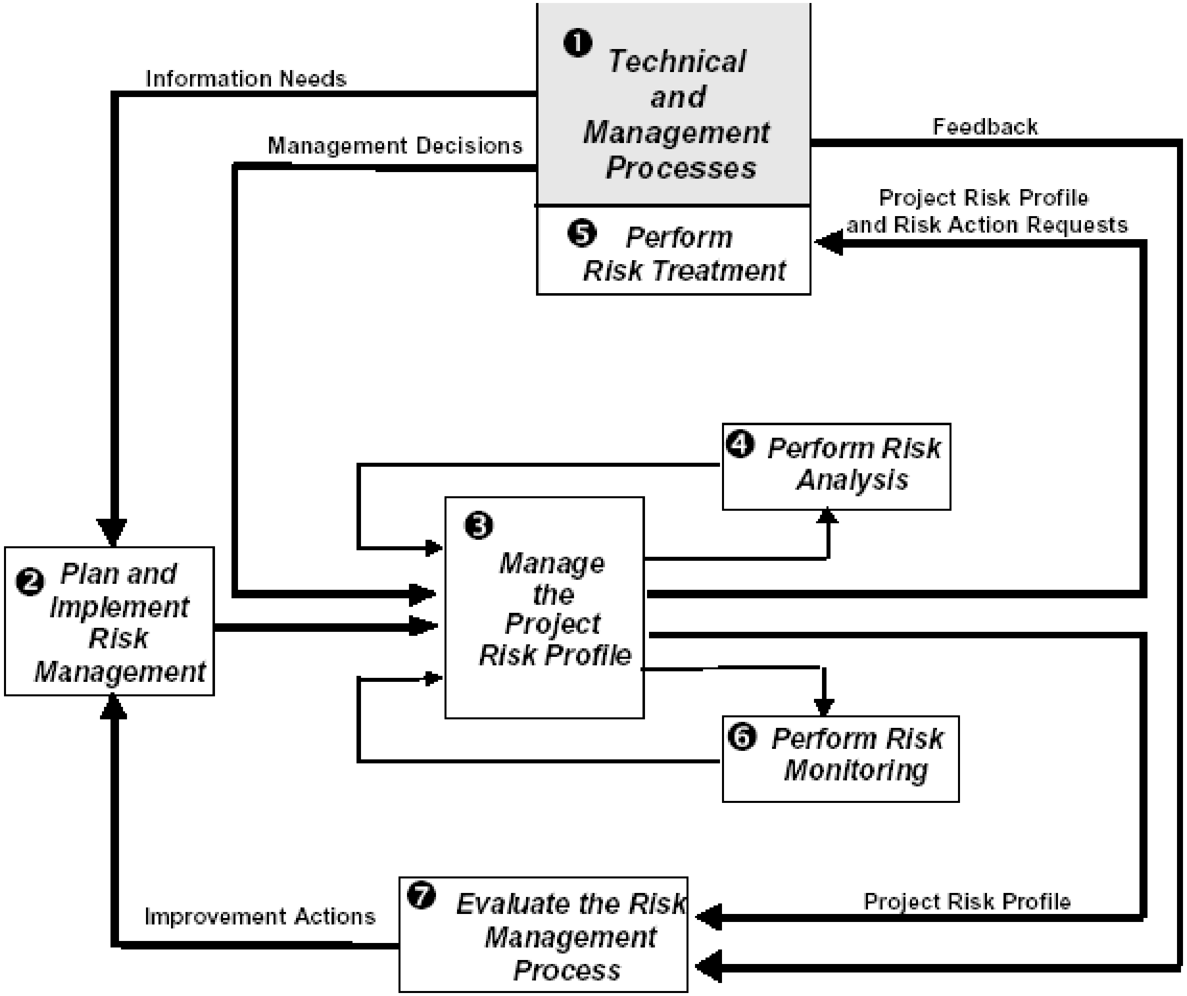
Software Project Management

Lecture 16

Last Lecture Summary


❶ Technical and Management Processes
❺ Perform Risk Treatment
Information Needs
Management Decisions
Feedback
Project Risk Profile and Risk Action Requests
❹ Perform Risk Analysis
❸ Manage the Project Risk Profile
❷ Plan and Implement Risk Management
❻ Perform Risk Monitoring
❼ Evaluate the Risk Management Process
Improvement Actions
Project Risk Profile

