Software Project Planning

Modified from Roger S. Pressman, Software Engineering:

A Practitioner's Approach 8th Edition, McGraw Hill, 2014

Software Project Planning

- The objective is to provide a framework to estimate resources, cost, and schedule
- Steps in software project planning:
  - Scoping: understand the problem and the work that must be done
  - Estimation: how much effort, time, and resources?
  - Risk: What can go wrong? How can we avoid it? What can we do about it?
  - Schedule: How do we allocate resources along the timeline? What are the milestones?
  - Control strategy: How do we control strategy? How do we control change?

บทนี้จะเน้นส่วนนี้เป็นพิเศษ
ส่วนอื่นๆ เรียนมาแล้ว

Software Project Estimation

- A cost estimation error can make profit and loss
- Options to achieve reliable cost and effort estimates are
  - delay estimation until late in the project
  - base estimates on similar projects that have already been completed
  - use decomposition techniques (conventional methods)
  - use empirical models

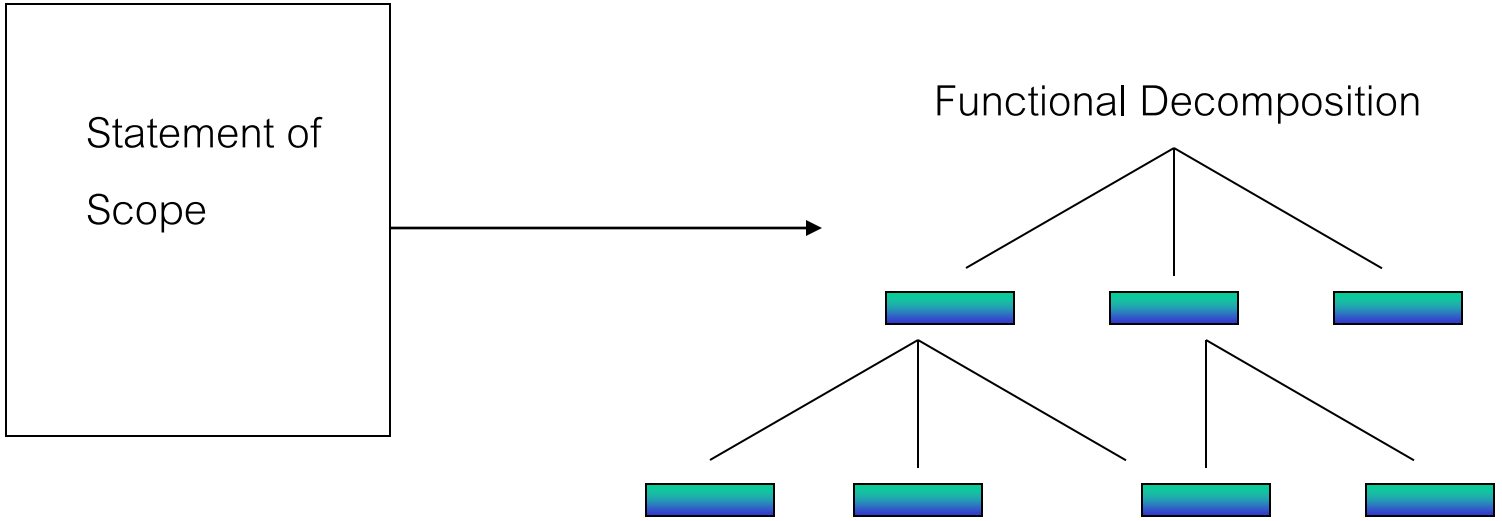
ยากสุดก็คือ
Cost Estimation

Decomposition Techniques

- Problem-Based
  - LOC-Based Estimation
  - FP-Based Estimation
- Process-Based

Problem-Based

(Functional Decomposition)



An Example of LOC-Based Estimation

- The CAD software will accept two- and three- dimensional geometric data from an engineer. The engineer will interact and control the CAD system through a user interface that will exhibit characteristics of good human machine interface design. All geometric data and other supporting information will be maintained in a CAD database. Design analysis modules will be developed to produce required output which will be displayed on a variety of graphics devices. The software will be designed to control and interact with peripheral that include a mouse, digitizer, and laser printer.

