Estimation for Software Projects

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-I

- Establish project scope
- Determine feasibility
- Analyze risks
  - Risk analysis is considered in detail Later.
- Define required resources
  - Determine require human resources
  - Define reusable software resources
  - Identify environmental resources

Project Planning Task Set-II

- 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
  - Scheduling is considered in detail Later.
    - Establish a meaningful task set
    - Define a task network
    - Use scheduling tools to develop a timeline chart
    - Define schedule tracking mechanisms

Estimation

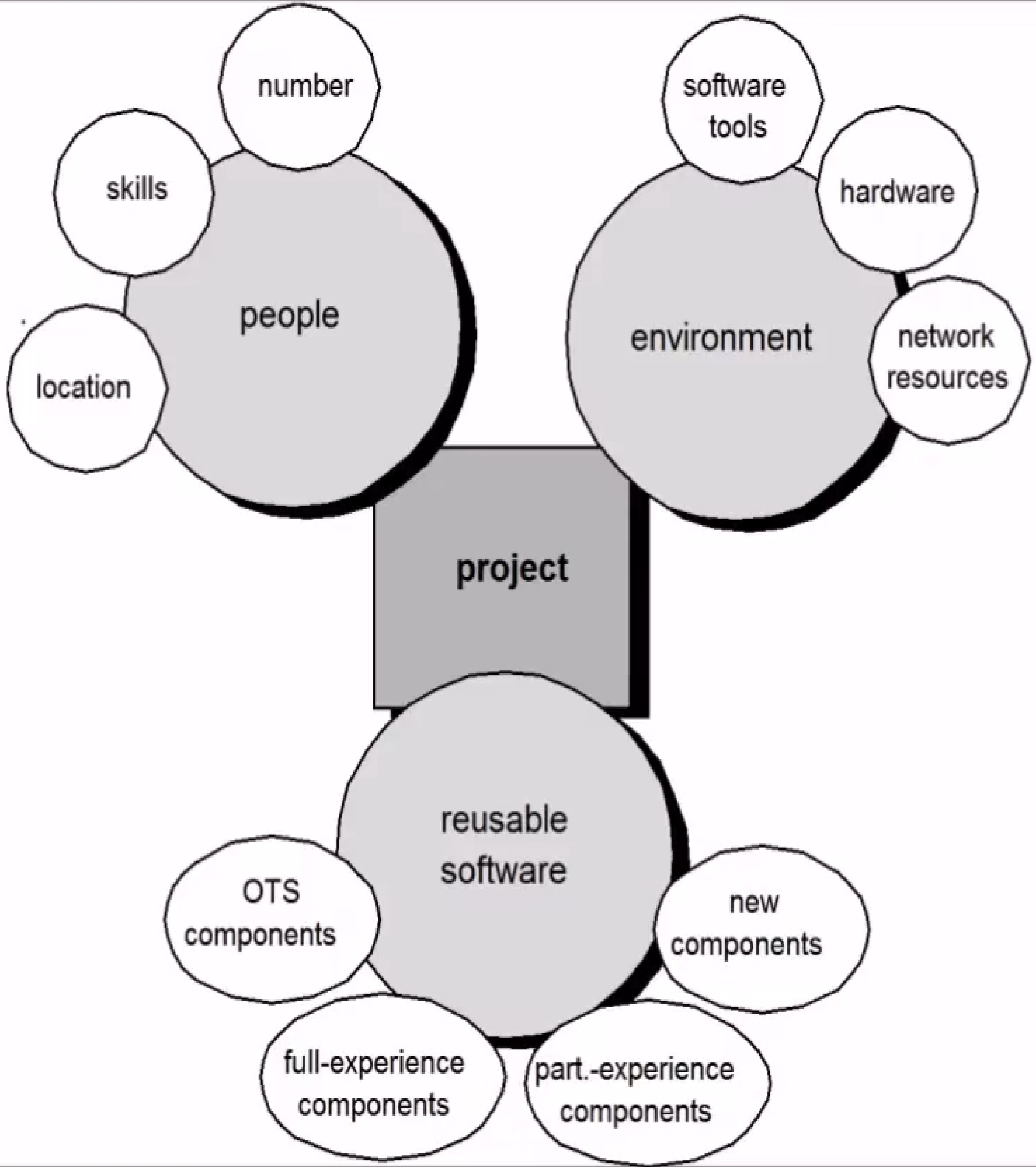
"Good estimating approaches and solid historical data offer the best hope that reality will win out over impossible demands."

