

บทที่ 6

การประเมินค่าใช้จ่ายซอฟต์แวร์ (Software Cost Estimation)

สอนโดย

ผศ.ดร.วรรัตน์ สงฆ์แป้น

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการคอมพิวเตอร์

มหาวิทยาลัยขอนแก่น

Cost Estimation

- การวิเคราะห์ต้นทุน
- วิเคราะห์ก่อนโครงการจะเริ่มขึ้น

เทคนิคในการประเมินราคาซอฟต์แวร์

- ความแม่นยำในการประเมินราคามีความสำคัญดังนี้คือ
 - สามารถคิดกำไรที่ได้มาและระบบการเงินในบริษัท
 - สามารถคำนวณการใช้ทรัพยากร HW/SW ได้อย่างถูกต้อง
 - สามารถคำนวณราคาต่อโปรเจกต์ย่อย ๆ ได้
- ในการประเมินราคาเราควรที่จะบวกค่าความเที่ยงตรงไว้ $+ - 20\%$

ความแม่นยำในการประเมิน

- แม่นยำในการประมาณขนาด
- แม่นยำในการประมาณกำลังบุคลากรและเวลาที่ใช้จากขนาด
- แม่นยำในการวางแผนที่แสดงถึงความสามารถที่แท้จริงของทีมงาน
- ความคงที่ของความต้องการของผลิตภัณฑ์ และสภาพแวดล้อมต่างๆ

Software Measurement

- direct measures
- indirect measures

Direct measures

- ราคา
- Effort + Time
- จำนวนคำสั่ง LOC (Line of Code)
- ความเร็ว
- จำนวนข้อผิดพลาด

Indirect measures

- ฟังก์ชัน (Function)
 - คุณภาพ
 - ความซับซ้อน
 - ความน่าเชื่อถือ
 - การบำรุงรักษาในภายหลัง

Software Measurement

- Size-Oriented Metrics
 - การวัดในเชิงของขนาด
- Function-Oriented Metrics
 - การวัดในเชิงของงานที่โปรแกรมกระทำ

Size-Oriented Metrics

project	LOC	Effort	\$(000)	pp.doc	Error	Defect	People
<i>A</i>	12,100	24	168	365	134	29	3
<i>B</i>	27,200	62	440	1224	321	86	5
<i>C</i>	20,200	43	314	1050	256	64	6

- ในการที่จะทำการเปรียบเทียบระหว่างโครงการ เรานิยมใช้ จำนวนบรรทัด (LOC-Line of Code) ในการปรับค่า

- errors per KLOC
- defect per KLOC
- \$ per LOC
- documents per KLOC
- error/person-month
- LOC/person-month
- \$/page of documentation

Size-Oriented Metrics

- ไม่เป็นที่ยอมรับสมบูรณ์ว่าเป็นวิธีที่ดีในการวัดประสิทธิภาพของกระบวนการพัฒนา
- ปัญหาของการใช้ **Line of Code (LOC)**
 - Programming language dependent
 - เป็นปัญหากับ programmer ที่ออกแบบโปรแกรมดีและกระชับ

Function-Oriented Metrics

- วัดทำงานที่โปรแกรมกระทำ
- ไม่สามารถวัดได้โดยตรง
- ใช้วิธี **Function Point (FP)**

คำนวณ Function Point (FP)

$$\mathbf{FP = UFP \times VAF}$$

จำนวนของฟังก์ชัน หาได้จาก FP ที่ยังไม่ได้ถูกปรับแต่ง (Unadjusted Function Point : UFP) คูณกับค่าปัจจัยคุณลักษณะของระบบ (Value Adjustment Factor : VAF)

$$\mathbf{VAF = 0.65 + [0.01 \times \text{Total DI}]}$$

DI : Degree of Influence

Function Point (FP)

- กระบวนการนับฟังก์ชันพอยต์ มีลักษณะดังนี้

ขั้นที่ 1 นำ Requirement ที่เก็บรวบรวมไว้มาทำการแบ่งฟังก์ชันพอยต์

ขั้นที่ 2 ประเมินความซับซ้อนของฟังก์ชัน

ขั้นที่ 3 เปรียบเทียบความซับซ้อน เพื่อให้ได้ระดับความซับซ้อน เพื่อคำนวณฟังก์ชันพอยต์ที่ยังไม่ได้ปรับค่า (Unadjusted Function Point : UFP)