An Example of LOC-Based Estimation

- Major software functions are identified.
  - User interface and control facilities (UICF)
  - two-dimensional geometric analysis (2DGA)
  - three-dimensional geometric analysis (3DGA)
  - database management (DBM)
  - computer graphics display facilities (CGDF)
  - peripheral control (PC)
  - design analysis modules (DAM)

An Example of LOC-Based Estimation

Three-point estimate

Function	Opt.	Likely	Pess.	Estimated LOC
UICF				2300
2DGA				5300
3DGA	4600	6900	8600	6800
DBM				3350
CGDF				4950
PC				2100
DAM				8400
Estimated LOC				33200

Three Point Estimation Formula

(PERT Estimate Formula)

- เป็นเครื่องมือจาก Program Evaluation and Review Technique (PERT)
- ใช้ประมาณการค่าตัววัด ที่มีความไม่แน่นอน

(Optimistic + (4xMost Likely) + Pessimistic) / 6

An Example of LOC-Based Estimation

- From historical data, average productivity is 620 LOC/PM
- Labor rate is $8000 per month
- COST/LOC = $13.00
- Total Estimated Project Cost = $432,000
- Total Estimated Effort = 54 PMs

An Example of FP-Based Estimation

Problem-based Estimation

Information Domain Value	Opt.	Likely	Pess.	Est. Weight	FP-Count
number of inputs	20	24	30	24 4	96
number of outputs	12	15	22	16 5	80
number of inquiries	16	22	28	22 4	88
number of files	4	4	5	4 10	40
number of ext. interfaces	2	2	3	2 7	14
count-total					318

An Example of FP-Based Estimation

- $S F_i = 52$
- $FP_{estimated}$ = count-total X [0.65 + 0.01 X $\sum F_i$]
- $FP_{estimated}$ = 372
- from historical data, average productivity is 6.5 FP/PM
- labor rate is $8000/month
- COST/FP = $1230
- total estimated project cost is $464,000
- total estimated effort is 58 PMs

An Example of Process-Based Estimation

Problem-based

Activity	Cust Com.	Plan	Risk Ana.	Eng.		Con.		Cus Eval.	Totals
Task				Ana.	Des	Code	Test		
UICF				0.5	2.5	0.4	5.0		8.4
2DGA				0.75	4.0	0.6	2.0		7.35
3DGA				0.50	4.0	1.0	3.0		8.5
DSM				0.50	3.0	1.0	1.5		6.0
CGDF				0.50	3.0	0.75	1.5		5.75
PCF				0.25	2.0	0.50	1.5		4.25
DAM				0.50	2.0	0.50	2.0		5.0
Total	0.25	0.25	0.25	3.5	20.5	4.75	16.5		46.0
%effort	1	1	1	8	45	10	36		

Process-based

Empirical Estimation Models

- LOC-oriented estimation models
  - $E = 5.2 \times (KLOC)^{0.91}$ Walston-Felix Model
  - $E = 5.5 + 0.73 \times (KLOC)^{1.16}$ Bailey-Basili Model
  - $E = 3.2 \times (KLOC)^{1.05}$ Boehm simple model
  - $E = 5.288 \times (KLOC)^{1.047}$ Doty Model for KLOC > 9
- FP-oriented models
  - $E = -13.39 + 0.0545\ FP$ Albrecht and Gaffney Model
  - $E == 60.62 \times 7.728 \times 10^{-8}\ FP^{3}$ Kemerer Model
  - $E = 585.7 + 15.12\ FP$ Matson, Barnett, and Mellichamp Model

