Software Project Management

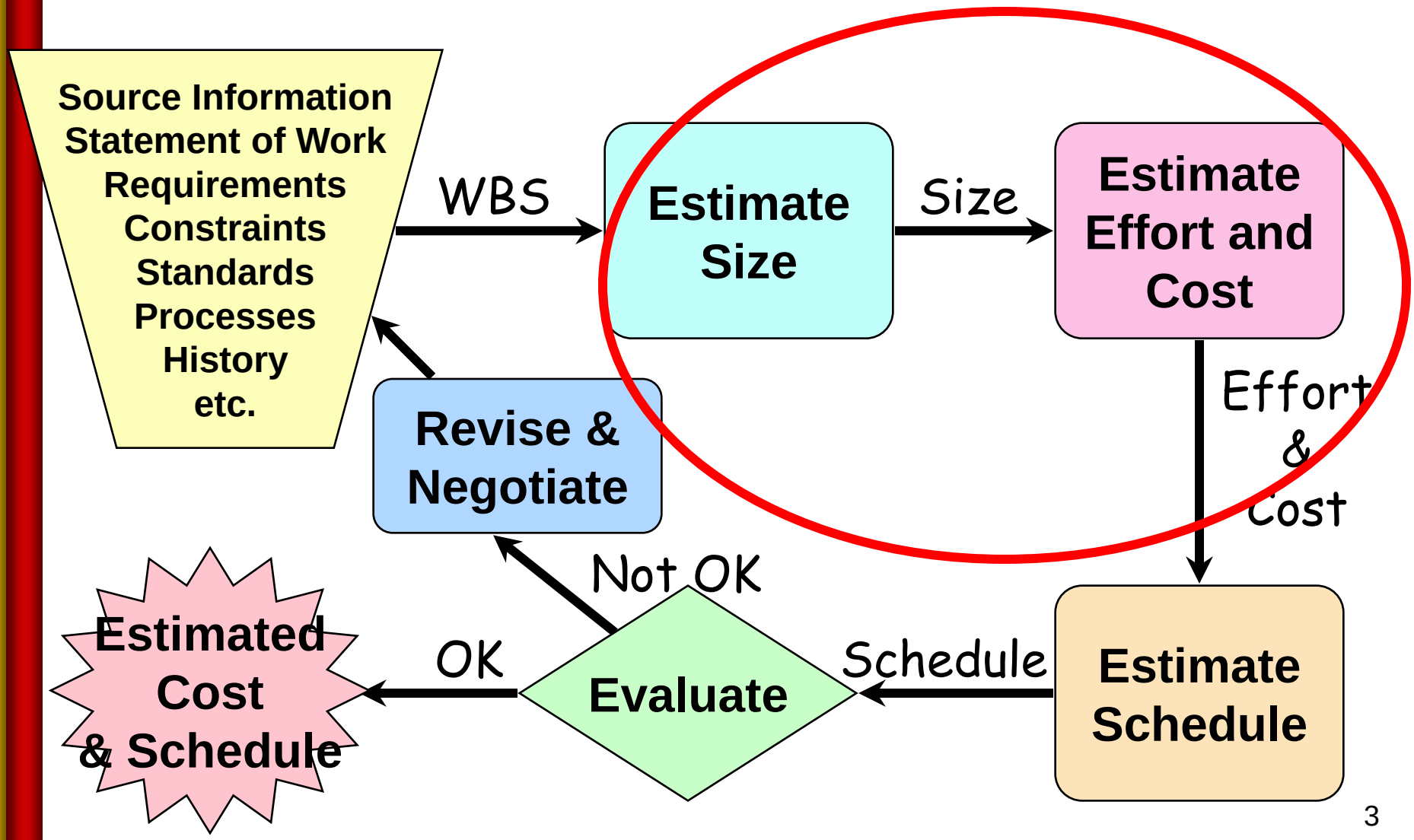
Lecture 18

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



Object Points

- Alternative function related measure to function points
- Object point metrics involve estimating the size of the application in terms of the number of objects that will be require delivering the desired functionality
- The term "objects" includes
  - The number of separate screens that are displayed
  - The number of reports that are produced by the system
  - The number of 3GL modules that must be developed to supplement the 4GL code