ขั้นที่ 4 คำนวณค่าตัวแปรปรับค่า (Value Adjustment Factor) ตามลักษณะของโครงการ

ขั้นที่ 5 คำนวณจำนวนฟังก์ชันพอยต์ที่ผ่านการปรับค่า (Adjusted Function Point : AFP)

ขั้นที่ 6 ฟังก์ชันพอยต์ที่ผ่านการปรับค่า สามารถนำไปคำนวณเป็น LOC ได้

ขั้นตอนที่ 1, 2

- จาก Product Backlog ดังรูป หรือสามารถแยกประเภทของ Use Case ตามฟังก์ชันพอยต์

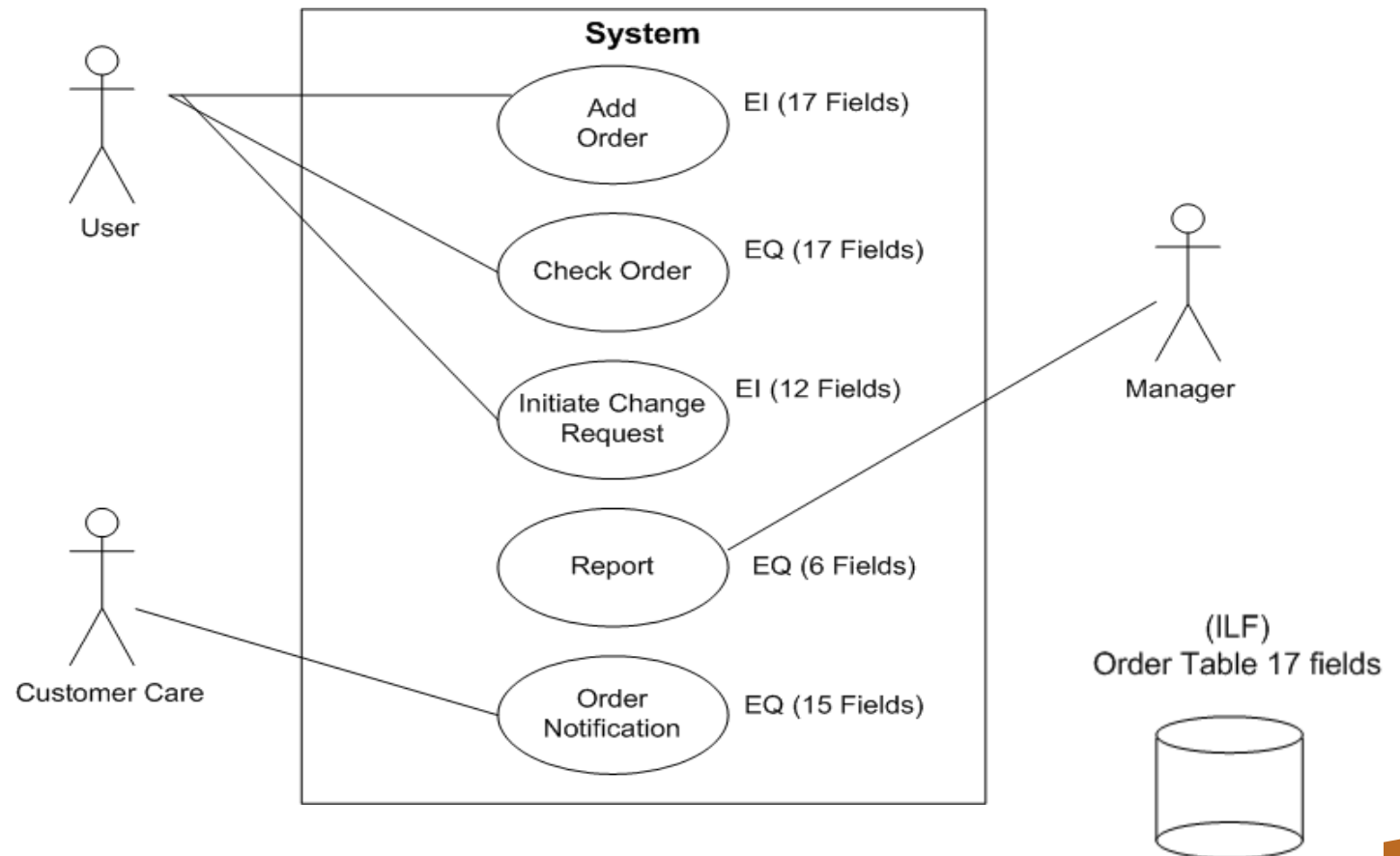
Add
Order

Check
Order

Initiate
Change
Request

Report

Order
Notification



ประเมินความซับซ้อนของฟังก์ชัน

- **External Input (EI)** ข้อมูลที่รับเข้ามาในระบบ เช่น เพิ่ม ลบ แก้ไขข้อมูล เป็นต้น
- **External Output (EO)** ข้อมูลที่เป็นผลลัพธ์จากการประมวลผลข้อมูลที่ได้รับจากภายในระบบ เช่น พิมพ์
- **External Queries(EQ)** กระบวนการดึงข้อมูลและประมวลผลการค้นหาข้อมูลของผู้ใช้
- **Internal Logical Files (ILF)** จัดการเกี่ยวกับไฟล์
- **External Interface Files (EIF)** ไฟล์ที่เกี่ยวข้องข้อมูลที่ใช้อ้างอิงหรือร่วมกับระบบอื่น

ขั้นตอนที่ 3 Unadjusted Function Point : UFP

Item	Type	DETs	RETs/ FTRs	Complexity	Value
Check Order Status	EQ	17	1	Low	1X 3
Add Order	EI	17	1	Average	1X 4
Initiate Change Request	EI	12	1	Low	1 x3
Order Notification	EQ	15	1	Low	1 x 3
Report	EQ	6	1	Low	1 x 3
Order File	ILF	17	1	Low	1 x 7
Summary Unadjusted Function Point: UFP					23 FP

Unadjusted Function Point : UFP

- แต่ละฟังก์ชันพอยต์นั้น มีองค์ประกอบต่างๆ ในฟังก์ชันแต่ละประเภท ซึ่งจะแตกต่างกันได้ เช่น
 - การเกี่ยวข้องกับองค์ประกอบข้อมูล (Data Element : DET)
 - เป็นข้อมูล เปรียบเสมือนฟิลด์ข้อมูลที่น่าสนใจในแต่ละฟิลด์
 - เรคคอร์ดข้อมูล (Record Element : RET)
