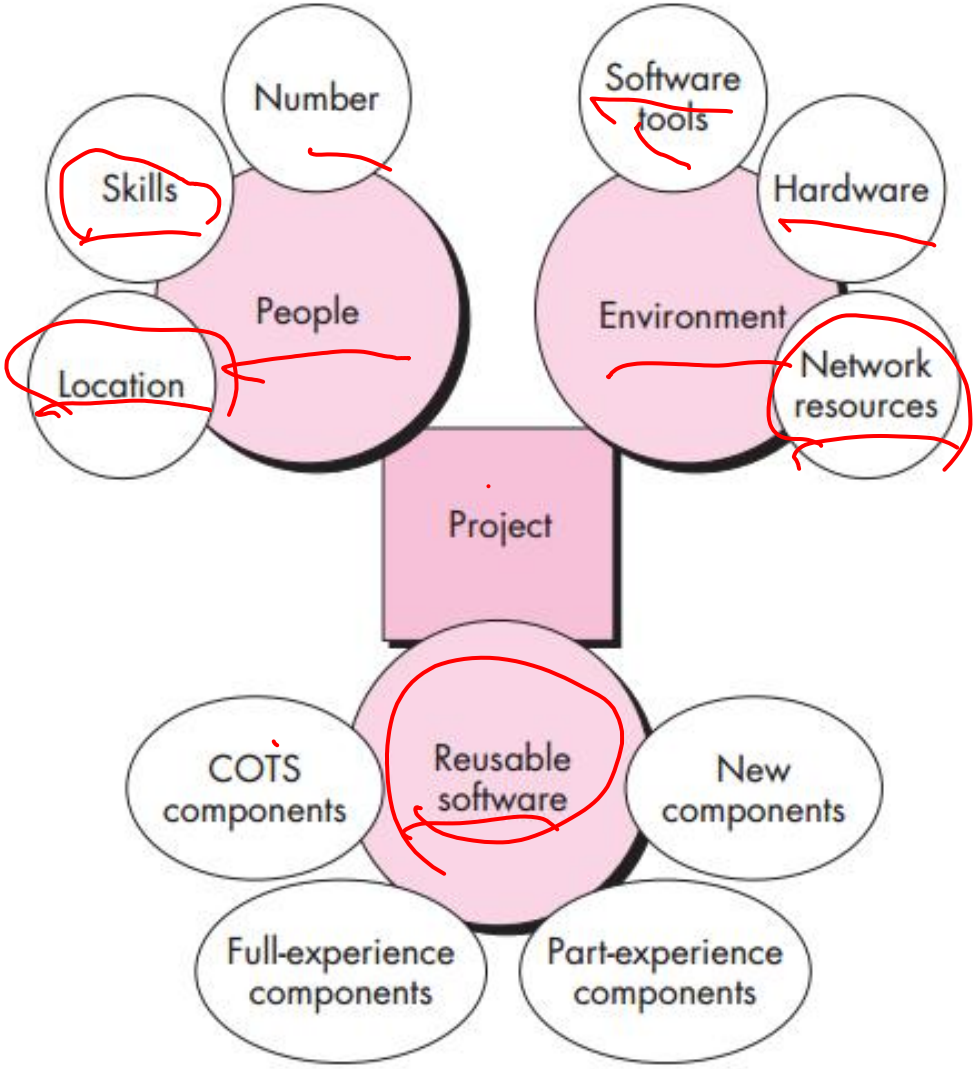
33.3 What is Scope?

- *Software scope* sample – project management

1.2 Scope

- ✓ Create different users with varied roles and scopes.
- ✓ Confirm each member by providing activation codes.
- ✓ Manage all project details like tasks, deadlines, team members and resources.
- ✓ Assign different tasks to different members.
- ✓ Provide documentation to the members about the tasks being added
- ✓ Update all members about new proceedings in the project.
- ✓ Bind all the information provided by the team members at one place and show it to all others.
- ✓ Maintain start date and end date of each task
- ✓ Maintain the overall timeline of the project.