Access and Classify Objects (Screens)

	Number and source of data table		
No of views contained	**Total < 4 (<2 srvr, <3 clnt)**	**Total < 8 (2/3 srvr, 3-5 clnt)**	**Total 8+ (> 3 srvr, < 5 clnt)**
< 3	Simple	Simple	Medium
3 – 7	Simple	Medium	Difficult
> 8	Medium	Difficult	Difficult

Access and Classify Objects (Reports)

	Number and source of data table		
No of sections contained	**Total < 4** (<2 srvr, <3 clnt)	**Total < 8** (2/3 srvr, 3-5 clnt)	**Total 8+** (> 3 srvr, < 5 clnt)
0 or 1	Simple	Simple	Medium
2 or 3	Simple	Medium	Difficult
4 +	Medium	Difficult	Difficult

Assess the Complexity

	Simple	Medium	Complex
Screens	1	2	3
Reports	2	5	8
3 GL Modules	-	-	10

Percentage of Re-Use

- Estimate percentage of reuse; expected to be in the project and compute new Object Points.
  - **NOP = (OPs) (100 - % reuse) / 100**

Determine the Productivity Rate

Developers experience And Capability	VL	L	N	H	VH
ICASE maturity and Capability	VL	L	N	H	VH
	4	7	13	25	50

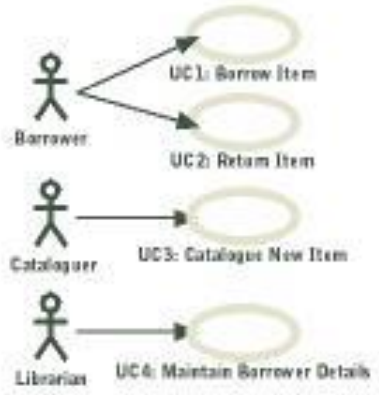
Compute the estimated person months

PM = NOP / PROD

Use Case Points

- 1993 Gustav Karner for estimating a project size
- Use case and function points have similarity
- Use case technique for estimation is similar as function point
  - Count key aspects of your requirements to form an *unadjusted point count*
  - Use several sets of questions about your team and their environment to *create a fudge factor.*
  - Multiply your original count by the fudge factor to come up with *an adjusted point count*
- UCP = UUCP * TCF * ECF

Do Use Cases Improve Estimates?



Use Cases provide a mechanism for articulating the product scope from the user's point of view

Size and Complexity can be attributed to each Use Case, but are really based upon the artefacts derived from Use Cases

Steps to generate estimates

- Identify, classify and weight actors
- Identify, classify and weight use cases
- Calculate Unadjusted Use Case Points
- Identify and Weight Technical Complexity Factors
- Identify and Weight Environmental Complexity Factors
- Calculate Adjusted Use Case Points
- Converting Points into Time

Identify, classify and weight actors

Actor Type	Description	Weight
Simple	The Actor represents another system with a defined API.	1
Average	The Actor interacting through a protocol, like TCP/IP.	2
Complex	The Actor is a person interacting via an interface.	3

The UAW is calculated by **count**ing the number of actors in each category, **multiply**ing each total by its specified weighting factor, and then **add**ing the products.

Identify, classify and weight use cases

Use Case Type	Description	Weight
Simple	• Its success scenario has 3 steps or less (transactions) OR • Its implementation involves less than 5 classes.	5
Average	• Between 4 to 7 steps OR • Its implementation involves between 5 to 10 classes.	10
Complex	• Over seven steps OR • Its implementation involves more than 10 classes.	15

The UUCW is computed by **count**ing the number of use cases in each category, **multiply**ing each category of use case with its weight and **add**ing the products.

a transaction is defined as an atomic set of activities that is performed entirely or not at all.