 - กลุ่มของข้อมูล หรือกลุ่มย่อยของ DET หรือการนับประเภทของเรคคอร์ด ข้อมูลที่เกี่ยวข้องสัมพันธ์กับฟังก์ชันที่น่าสนใจ
 - ประเภทไฟล์ (File Type of Record : FTR)

Unadjusted Function Point : UFP

Item	Type	DETs	RETs/ FTRs	Complexity	Value
Check Order Status	EQ	17	1	Low	1X 3
Add Order	EI	17	1	Average	1X 4
Initiate Change Request	EI	12	1	Low	1 x3
Order Notification	EQ	15	1	Low	1 x 3
Report	EQ	6	1	Low	1 x 3
Order File	ILF	17	1	Low	1 x 7
Summary Unadjusted Function Point: UFP					23 FP

Unadjusted Function Point : UFP

FP Complexity Level			
Internal Logical Files and External Interface files			
Record Elements	Data Elements		
	1-19	20-50	51+
1	Low	Low	Average
2-5	Low	Average	High
6+	Average	High	High
Internal External Output and External Inquiry			
File Types	Data Elements		
	1-5	6-19	20+
0 or 1	Low	Low	Average
2-3	Low	Average	High
4+	Average	High	High
For External Input			
File Types	Data Elements		
	1-4	5-15	16+
0 or 1	Low	Low	Average
2-3	Low	Average	High
3+	Average	High	High

Unadjusted Function Point : UFP

Item	Type	DETs	RETs/ FTRs	Complexity	Value
Check Order Status	EQ	17	1	Low	1X 3
Add Order	EI	17	1	Average	1X 4
Initiate Change Request	EI	12	1	Low	1 x3
Order Notification	EQ	15	1	Low	1 x 3
Report	EQ	6	1	Low	1 x 3
Order File	ILF	17	1	Low	1 x 7
Summary Unadjusted Function Point: UFP					23 FP

Unadjusted Function Point : UFP

Complexity-Weight			
	Complexity-Weight		
Function Type	Low	Average	High
1. Internal Logical File (ILF)	7	10	15
2. External Interface Files (EIF)	5	7	10
3. External Input (EI)	3	4	6
4. External Output (EO)	4	5	7
5. External Inquiry (EQ)	3	4	6