33.4 Resources



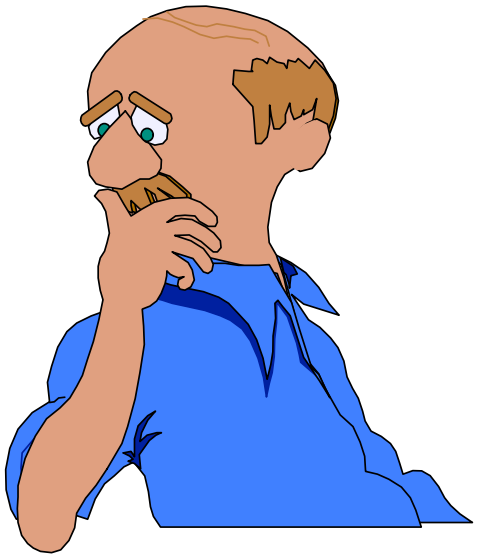
Define required resources

33.5 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

33.5 Estimation Techniques

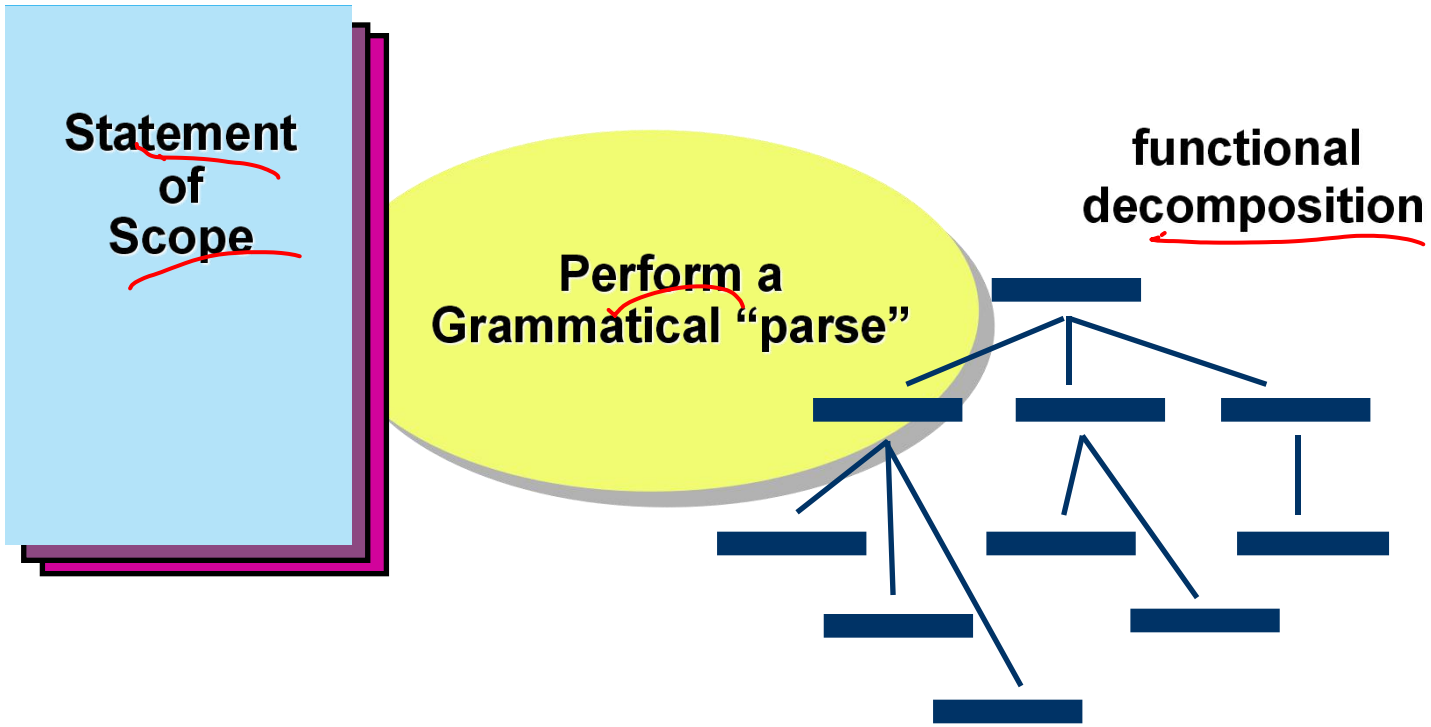
- Past (similar) project experience
- Conventional estimation techniques
  - task breakdown and effort estimates
  - size (e.g., FP) estimates
- Empirical models
- Automated tools