IEEE 1540:2001 – Standard for Software Life Cycle Processes - Risk Management

- **Describes a process for the management of risk during software acquisition, supply, development, operations and maintenance.**

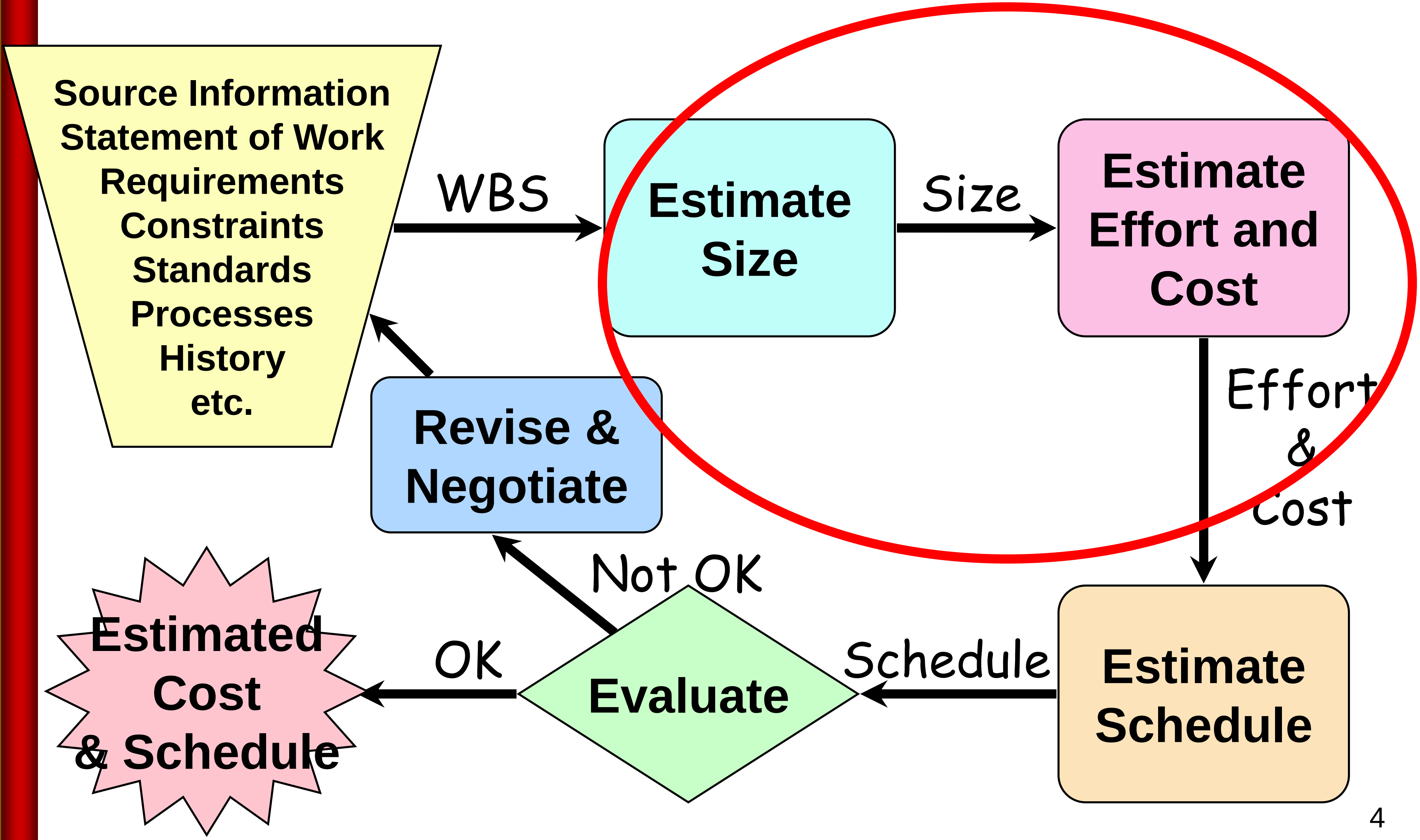
Source:
IEEE Standard 1540:2001
© IEEE 2001.
All rights reserved.