Unadjusted Function Point : UFP

Item	Type	DETs	RETs/ FTRs	Complexity	Value
Check Order Status	EQ	17	1	Low	1X 3
Add Order	EI	17	1	Average	1X 4
Initiate Change Request	EI	12	1	Low	1 x3
Order Notification	EQ	15	1	Low	1 x 3
Report	EQ	6	1	Low	1 x 3
Order File	ILF	17	1	Low	1 x 7
Summary Unadjusted Function Point: UFP					23 FP

UFP = 23 FP

คำนวณ Function Point (FP)

$$\text{FP} = \text{UFP} \times \text{VAF}$$

จำนวนของฟังก์ชัน หาได้จาก FP ที่ยังไม่ได้ถูกปรับแต่ง (Unadjusted Function Point : UFP) คูณกับค่าปัจจัยคุณลักษณะของระบบ (Value Adjustment Factor : VAF)

$$\text{VAF} = 0.65 + [0.01 \times \text{Total DI}]$$

DI : Degree of Influence

ขั้นตอนที่ 4

มีค่า [0-5]

- ทำการเปรียบเทียบค่าของ Value Adjustment Factors : VAF เพื่อหา Total DI ตามสูตร

$$\text{VAF} = 0.65 + [0.01 \times \text{Total DI}]$$

1.	Data communication	5
2.	Distributed data processing	2
3.	Performance	3
4.	Heavily use configuration	0
5.	Transaction rate	0
6.	Online data entry	4
7.	End user efficiency	2
8.	Online Update	4
9.	Complex processing	0
10.	Reusability	3
11.	Installation ease	0
12.	Operational ease	0
13.	Multiple sites	0
14.	Facilitate Change	0

→ Total Degree Influence (TDI) 17

การประเมิน VAF นั้นจะประเมินค่าของ 14 ปัจจัย ดังนี้

1. การติดต่อสื่อสารข้อมูล (Data Communication)
2. การประมวลผลข้อมูลแบบกระจาย (Distributed Data Processing)
3. ประสิทธิภาพของระบบ (Performance)
4. การแก้ไขค่าของระบบ (Configuration)
5. ปริมาณรายการข้อมูล (Transaction)
6. การป้อนข้อมูลเข้าสู่ระบบแบบออนไลน์ (Online Data Entry)
7. ประสิทธิภาพการใช้งานของผู้ใช้ (End user Efficiency)
8. การปรับปรุงข้อมูลแบบออนไลน์ (Online Update)
9. ความซับซ้อนของการประมวลผล (Complex Processing)
10. การนำไปใช้ซ้ำได้ (Reusability)
11. ความง่ายในการติดตั้ง (Installation Ease)
12. ความง่ายในการดำเนินงาน (Operational Ease)
13. การใช้งานได้หลายไซต์ (Multiple Sites)
14. รองรับการเปลี่ยนแปลงความต้องการของผู้ใช้ (Change Requirement)

แทนค่าในสูตร

$$VAF = 0.65 + [0.01 \times 17]$$

$$= 0.82$$

$$FP = UFP \times VAF$$

$$= 23 \times 0.82$$

$$= 18.86 \text{ FP}$$

ตารางเปรียบเทียบค่า FP เพื่อแปลงไปเป็น LOC

Programming language	LOC/FP (average estimate)
assembly language	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
php	67
Java	31
object-oriented languages	30
fourth generation languages (4GLs)	20
code generators	15
spreadsheets (excel programming)	6

ถ้าหากจัดทำซอฟต์แวร์โดยใช้ภาษา PHP
จะได้ค่า LOC

$$= 18.86 \times 67 = 1263.62 \sim 1200 \text{ LOC}$$

การประมาณการบุคลากร

- Productivity : ประสิทธิภาพในการผลิตงาน

$$\text{Productivity} = \frac{\text{Output Size (LOC or Function Point)}}{\text{Effort (Man-Month)}}$$

COCOMO Model

- Boehm B.W. ได้พัฒนา COCOMO Model (Constructive Cost Model) เพื่อวัด Effort ในการพัฒนาซอฟต์แวร์ที่คิดเป็นหน่วยคน-เดือน (person-month) ที่ประมาณจากขนาดของโปรแกรมโดยนับจำนวนบรรทัดของโปรแกรมต้นฉบับเป็นหลัก

COCOMO Model

1000 LOC = 1 KLOC

$$E = a (KLOC)^b \times EAF \quad \text{effort (person-month)}$$

$$D = c E^d \quad \text{duration (month)}$$

- a b c และ d : constant ที่ขึ้นกับประเภทของ software
- EAF : Effort Adjustment Factor