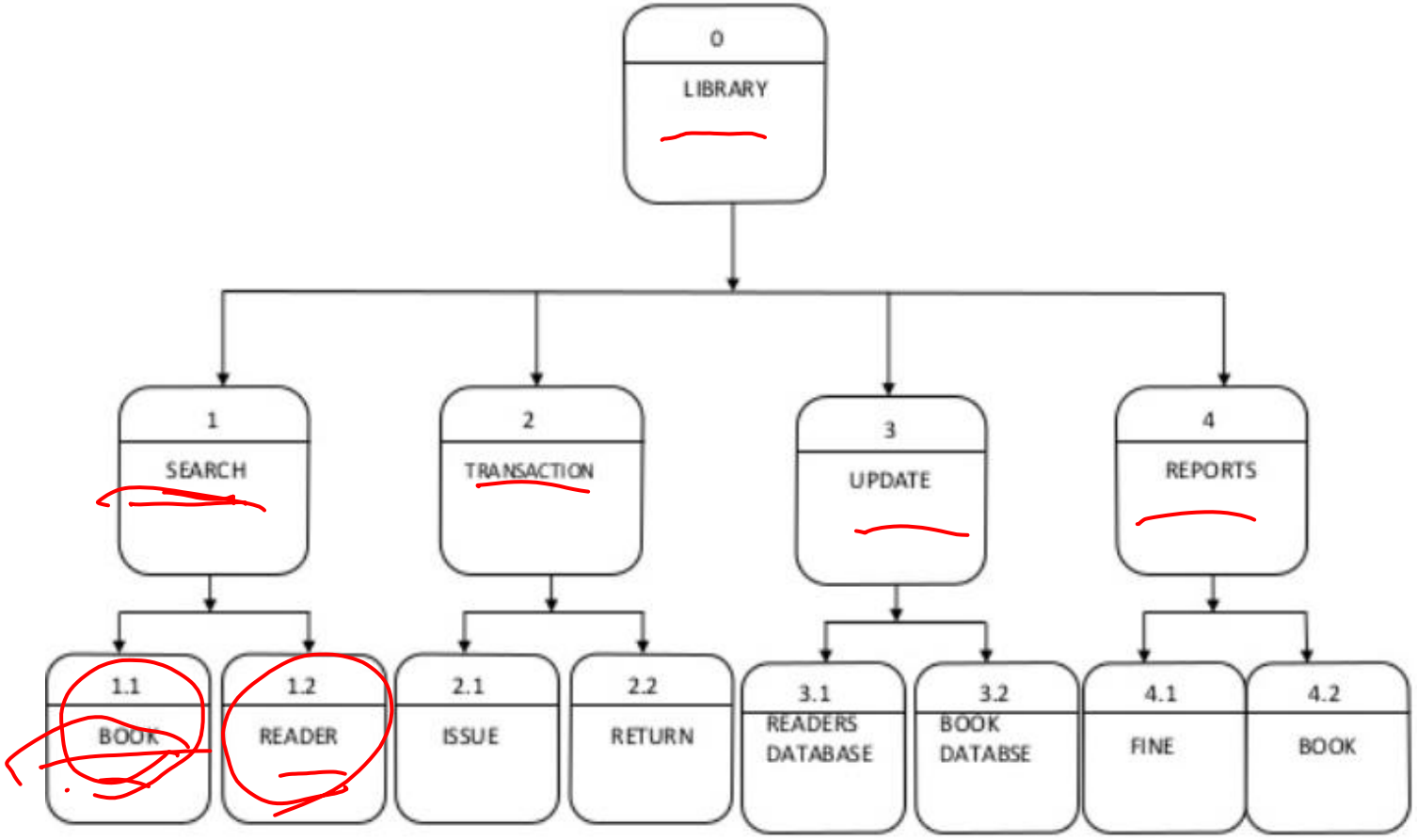
33.5 Estimation Accuracy

- 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 dollars
  - 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.

