รายการอ้างอิง

1) Sommerville, I. Software Engineering. 5th ed. England: Addison-Wesley, 1997.

2) Barry W. Boehm , Software Engineering Economics , Englewood Cliffs ,New Jersey : Prentice-Hall, Inc., ,1981.

3) COCOMOII Model Definition Manual , USC , 1997
URL: http://sunset.usc.edu/COCOMOII/Cocomo.html/

4) Boehm, B., Madachy, R., and Selby, R., Cost Models for Future Software life Cycle Processes : COCOMO2.0 , USC Center for Software Engineering , 1997.
URL: ftp://ftp.usc.edu/pub/soft_engineering/cocomo2/

5) Maurice R.Davie. Folkways. Encyclopedia of the Social Sciences. V(1931).

6) เกษม บุญอ่อน. เคลฟ่าย: เทคนิคการวิจัย ครุปริทัศน์, 2522

7) Quinlan J.R , C4.5:Programs for Machine learning , San Matco, CA: Morgan Kaufmann,1993.

8) อุไร ทองหัวไผ่ , การพัฒนาโปรแกรมสำหรับประมาณค่าใช้จ่ายซอฟต์แวร์ โดยใช้แนวความคิดตามแบบจำลองโคโคโม , วิทยานิพนธ์มหาบัณฑิตจุฬาลงกรณ์มหาวิทยาลัย , 2537

9) Clark, B.K., Cost modeling process maturity-COCOMO 2.0, Aerospace Applications Conference, 1996. Proceedings., 1996 IEEE Volume: 3 , Page(s): 347 -369

10) Mukhopadhyay, T.; Kekre, S. Software effort models for early estimation of process control applications, Software Engineering, IEEE Transactions on Volume: 18 , Page(s): 915 –924.

11) Ronald E.Walpole and Raymond H.Myers. Probablty and Satiatics for Engineers and Scientistics,5th ed. New Jersey: A Simon &Schuster Company,1972.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

แบบสอบถามรอบที่ 1

เรื่อง การสร้างชุดคำถามเพื่อใช้ในการประมาณค่าใช้จ่ายของการพัฒนาซอฟต์แวร์ โดยใช้แบบจำลองโคโคโม2 ตามทัศนะของผู้เชี่ยวชาญ

วัตถุประสงค์

เพื่อให้ผู้เชี่ยวชาญแสดงความคิดเห็นเกี่ยวกับชุดคำถามที่ผู้วิจัยสร้างขึ้นในแต่ละตัวขับค่าใช้จ่าย (Cost Driver)

ขั้นตอนและการดำเนินการ

1) ให้ผู้เชี่ยวชาญอ่านคำจำกัดความและตารางอธิบายความแตกต่างแต่ละระดับของตัวขับค่าใช้จ่ายทั้ง 17 ตัว ซึ่งตัวขับค่าใช้จ่ายทั้ง 17 ตัวนี้ คือ ปัจจัยพื้นฐานที่ใช้ประมาณค่าใช้จ่ายของการพัฒนาซอฟต์แวร์ สามารถแบ่งออกเป็น 4 กลุ่มใหญ่ๆ คือ ปัจจัยกลุ่มผลิตภัณฑ์(Product Factors) ปัจจัยกลุ่มแพลทฟอร์ม(Platform Factors) ปัจจัยกลุ่มบุคลากร(Personal Factors) และปัจจัยกลุ่มโครงการ(Project Factors)

2) ให้ผู้เชี่ยวชาญแสดงความคิดเห็นเกี่ยวกับคำถามที่ผู้วิจัยสร้างขึ้น โดย

2.1) ให้น้ำหนักกับชุดคำถามที่สร้างขึ้นโดยทำเครื่องหมายถูก(✓)ลงในช่องคะแนน ระดับคะแนนมีความหมายดังนี้

5 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับมากที่สุด

4 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับมาก

3 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับปานกลาง

2 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับน้อย

1 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับน้อยที่สุด

2.2) ถ้าท่านมีคำแนะนำเกี่ยวกับความไม่ชัดเจนของคำถามหรือมีคำแนะนำอื่นๆ สามารถแสดงเพิ่มเติมลงในส่วนที่เป็นข้อเสนอแนะอื่นๆ

ผู้วิจัยใคร่ขอความกรุณาจากท่านตอบแบบสอบถามในแต่ละข้อให้ครบถ้วน เพื่อเป็นประโยชน์อย่างยิ่งต่อการสรุปผลการวิจัย

ขอขอบพระคุณอย่างสูง

.....