Unadjusted Use Case Points (UUCP)

- UUC Points are computed based on two computations:
- The Unadjusted Use Case Weight (UUCW)
  - based on the total number of activities (or steps) contained in all the use case Scenarios.
  - Alternatively may be based on number or classes
- The Unadjusted Actor Weight (UAW)
  - based on the combined complexity of all the use cases Actors.
- UUCP = UUCW + UAW

Technical Complexity Factors

Factor	Description	Weight
T1	Distributed System	2
T2	Performance	2
T3	End-user efficiency	1
T4	Complex internal processing	1
T5	Reusability	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Security features	1
T12	Access for third parties	1
T13	Special user training required	1

Technical Complexity Factors

- The technical factors are evaluated by the development team and assigned a value from 0 to 5 according to their perceived complexity
- A perceived complexity of
  - 0 means factor is irrelevant
  - 3 is average
  - 5 means it has strong influence.
- Each factor's weight is multiplied by its perceived complexity to produce its calculated factor. The calculated factors are summed to produce the Total *TFactor.*
- TCF = 0.6 + (0.01 * TFactor)

Environment Factors

Factor	Description	Weight
F1	Familiar with UML	1.5
F2	Application experience	0.5
F3	OO experience	1
F4	Lead analyst capability	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part-time workers	-1
F8	Difficult programming language	2