33.6 Functional Decomposition



33.6 Functional Decomposition



https://www.slideshare.net/ashu6/library-management-system-6029783

33.6.1 LOC/FP Approach

- compute LOC/FP using estimates of information domain values

- use historical data to build estimates for the project

33.6.2 Problem-based Estimation

- LOC/FP

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

Sopt: the optimistic estimates
Sm: the most likely estimates
Spess: the pessimistic estimates

Any estimation technique, no matter how sophisticated, must be cross-checked with another approach.

33.6.3 Example: LOC Approach (breakdown)

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 (8000/620 = 12.9).

Based on the LOC estimate and the historical productivity data,

- **the estimated project cost is ?** **$431,000 (33200*13=431,600)**
- **the estimated effort is ?** **54 person-months (33,200/620=54)**

33.6.4 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

Factor	Value
Backup and recovery	4
Data communications	2
Distributed processing	0
Performance critical	4
Existing operating environment	3
Online data entry	4
Input transaction over multiple screens	5
Master files updated online	3
Information domain values complex	5
Internal processing complex	5
Code designed for reuse	4
Conversion/installation in design	3
Multiple installations	5
Application designed for change	5
Value adjustment factor	**1.17**

The estimated number of FP is derived:

$$FP_{estimated} = \text{count total} \times [0.65 + 0.01 \times \Sigma(F_i)] = 375$$

Given:

organizational average productivity = 6.5 FP/pm.

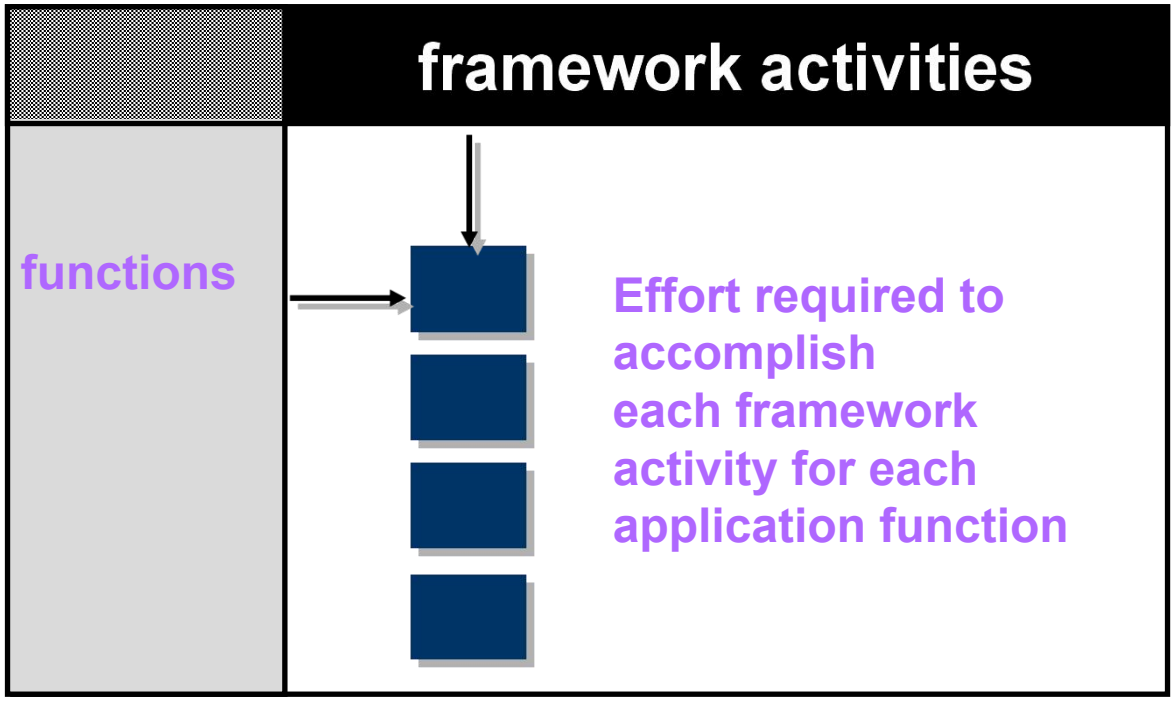
burdened labor rate = $8000 per month, approximately $1230/FP.