Words of Wisdom

- **Accept the total responsibility of your choices and actions.**

- *Every soul will be (held) in pledge for its deeds. Al-Qur'an (74:38)*

Planning - Estimating



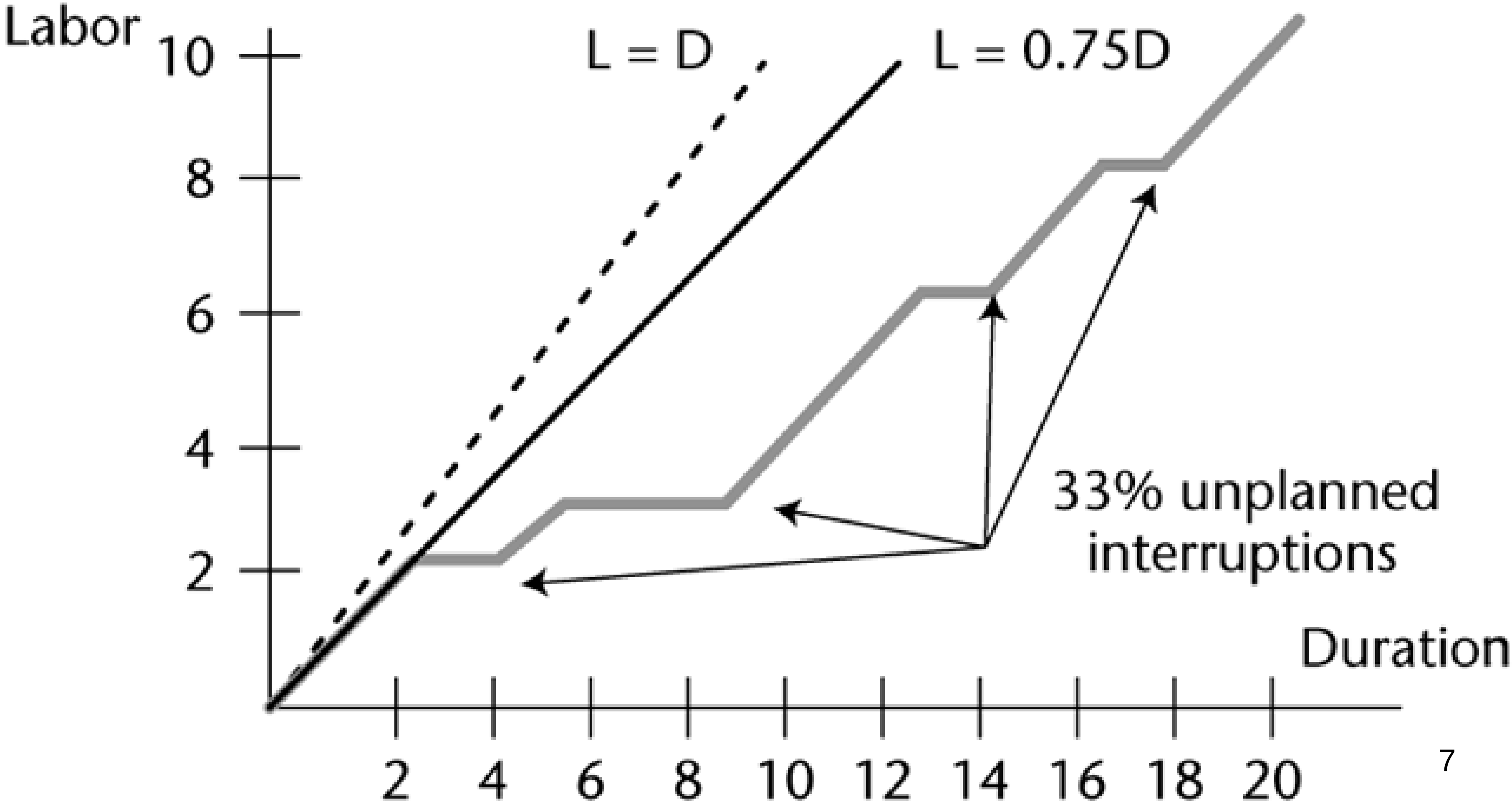
Estimation

- The process of forecasting/predicting, or approximating the amount of
  - Effort
  - Cost
  - Duration
- required for completing project and/or project deliverables.