- 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 complexity, project size, and the degree of structural uncertainty all affect the reliability of estimates.

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.
- Scope is defined using one of two techniques:
  - A narrative description of software scope is developed after communication with all stakeholders.
  - A set of use-cases is developed by end-users.

Resources



OTS->Off the shelf

Software Project Estimation

To achieve reliable cost and effort estimates, a number of options arise:

1. Delay estimation until late in the project (obviously, we can achieve 100 percent accurate estimates after the project is complete!).
2. Base estimates on similar projects that have already been completed.
3. Use relatively simple decomposition techniques to generate project cost and effort estimates.
4. Use one or more empirical models for software cost and effort estimation.

Estimation Techniques

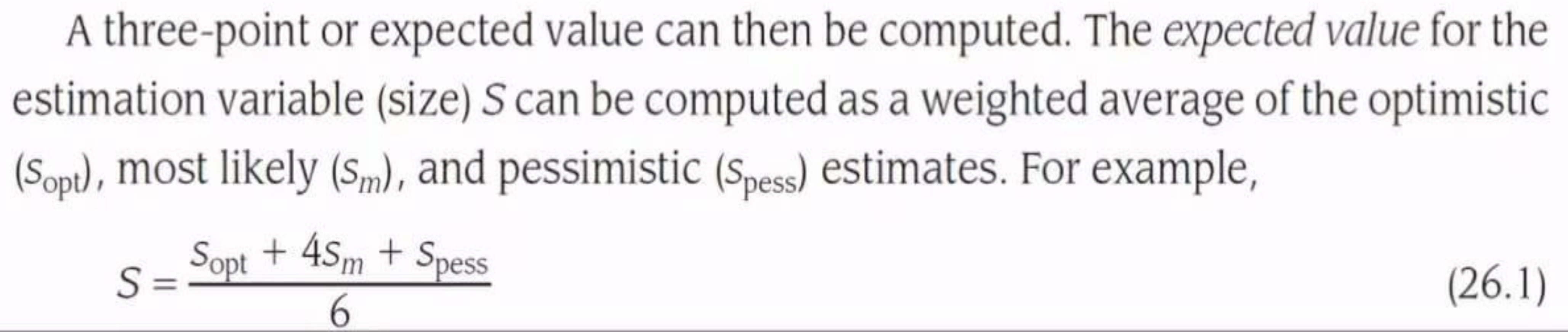
- Past (similar) project experience
- Conventional estimation techniques
  - Task breakdown and effort estimates
  - Size (e.g., FP) estimates
- Empirical models
- Automated tools

Decomposition Techniques
Software Sizing

- The accuracy of a software project estimate is predicated on a number of things:
  - 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 (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.

Problem-Based Estimation

- lines of code and function points were described as measures from which productivity metrics can be computed. LOC and FP data are used in two ways during software project estimation:
- (1) as estimation variables to “size” each element of the software and
- (2) as baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.



A three-point or expected value can then be computed. The *expected value* for the estimation variable (size) S can be computed as a weighted average of the optimistic (s_{opt}), most likely (s_m), and pessimistic (s_{pess}) estimates. For example,

$$S = \frac{s_{\text{opt}} + 4s_m + s_{\text{pess}}}{6} \qquad (26.1)$$

Conventional Methods:LOC/FP Approach

- Compute LOC/FP using estimates of information domain values
- Use historical data to build estimates for the project

An Example of LOC-Based Estimation

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

Estimation table for the LOC methods

An Example of FP-Based Estimation

Estimation table for the LOC methods

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

An Example of FP-Based Estimation

EAF

Estimation table for the LOC methods

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

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