ค่า **constant** ขึ้นอยู่กับประเภทของ **Software**

System	a	b	c	d
organic	3.2	1.05	2.5	0.38
semidetached	3.0	1.12	2.5	0.35
embedded	2.8	1.20	2.5	0.32

Organic

- ใช้ทีมขนาดเล็ก(1-3คน) ภายใต้ HW, SW และการประยุกต์ที่คุ้นเคย
- บุคลากรที่เกี่ยวข้องกับการพัฒนามีประสบการณ์เป็นอย่างดีเกี่ยวกับซอฟต์แวร์ที่มีลักษณะคล้ายๆกันมาก่อน
- กิจกรรมต่างๆเริ่มต้นได้อย่างรวดเร็ว
- ซอฟต์แวร์มีขนาดเล็ก

Semi-detached

- เป็นซอฟต์แวร์ที่มีขนาดและความยากปานกลาง
- ทีมงานประกอบด้วยกลุ่มบุคคลที่มีทั้งประสบการณ์และไม่มีประสบการณ์
- อยู่ระหว่างประเภท **Organic** และ **Embedded**

Embedded

- เป็นซอฟต์แวร์ที่พัฒนาภายใต้สิ่งแวดล้อมที่มีความซับซ้อน มีความเสี่ยงสูงและมีเงื่อนไขที่ยุ่งยาก
- ซอฟต์แวร์ที่พัฒนาแล้วจะถูกนำไปใช้ภายใต้สิ่งแวดล้อมที่ไม่มีความยืดหยุ่น
- เช่น ซอฟต์แวร์ที่ใช้ควบคุมเกี่ยวกับอุปกรณ์ เช่น **Internet of Things (IoTs)**

Effective Adjustment Factor: EAF

- Product AttributesEffective Adjustment Factor: EAF

Cost Drivers	Very Low	Low	Nominal ;	High	Very High
--------------	----------	-----	-----------	------	-----------

Product Attributes

Required Software Reliability	0.75	0.88	1.00	1.15	1.40
Size of Application Database		0.94	1.00	1.08	1.16
Complexity of The Product	0.70	0.85	1.00	1.15	1.30

- Hardware Attributes

Hardware Attributes				
Runtime Performance Constraints		1.00	1.11	1.30
Memory Constraints		1.00	1.06	1.21
Volatility of the virtual machine environment	0.87	1.00	1.15	1.30
Required turnabout time	0.94	1.00	1.07	1.15

- Personnel Attributes

Personnel attributes					
Analyst capability	1.46	1.19	1.00	0.86	0.71
Applications experience	1.29	1.13	1.00	0.91	0.82
Software engineer capability	1.42	1.17	1.00	0.86	0.70
Virtual machine experience	1.21	1.10	1.00	0.90	
Programming language experience	1.14	1.07	1.00	0.95	

■ Project Attributes

Project Attributes					
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82
Use of software tools	1.24	1.10	1.00	0.91	0.83
Required development schedule	1.23	1.08	1.00	1.04	1.10

Effective Adjustment Factor: EAF

- cost driver attribute

■ Product	high	1.15
■ Hardware	high	1.06
■ Personal	low	1.13
■ Project	low	1.17

<https://www.geeksforgeeks.org/software-engineering-cocomo-model/>

ตัวอย่าง

- $EAF = 1.15 * 1.06 * 1.13 * 1.17 = 1.61$
- สมมติว่าทีมของคุณเป็นแบบ Organic
- $E = a (KLOC)^b \times EAF$
- $E = 3.2 \times (1.2)^{1.05} \times 1.61 = \dots PM$
 - ใช้กำลังบุคลากรประมาณ คน-เดือน
- $D = c E^d$
- $D = 2.5 \times (\dots)^{0.38} = \dots M$
 - ใช้เวลาประมาณ เดือน
- สรุปคือ ถ้าคุณจะงบประมาณสำหรับโปรเจกต์นี้ สมมติคุณให้เงินเดือนเฉลี่ย 12,000/คน
- คน * 12000 บาท * เดือน =+-20%

Homework

- ให้แต่ละกลุ่ม ทำการคำนวณการประมาณการค่าใช้จ่ายโครงการของห้องประชุมดังกล่าว
- ส่งภายในวันที่ 10 มีนาคม 2565 ก่อนเวลา 21.00 น. ในอีเมล wararat@kku.ac.th