Duration vs. Effort vs. Productive Time

- Duration is the elapsed time in business working days
- Work effort is the labor required to complete an activity.
  - Work effort is typically the amount of focused and uninterrupted labor time required to complete an activity.
- Productive time considers the percentage of the work day that can be devoted to project activity work.
  - Estimates in software range from 66-75%, recent estimates of about 50-65% (same client base). This doesn’t include unexpected interruptions!

Elapsed time vs. work time



Why Estimation?

- The most common problem in building the software systems is not the construction of them itself, but rather the estimation of the cost of that construction [Robert L Glass (Software Runaways)]
- According to the Standish Group CHAOS report of 2003:
  - 15% of software projects are terminated before they produce anything
  - 66% are considered to have failed
  - Of those that do complete the average cost blowout is 43%
- "On average, Government projects are being delivered 37% later than their estimate and 22% over their cost estimate." ISBS Group report of January 2004 on Government projects in Australia.

Why Estimation ?

Project	First ; Last Estimates		Status at Completion
	Cost ($M)	Schedule (months)	
PROMS (Royalty Collection)	12 ; 21+	22 ; 46	Canceled, Month 28
London Ambulance	1.5 ; 6+	7 ; 17+	Canceled, Month 17
London Stock Exchange	60-75 ; 150	19 ; 70	Canceled, Month 36
Confirm (Travel Reservation)	56 ; 160+	45 ; 60+	Canceled, Month 48
Master Net (banking)	22 ; 80+	9 ; 48+	Canceled, Month 48

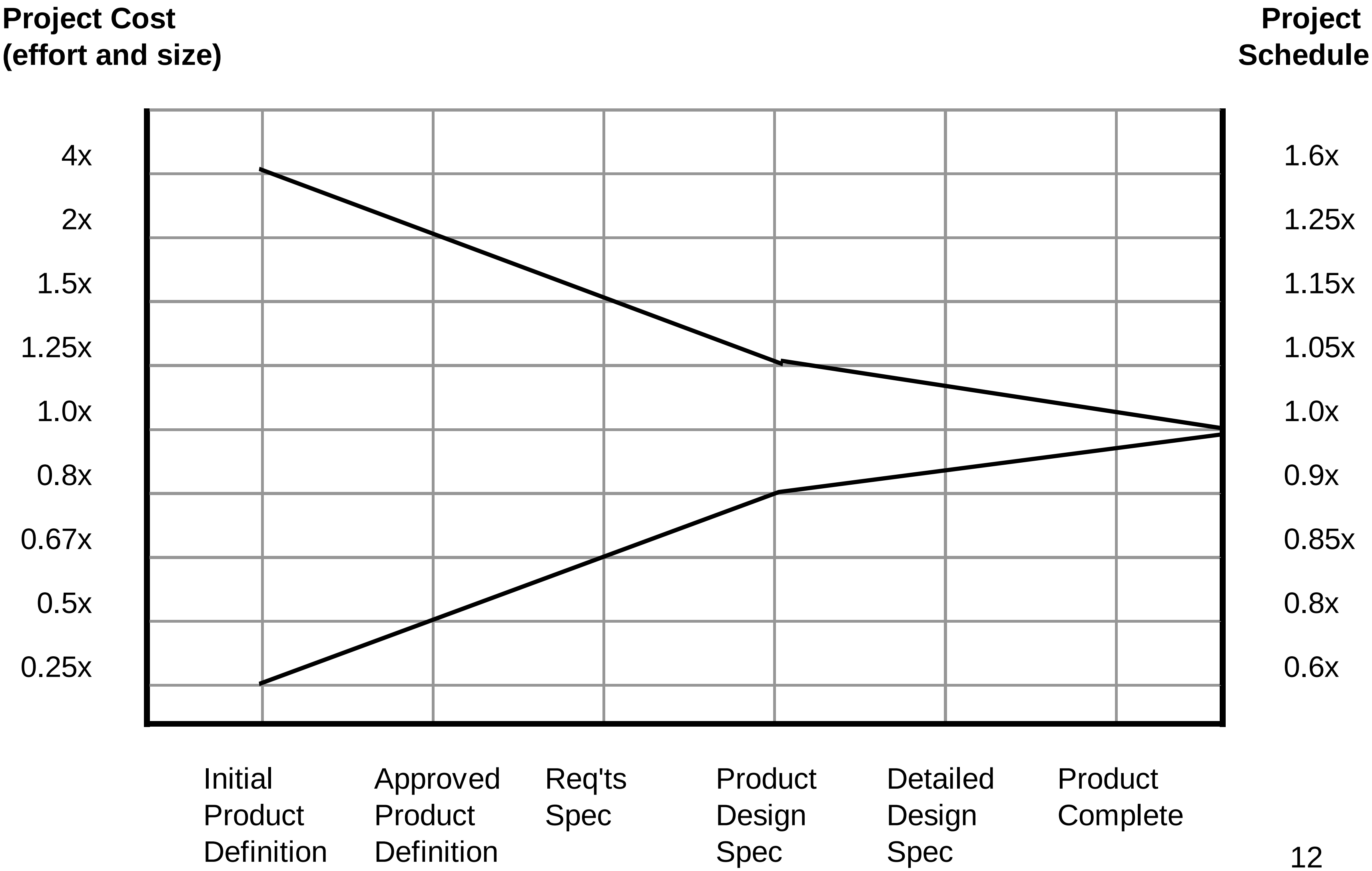
Why Estimate ?