The COCOMO Model version I

- COCOMO stands for COnstructive COst MOdel
- has 3 forms:
  - The Basic COCOMO: a gross estimator that does not take individual complexity characteristics into account
  - The Intermediate COCOMO: considers 15 factors that are called "cost driver attributes"
  - The Advanced COCOMO: introduces "phase sensitive effort multipliers" that enable the estimate to be modified and updated as the project continues

COCOMO I	COCOMO II
COCOMO I is useful in the waterfall models of the software development cycle.	COCOMO II is useful in non-sequential, rapid development and reuse models of software.
It provides estimates of effort and schedule.	It provides estimates that represent one standard deviation around the most likely estimate.
This model is based upon the linear reuse formula.	This model is based upon the non linear reuse formula
This model is also based upon the assumption of reasonably stable requirements.	This model is also based upon reuse model which looks at effort needed to understand and estimate.
Effort equation's exponent is determined by 3 development modes.	Effort equation's exponent is determined by 5 scale factors.
Development begins with the requirements assigned to the software.	It follows a spiral type of development.
Number of submodels in COCOMO I is 3 and 15 cost drivers are assigned	In COCOMO II, Number of submodel are 4 and 17 cost drivers are assigned
Size of software stated in terms of Lines of code	Size of software stated in terms of Object points, function points and lines of code

3 Classes of Software Projects

- Organic Mode
- Semi-detached Mode
- Embedded Mode

Organic Mode

- small software teams develop software in a highly familiar
- in-house environment
- people connected with the project have extensive experience in working with related systems within organization
- no communication overhead
- generally stable development environment

Semidetached Mode

- intermediate level of project characteristics
- mixture of the organic and embedded mode characteristics
- team members all have an intermediate level of experience with related system
- the team has a wide mixture of experienced and inexperienced people
- team members have experience related to some aspects of the system

Embedded Mode

- operate within tight constraints
- product must operate within a strong coupled complex of hw, sw, regulations, and operational procedures
- does not have the option of negotiating easier sw changes and fixes by modifying the requirements and interface specifications
- spend more effort in assuring conformance to the specs and in assuring that the changes are made correctly

Basic COCOMO Model

- $E = a\,(KLOC)^{b}$ --> effort (person-months)
- $D = c * E^{d}$ --> development time (months)

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

Intermediate COCOMO Model

- Organic mode --> a = 3.2
- Semi-detached mode --> a = 3.0
- Embedded mode --> a = 2.8
- $E = a\,(KLOC)^{b} * EAF$
- EAF = Effort Adjustment Factor
- there are many candidate factors to consider in developing a better model for estimating the cost of a software project

Factors Used in the Intermediate COCOMO

- Product attributes
  - RELY: required software reliability
  - DATA: database size
  - CPLX: product complexity
- Computer Attributes
  - TIME: execution time constraint
  - STOR: main storage constraint
  - VIRT: virtual machine volatility
  - TURN: turnaround time

Factors Used in the Intermediate COCOMO

- Personal attributes
  - ACAP: analyst capabilities
  - AEXP: applications experience
  - PCAP: programmer capability
  - VEXP: virtual machine experience
  - LEXP: programming language experience
- Project attributes
  - MODP: modern programming practices
  - TOOL: use of software tools
  - SCED: required development schedule

TABLE 8–2 Software Development Effort Multipliers

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY Required software reliability	.75	.88	1.00	1.15	1.40	
DATA Data base size		.94	1.00	1.08	1.16	
CPLX Product complexity	.70	.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME Execution time constraint			1.00	1.11	1.30	1.66
STOR Main storage constraint			1.00	1.06	1.21	1.56
VIRT Virtual machine volatility[a]		.87	1.00	1.15	1.30	
TURN Computer turnaround time		.87	1.00	1.07	1.15	
Personnel Attributes						
ACAP Analyst capability	1.46	1.19	1.00	.86	.71	
AEXP Applications experience	1.29	1.13	1.00	.91	.82	
PCAP Programmer capability	1.42	1.17	1.00	.86	.70	
VEXP Virtual machine experience[a]	1.21	1.10	1.00	.90		
LEXP Programming language experience	1.14	1.07	1.00	.95		
Project Attributes						
MODP Use of modern programming practices	1.24	1.10	1.00	.91	.82	
TOOL Use of software tools	1.24	1.10	1.00	.91	.83	
SCED Required development schedule	1.23	1.08	1.00	1.04	1.10	

[a] For a given software product, the underlying virtual machine is the complex of hardware and software (OS, DBMS, etc.) it calls on to accomplish its tasks.