Then:

- **total estimated project cost is $461,000 (1230*375=461,250).**
- **estimated effort is 58 (374/6.5=58) person-months.**

33.6.5 Process-Based Estimation

Obtained from "process framework"



33.6.6 Process-Based Estimation Example

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%		

CC = customer communication CE = customer evaluation

man-month

Given: an average burdened labor rate of $8,000 per month,

Then:

- **the estimated effort is 46 person-months.**
- **the total estimated project cost is $368,000**

Take a break

Five minutes

33.7.1 Empirical Estimation Models

General form --- regression analysis

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

- Effort: person-months
- either a constant or a number derived based on complexity of project
- LOC or FP
- empirically derived

estimation models

$E = 5.2 \times (\text{KLOC})^{0.91}$	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
$E = -91.4 + 0.355\ \text{FP}$	Albrecht and Gaffney model
$E = -37 + 0.96\ \text{FP}$	Kemerer model
$E = -12.88 + 0.405\ \text{FP}$	Small project regression model

33.7.2 COCOMO-II

- COCOMO II is actually a hierarchy of estimation models that address the following areas:
  1. *Early prototype model.* When prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount.
  2. *Early design stage model.* Used once requirements have been stabilized and basic software architecture has been established.
  3. *Post-architecture-stage model.* Used during the construction of the software.

33.7.2 COCOMO-II

Model 1

Early Prototyping Level

- Supports prototyping projects and projects where there is extensive reuse
- Estimates effort in object points/staff month
- PM = (NOP × (1 - %reuse/100)) / PROD, where:
  - PM is the effort in person-months
  - NOP is the number of object points
  - PROD is the productivity

33.7.2 COCOMO-II

Model 2

Early Design Level

- Estimates made after requirements confirmed

$$PM = A \times Size^{B} \times M + \sum_{i=1}^{n} EM_i$$

where:

- A = 2.5 in initial calibration
- Size in KLOC
- B varies from 1.1 to 1.24 depending on novelty of project, development flexibility, risk management approaches, and process maturity
- EM = (ASLOC × (AT / 100)) / ATPROD
- M = PERS × RCPX × RUSE × PDIF × PREX × FCIL × SCED

33.7.2 COCOMO-II

M = RCPX * RUSE * PDIF * PREX * PERS * SCED * FCIL

Multipliers

Multipliers reflect capability of developers, non-functional requirements, familiarity with development platform, etc.

RCPX - product reliability and complexity
RUSE - the reuse required
PDIF - platform difficulty
PREX - personnel experience
PERS - personnel capability
SCED - required schedule
FCIL - the team support facilities

33.7.2 COCOMO-II

Model 3

Post-Architecture Level

- Uses same formula as early design estimates
- Estimate of size adjusted to account for:
  - Requirements volatility
  - Rework required to support change
  - Extent of possible reuse

33.7.2 COCOMO-II

Post-Architecture Level (continued)

ESLOC = ASLOC x (AA + SU +0.4DM + 0.3CM + 0.3IM) / 100

- ESLOC is equivalent number of lines of new code
- ASLOC is the adjusted number of lines of reusable code which must be modified
- DM is the % of design modified
- CM is the % of the code that is modified
- IM is the % of the original integration effort required for integrating the reused software
- SU is a factor based on the cost of software understanding
- AA is a factor which reflects the initial assessment costs of deciding if software may be reused

33.7.3 The Software Equation

A dynamic multivariable model

$$E = \frac{\text{LOC} \times B^{0.333}}{P^3} \times \frac{1}{t^4}$$

where

E = effort in person-months or person-years

LOC = size of software (independent)

t = project duration in months or years (independent)