- To determine if planned features are realistic in terms of
  - Resources (effort)
  - Schedule, and
  - Cost
- To determine staff loading during phases
- To be competitive
- To allow project monitoring and controlling

When to Estimate?

- Throughout the software lifecycle
  - Initial estimates
  - Detailed estimates
  - Re estimates
- Software cost estimation is a process of gradual refinement

Cone of Uncertainty



Types of Estimates

- Top-down (macro) estimates
  - E.g., Analogy, Group Consensus
- Bottom-up (micro) estimates
  - E.g., Estimates of elements of the work breakdown structure

Conditions for Preferring Top-Down or Bottom-up Time and Cost Estimates

Condition	Macro Estimates	Micro Estimates
Strategic decision making	X	
Cost and time important		X
High uncertainty	X	
Internal, small project	X	
Fixed-price contract		X
Customer wants details		X
Unstable scope	X	

Problems in Estimation

- Complexity and Uniqueness of Software Development Project
- Imprecise estimation models and methods
- Ineffective management of requirements
- In-experienced project management
- Lack of application domain expertise
- Software processes and process maturity
- Lack of historical data

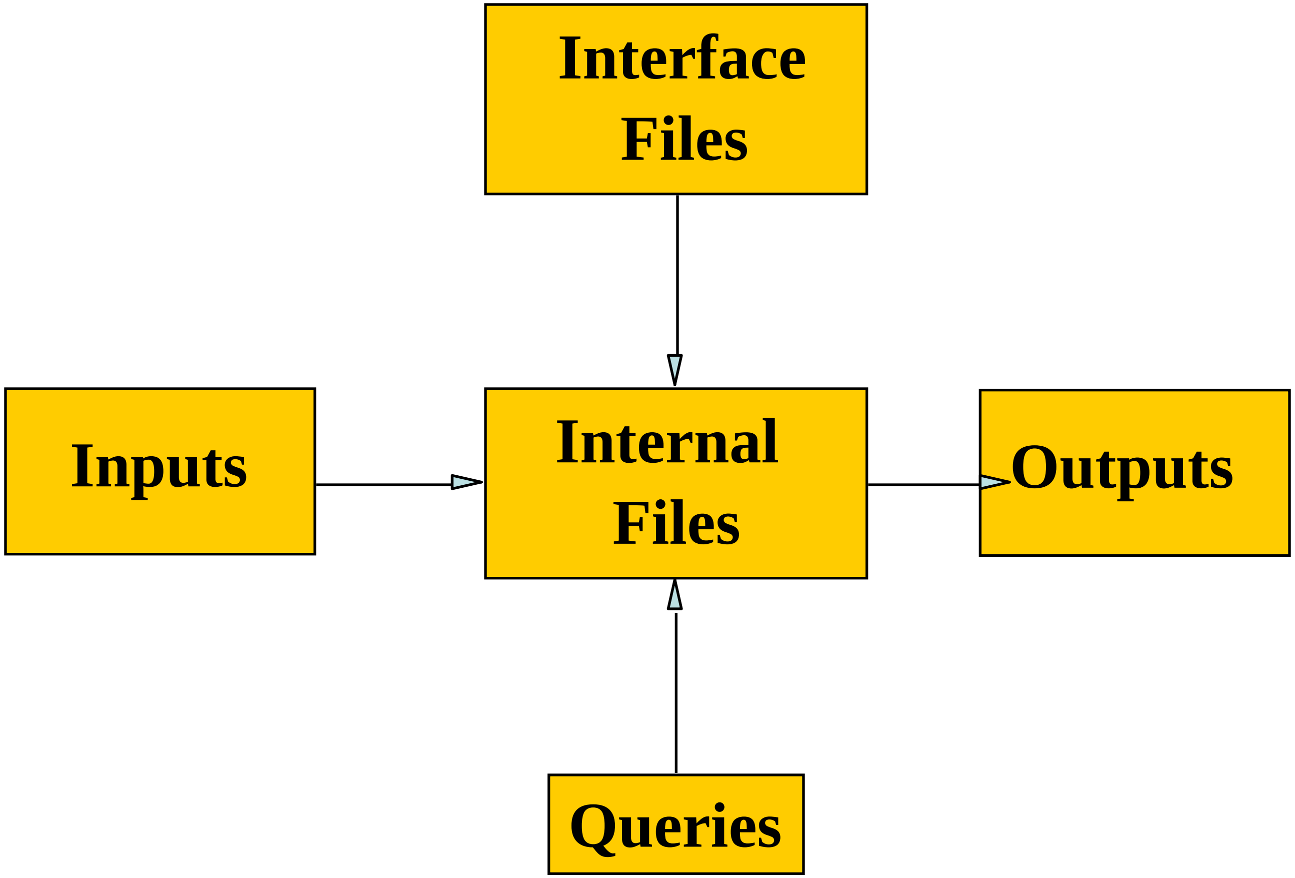
Size Estimation

- Metric used in estimating the size of software product are
  - LOC, function points, …
- Notion of size changes in project from requirement analysis to implementation and test
  - at requirements time
  - at design time
  - at implementation/test time
- Size can be estimated
  - Using an algorithmic approach
  - Using old similar project's size as a basis
  - Using expert judgment

Function Points

- It was developed by A.J. Albrecht of the IBM Corporation in the early 1980s (1979, 1983).
- Economic output unit of software projects should be valid for all languages
- The size is measured from a functional, or user, point of view
- Used at design stages of a project

Systems View (FP perspective)