(นางสาวศรินทร์ วัชรบุศราคำ)

แบบจำลองโคโคโม2 (COCOMO2 ย่อมาจาก Constructive Cost Model) เป็นแบบจำลองที่ใช้ในการประมาณค่าใช้จ่ายของการพัฒนาซอฟต์แวร์หรือพัฒนาระบบ ซึ่งในการประมาณค่าใช้จ่ายจะต้องอาศัยปัจจัยหลายอย่างที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ ในแบบจำลองนี้จะเรียกปัจจัยที่เกี่ยวข้องเหล่านั้นว่าตัวขับค่าใช้จ่าย(Cost Driver) ซึ่งผู้วิจัยได้สร้างชุดคำถามที่สอดคล้องกับตัวขับค่าใช้จ่ายแต่ละตัวเพื่อช่วยให้ผู้ที่จะใช้แบบจำลองโคโคโม2 เกิดความสะดวกในการใช้งานเนื่องจากไม่ต้องทำการกำหนดระดับเอง เพียงแต่ตอบคำถามที่ผู้วิจัยสร้างขึ้นเท่านั้น และคำถามที่สร้างขึ้นในบางตัวขับค่าใช้จ่ายจะมีลักษณะเป็นทางเลือกเพียง 2 ทางเลือก คือ ใช่หรือไม่ใช่ เท่านั้น ทั้งนี้เนื่องจากผู้วิจัยได้นำเทคนิคของโปรแกรมการเรียนรู้(Learning Programming) มาช่วยในการวิเคราะห์ผลของคำตอบด้วย แบบจำลองโคโคโม2ได้แบ่งตัวขับค่าใช้จ่ายออกเป็น 4 กลุ่ม ดังนี้

1 <u>ปัจจัยกลุ่มผลิตภัณฑ์</u>

ปัจจัยที่เกี่ยวข้องกับคุณลักษณะของซอฟต์แวร์ที่ต้องการจะพัฒนา ซึ่งแบ่งออกเป็น 5 ประเภท ดังนี้ ความน่าเชื่อถือของซอฟต์แวร์ ขนาดของซอฟต์แวร์ต่อขนาดของข้อมูล ความซับซ้อนของซอฟต์แวร์ การนำซอฟต์แวร์กลับมาใช้ และการจัดทำเอกสารประกอบการพัฒนา

1.1.ความน่าเชื่อถือของซอฟต์แวร์ที่ต้องการ (Required Software Reliability: RELY)

<u>คำจำกัดความ</u>

ความน่าเชื่อถือของซอฟต์แวร์หมายถึงซอฟต์แวร์ต้องสามารถทำงานได้ในช่วงเวลาที่กำหนดอย่างต่อเนื่องและถูกต้อง โดยไม่มีเหตุผิดปกติต่างๆ เช่น การคำนวณผิดพลาด การไม่ทำงานตามขั้นตอน การหยุดทำงานโดยที่ยังไม่จบการทำงานหรือไม่ถูกสั่งให้หยุด เป็นต้น โดยการพิจารณาความน่าเชื่อถือของผลิตภัณฑ์ซอฟต์แวร์ตามแนวทางของโคโคโม2จะพิจารณาจากความเสียหายที่เกิดขึ้นหลังจากซอฟต์แวร์ไม่สามารถทำงานได้ เพื่อประเมินระดับของความพยายามที่จะใช้ในการพัฒนาซอฟต์แวร์

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำที่สุด	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความไม่สะดวกในการทำงานแต่สามารถดำเนินงานต่อไปได้
ต่ำ	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายเพียงเล็กน้อยในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม
ปานกลาง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายปานกลางในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม
สูง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความสูญเสียทางการเงินอย่างมาก
สูงมาก	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความเสี่ยงต่อชีวิตมนุษย์

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและถูกต้องของคำถาม RELY

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์ที่พัฒนาจะถูกนำไปใช้ในด้านที่เกี่ยวข้องกับชีวิตมนุษย์ หรือในด้านที่มีผลต่อชีวิตมนุษย์ ใช่หรือไม่ เช่น ซอฟต์แวร์ที่ใช้ควบคุมอุปกรณ์ทางการแพทย์ เป็นต้น					
2) ซอฟต์แวร์ที่จะพัฒนาสามารถทำให้เกิดความเสี่ยงต่อชีวิตและร่างกายของมนุษย์ได้ ใช่หรือไม่ เช่น ซอฟต์แวร์ที่ควบคุมเครื่องจักรในโรงงาน เป็นต้น					
3) ซอฟต์แวร์ที่จะพัฒนาเป็นซอฟต์แวร์ที่เกี่ยวข้องกับรายรับหรือรายจ่ายขององค์กรเป็นอย่างมาก ใช่หรือไม่					
4) ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้องค์กรเสียค่าใช้จ่ายหรือเสียรายได้จำนวนมากใช่หรือไม่ เช่นเสียรายได้ เสียผลประโยชน์ เป็นต้น					
5) ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติ ท่านจำเป็นต้องดำเนินการให้มีการแก้ไขทันที ใช่หรือไม่					
6) การแก้ไขให้ซอฟต์แวร์สามารถกลับมาทำงานได้ตามปกติ ท่านคาดว่าจะทำให้เกิดการสูญเสียทางการเงินหรือทำให้เกิดความเสียหายกับข้อมูลโดยเฉลี่ยไม่มาก ใช่หรือไม่					
7) ท่านสามารถดำเนินงานในลักษณะนั้นต่อได้ ถึงแม้ว่าซอฟต์แวร์จะไม่สามารถทำงานได้ก็ตาม ใช่หรือไม่ เช่นการใช้บุคลากรทำงานแทน เป็นต้น					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