B = "special skills factor"

P = "productivity parameter"

33.8 Estimation for OO Projects

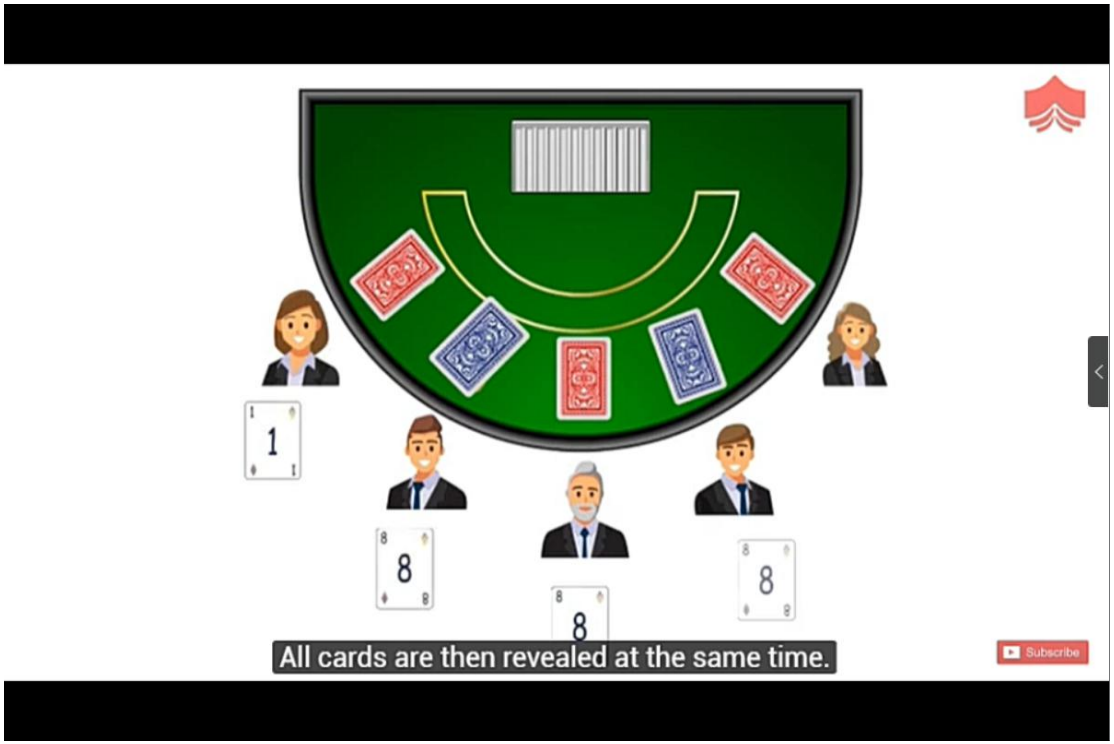
1. Develop estimates using effort decomposition, FP analysis, and any other method that is applicable for conventional applications.
2. Using object-oriented requirements modeling, develop use-cases and determine a count.
3. From the analysis model, determine the number of key classes .
4. Categorize the type of interface for the application and develop a multiplier for support classes.

Interface Type	Multiplier
No GUI	2.0
Text-based user interface	2.25
GUI	2.5
Complex GUI	3.0

5. Multiply the total number of classes (key + support) by the average number of work units per class. (Lorenz and Kidd suggest :15 to 20 person-days per class)
6. Cross-check the class-based estimate by multiplying the average number of work units per use case.

33.9 Estimation for Agile Development

Let's watch a video!



https://www.youtube.com/watch?v=TxSzo3lwwWQ

Summary

- Breakdown（size-based）
- COCOMOII
- Agile (planning poker)

Practice

1. Project training: Use the estimation techniques developed in this chapter to estimate the effort for your project.

2. Academic training: ICSE2021(43rd International Conference on Software Engineering): Skim the table of content and share one paper's abstract with us.

   https://conf.researchr.org/track/icse-2021/icse-2021-papers

THE END