Interface Files

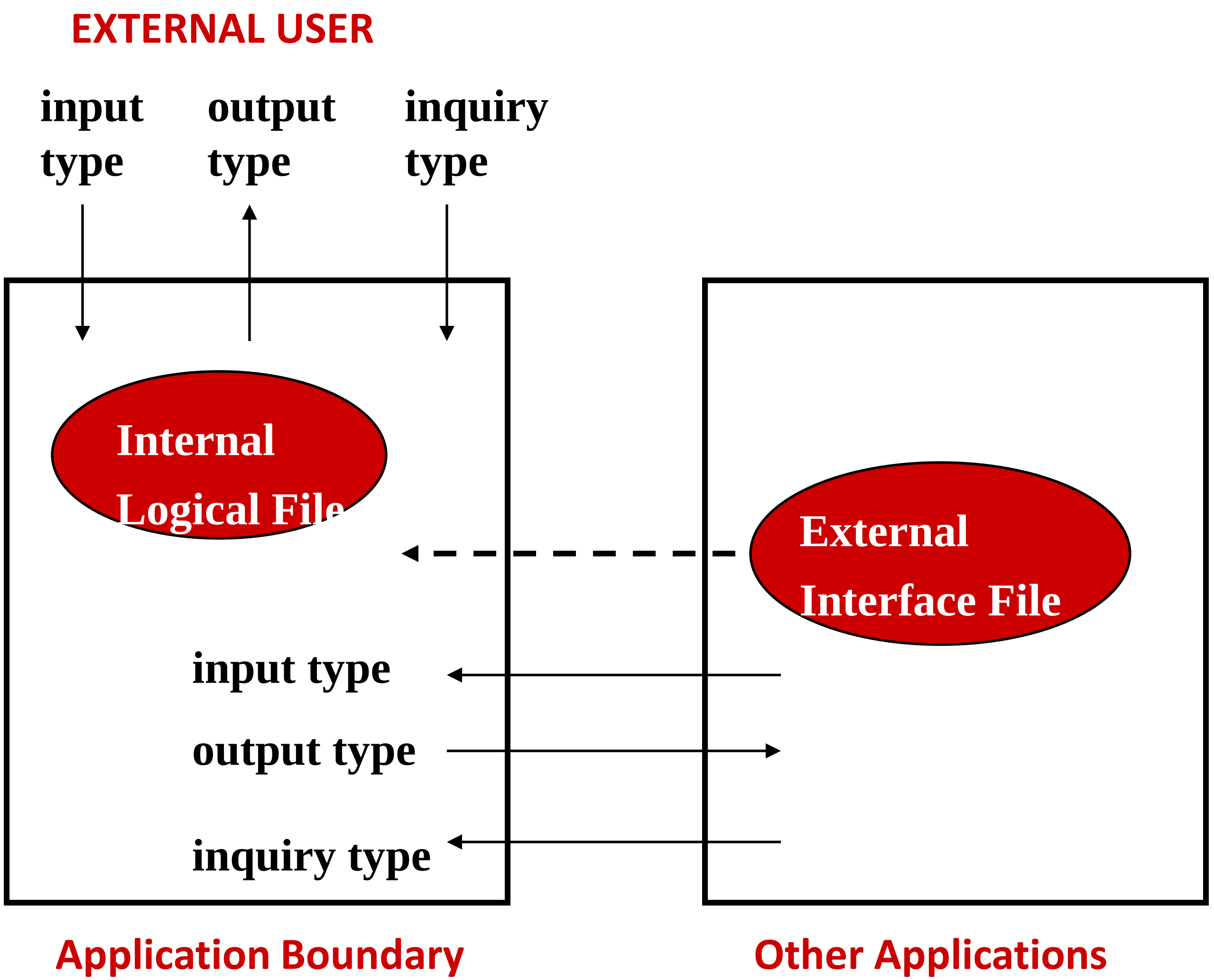
Inputs

Internal Files

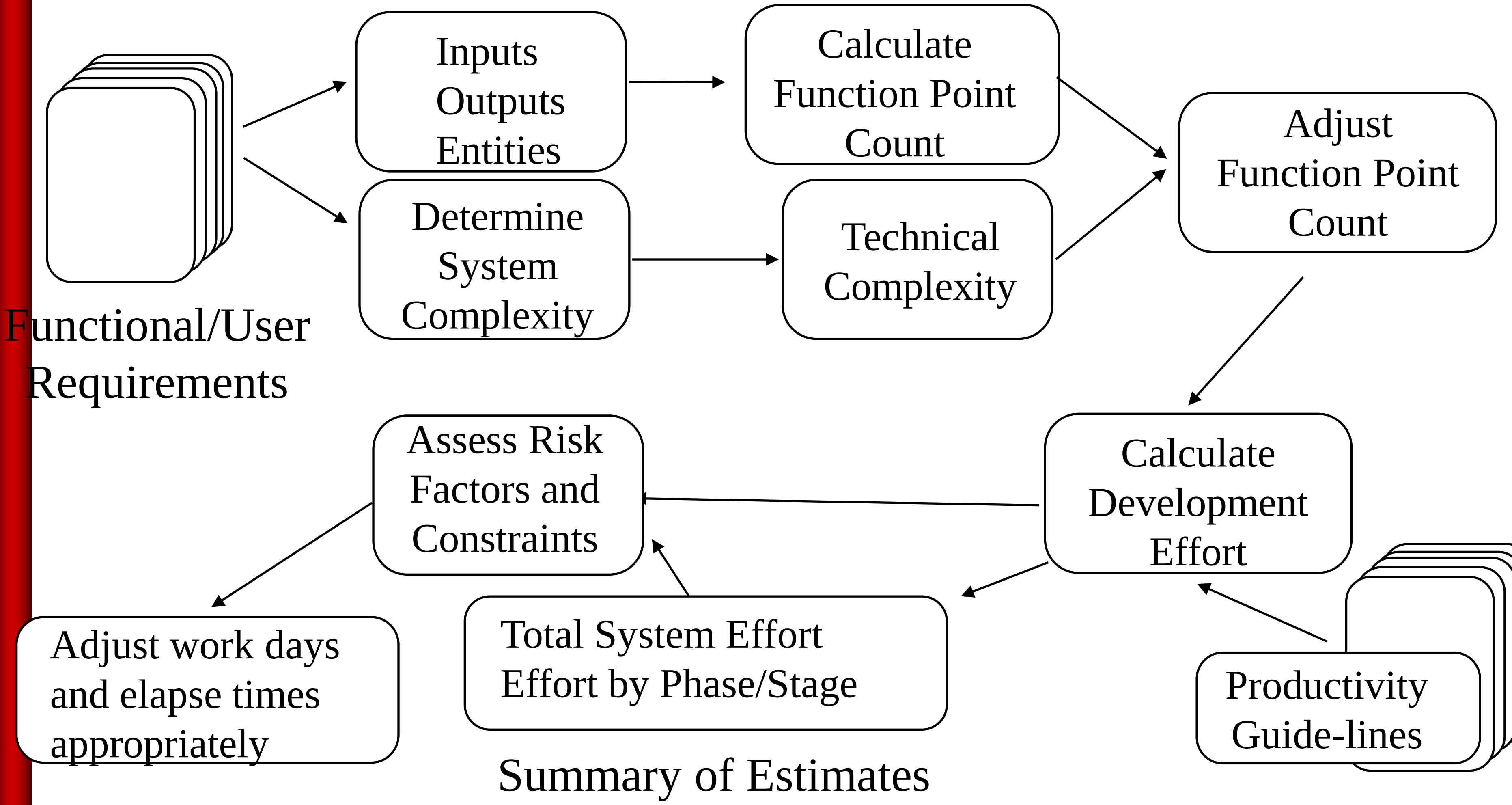
Outputs

Queries

FP- Functionality Types



FPA Estimates



Functional/User Requirements

Inputs Outputs Entities

Determine System Complexity

Calculate Function Point Count

Technical Complexity

Adjust Function Point Count

Calculate Development Effort

Productivity Guide-lines

Assess Risk Factors and Constraints

Total System Effort
Effort by Phase/Stage

Summary of Estimates

Adjust work days and elapse times appropriately

Components of FP

- Data functions
  - Internal logical files
    - Major logical group of end user data
    - Might consists of single flat file or a single table in relational databases
  - External interface files
    - The data resides in another system and is maintained by another user or system
    - Shared databases, shared routines