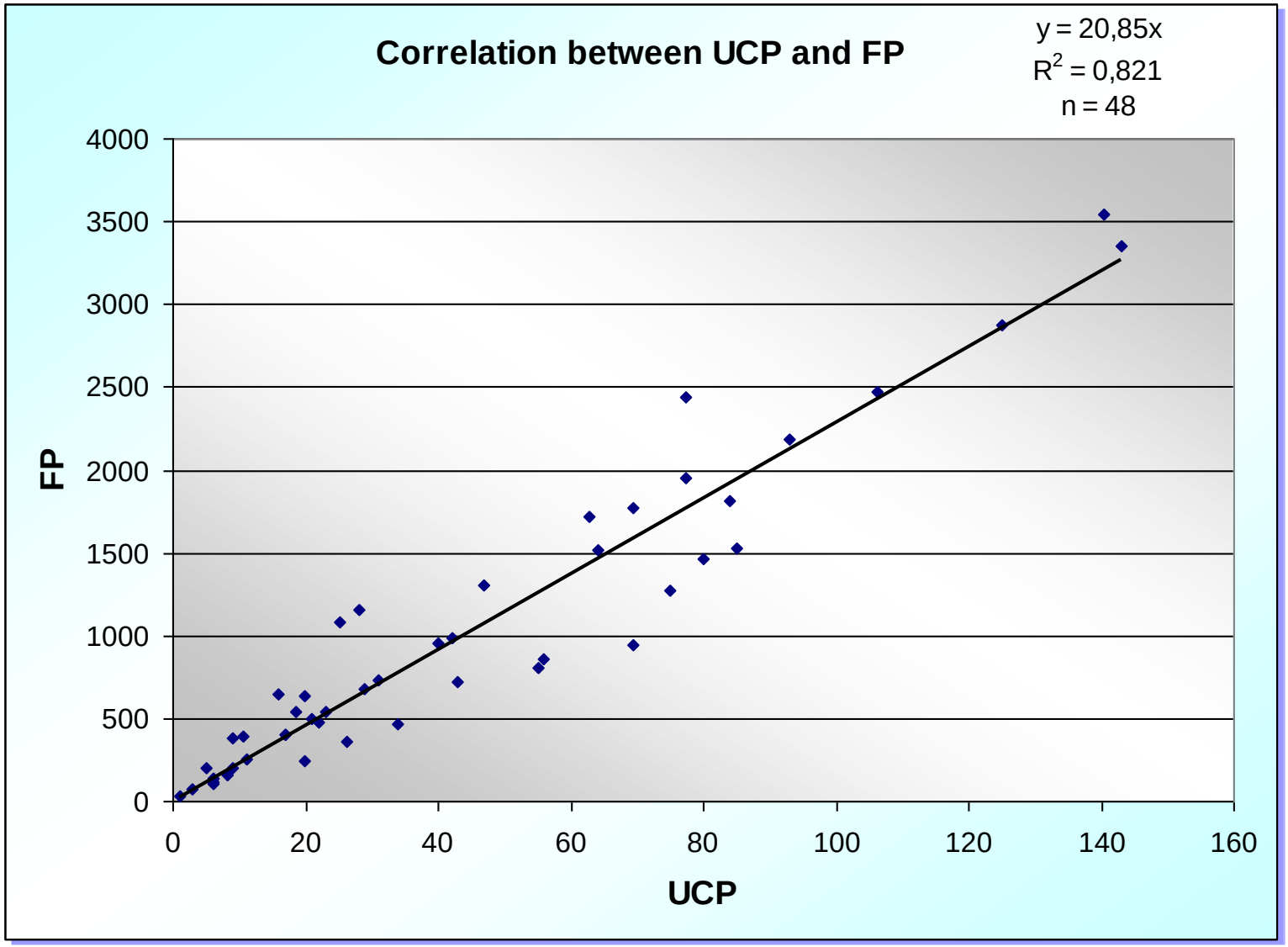
Environment Factors

- Influence
  - 0 means no influence
  - 3 means average influence
  - 5 means strong influence
- EFactor
  - Each factor's weight is multiplied by its perceived complexity to produce its calculated factor. The calculated factors are summed to produce the Total EFactor.
- ECF = 1.4 + (-0.03 * EFactor)

Adjusted UCP and Effort

- UCP = UUCP * TCF * ECF
- UCP is the size
- Time? Effort?
- Productivity Factor (PF)
  - a ratio of the number of person hours per use case point based on past projects.
  - If no historical data has been collected, a figure between 15 and 30 is suggested by industry experts.
    - Sun's Experience
  - A typical value is 20.
- Use PF to calculate person hours required

Correlation between UCP and FP



DELPHI

- Based on the Hegelian Principle (George Wilhelm Friedrich Hegel, 1770-1831)
- Three-step process of
  - Thesis
    - All present their opinion or views on a given subject
  - Antithesis
    - Establishing views and opposing views
  - Synthesis
    - opposites are brought together to form the new thesis

DELPHI

- A group meeting is held to discuss the product and estimation issues
- Experts produce an independent estimate
- Estimates are returned indicating the median estimate and the expert's personal estimate
- Another group meeting is held to discuss results
- Experts prepare a revised independent estimate
- Steps 3-6 are repeated until the panel of experts reaches a consensus

Wideband Delphi

- Rand corporation used original Delphi approach to predict future technologies
- 1970s, Barry Boehm and his RAND colleagues modified DELPHI method into Wideband DELPHI, which included more estimation team interaction
- Group consensus approach
  - This is a disciplined method of using the experience of several people to reach an estimate that incorporates all of their knowledge.
- Present experts with a problem and response form
- Conduct group discussion, collect anonymous opinions, then feedback
- Conduct another discussion & iterate until consensus

Wideband Delphi

- 1) Get a few experts (typically 3 to 5)
  - Include experience in all of the “risk” areas -- application domain, programming language, algorithms, target hardware, operating system, etc.
- 2) Meet with them to discuss issues and describe the software to them
  - Specifications, other source documents, WBS, etc.
  - Let them add their own information, questions, etc.
  - All take notes
- 3) Each expert develops an estimate
  - min, expected, max -- or just expected
  - independent and anonymous
- 4) Record estimates anonymously
  - Usually done by a facilitator

WBD

- 5) Meet and have each expert discuss his/her estimate
  - Assumptions
  - Rationale
- 6) Seek consensus on assumptions, etc.
  - May result in action items to gather factual data
- 7) Each expert updates his or her estimate based on the new information
- 8) Record the new estimates

WBD

- 9) Discuss again
  - Typically, new questions will come up
  - But this round is typically much shorter
- 10) Repeat from step 7 until you reach a consensus
- 11) If no consensus, break until you can gather additional data, then repeat from step 7

WBD

- Stop repeating when:
  - a) You reach consensus (i.e., the experts agree)
    or
    b) Two consecutive cycles do not change much and there is no significant additional data available (i.e., they agree to disagree)

- At the end, you have a consensus estimate on the expected value. You should also agree on a minimum and maximum so you understand the degree of confidence in the estimate.

Advantages of WBD

- Takes advantage of the expertise of several people
- All participants become better educated about the software
- Buy-in to final estimate
- Does not require historical data (although it can be useful input if it is available)

Drawbacks of WBD

- You can reach consensus on an incorrect estimate
  - Because you all “buy in”, you may not be skeptical enough when actual data shows it is wrong
- You can develop a false sense of confidence
- You may fail to reach a consensus

Q&A