Intermediate COCOMO

- each of 15 attributes is rated on 6-point scale (very low, low, nominal, high, very high, extra high)
- derive EAF (Effort Adjustment Factor) based on rating and effort multiplier tables
- Effort = $a(KLOC)^b$ * EAF

COCOMO II

- Application composition model
  - used during the early stages of software engineering
- Early design stage model
  - used once requirements have been stabilized
- Post-architecture-stage model
  - used during the construction of the software

โจทย์เรื่อง COCOMO

1. จงคำนวณหาค่าความพยายาม (Effort) หน่วย Man-Month และเวลาที่ใช้ที่เหมาะสม (Development time) หน่วย Month ในการพัฒนาระบบ Web Online Shopping ซึ่งมีลักษณะดังต่อไปนี้

ระบบ Web Online shopping เป็นระบบขายสินค้าออนไลน์ พัฒนาด้วยภาษาจาวาประกอบด้วย 3 ส่วนย่อยคือ
- ระบบย่อยดูแลลูกค้า (Optimistic=5000 LOC, Most likely=6000 LOC, Pessimistic=8000 LOC)
- ระบบย่อยสินค้า (Optimistic=8000 LOC, Most likely=10000 LOC, Pessimistic=12000 LOC)
- ระบบการสั่งซื้อสินค้า (Optimistic=2000 LOC, Most likely=4000 LOC, Pessimistic=6000 LOC)

แสดงวิธีทำโดยให้ใช้ Intermediate COCOMO model (อย่าลืมคำนวณค่า EAF ด้วยโดยใช้ 15 cost drivers)

2. กำหนดให้ว่า เงินเดือนสมาชิกในทีมเฉลี่ยอยู่ที่ 40,000 บาท

2.1 จะต้องใช้คนกี่คนในการทำงานเพื่อให้เสร็จภายในเวลาที่เหมาะสมที่คำนวณได้ (D)

2.2 จงคำนวณหาประมาณการค่าใช้จ่าย หน่วยเป็นบาท จากค่าที่คำนวณได้ในข้อ 1 และ 2.1

TABLE 8–2 Software Development Effort Multipliers

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY Required software reliability	.75	.88	1.00	1.15	1.40	
DATA Data base size		.94	1.00	1.08	1.16	
CPLX Product complexity	.70	.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME Execution time constraint			1.00	1.11	1.30	1.66
STOR Main storage constraint			1.00	1.06	1.21	1.56
VIRT Virtual machine volatility[a]		.87	1.00	1.15	1.30	
TURN Computer turnaround time		.87	1.00	1.07	1.15	
Personnel Attributes						
ACAP Analyst capability	1.46	1.19	1.00	.86	.71	
AEXP Applications experience	1.29	1.13	1.00	.91	.82	
PCAP Programmer capability	1.42	1.17	1.00	.86	.70	
VEXP Virtual machine experience[a]	1.21	1.10	1.00	.90		
LEXP Programming language experience	1.14	1.07	1.00	.95		
Project Attributes						
MODP Use of modern programming practices	1.24	1.10	1.00	.91	.82	
TOOL Use of software tools	1.24	1.10	1.00	.91	.83	
SCED Required development schedule	1.23	1.08	1.00	1.04	1.10	

[a] For a given software product, the underlying virtual machine is the complex of hardware and software (OS, DBMS, etc.) it calls on to accomplish its tasks.