Components of FP

- Transactional functions
  - External inputs
    - Screens, forms, dialog boxes, controls or messages through which an end user or other program adds, deletes or changes a program's data
  - External outputs
    - Screens, reports, graphs or messages that a program generates for a use of an end user or other program
  - External inquiries
    - Input/output combination

Complexity Level

For ILF and EIF			
Record Elements	Data Elements		
	1 - 19	20 - 50	51+
1	Low	Low	Avg
2 – 5	Low	Avg	High
6+	Avg	High	High

Complexity Level

For EO and EQ			
File Type	Data Elements		
	1 – 5	6 - 19	20+
0 – 1	Low	Low	Avg
2 – 3	Low	Avg	High
4+	Avg	High	High

Complexity Level

For EI			
File Types	Data Elements		
	1 – 4	5 - 15	16+
0 – 1	Low	Low	Avg
2 – 3	Low	Avg	High
3+	Avg	High	High

Weighting Factors

	Low	Average	Complex
Internal logical files	7	10	15
External interface files	5	7	10
External inputs	3	4	6
External outputs	4	5	7
External inquiries	3	4	6

Un-adjusted function points are computed using
UFP = $\sum \sum X_i W_j$ where i=1..5, j=1..3

Environment Complexity Factor

- Is data communication required?
- Are there distributed processing functions?
- Is performance critical?
- Will the system run in an existing, heavily utilized environment?
- Does the system require online data entry?
- Is application designed to facilitate the ease of use by user?
- Are inputs, outputs or inquiries complex?

- Are the master files updated online?
- Is the internal processing complex?
- Is the code designed to be reusable?
- Are conversation and installation included in the design?
- Does the system require reliable backup?
- Is the system designed for multiple installations on different sites?
- Is the application designed to facilitate the change?

Q&A