1.2.ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ (Data Base Size : DATA)

คำจำกัดความ

ตัวขับค่าใช้จ่ายประเภทนี้จะใช้สมการในการคำนวณคือ D/P

โดย D คือ ขนาดของฐานข้อมูล(ไบต์)

P คือ ขนาดของซอฟต์แวร์(จำนวนบรรทัดของโปรแกรม)

ซึ่งจะพบว่าขนาดของฐานข้อมูลมีความสำคัญสำหรับการพิจารณา เนื่องจากถ้าขนาดของฐานข้อมูลมีใหญ่มากก็จะทำให้ได้ค่าจาก D/P มากด้วย ซึ่งหมายถึงว่าข้อมูลที่ใช้ทดสอบซอฟต์แวร์จะต้องมีขนาดใหญ่มากขึ้นด้วย

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำ	D/P น้อยกว่า 10
ปานกลาง	D/P มากกว่าเท่ากับ 10 แต่น้อยกว่า 100
สูง	D/P มากกว่าเท่ากับ 100 แต่น้อยกว่า 1000
สูงมาก	D/P มากกว่า 1000

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม DATA

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ขนาดของหน่วยความจำสำรองที่ใช้เก็บข้อมูล(กิโลไบต์) ☐ในกรณีที่ท่านทราบ ขนาดของหน่วยความจำสำรองที่ใช้เก็บข้อมูลประมาณ.....KB ☐ในกรณีที่ท่านไม่ทราบ ให้ท่านประมาณจำนวนแฟ้มข้อมูล(Files) จำนวนรายการ(Transactions) จำนวนเขตข้อมูล(Fields) และจำนวนอักษร Files จำนวนRecord ความยาวของRecord 1 2 3 ฯลฯ					
2) ขนาดของซอฟต์แวร์ที่จะพัฒนา(จำนวนบรรทัดของโปรแกรม) ☐ในกรณีที่ท่านทราบ ขนาดของซอฟต์แวร์ที่จะพัฒนามีจำนวนบรรทัดของโปรแกรม ประมาณ.....บรรทัด					

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม DATA(ต่อ)

คำถาม	ระดับคะแนน				
	5	4	3	2	1
☐ ในกรณีที่ท่านไม่ทราบ(ประมาณโดยวิธีของFunction Point)					
(1) ภาษาที่ใช้พัฒนาซอฟต์แวร์จัดอยู่ในประเภทใด					
( ) Ada ( ) ANSI/QUICK/Turbo Basic ( ) AI					
( ) Basic Assembly ( ) Macro Assembly ( ) APL					
( ) Basic – Compiled ( ) Fourth-Generator					
( ) Basic – Interpreted ( ) ANSI Cobol 85 ( ) C					
( ) C++ ( ) Fortan77 ( ) Forth					
( ) High-Level ( ) Jovial ( ) Lisp					
( ) Modula2 ( ) Object-Oriented ( ) Procedural Language					
( ) Pascal ( ) Program Generator ( ) Prolog ( ) Report Generator					
( ) Spreadsheet ( ) Shell APL ( ) Query Language					
(2) ประเภทของแฟ้มของข้อมูลที่จะใช้กับซอฟต์แวร์					
(2.1) จำนวนชุดข้อมูลที่รับมาจากภายนอก					
(External Input: EI).....ชุด					
(2.2) จำนวนชุดข้อมูลที่ส่งออกไปภายนอก					
(External Output: EO) .....ชุด					
(2.3) จำนวนชุดข้อมูลภายในเชิงตรรกะ					
(Internal Logical File: ILF) .....ชุด					
(2.4) จำนวนชุดข้อมูลที่ต่อประสานกับภายนอก					
(External Interface File: EIF) .....ชุด					
(2.5) จำนวนชุดข้อมูลที่เป็นคำถามจากภายนอก					
(External Inquiry: EQ) .....ชุด					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

1.3 ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์(Complexity : CPLX)

คำจำกัดความ

ในโคโคโม2จะกำหนดระดับของความซับซ้อนของโปรแกรมที่จะพัฒนา โดยจะแบ่งหัวข้อการพิจารณาออกเป็น 5 หัวข้อย่อยดังนี้

1.3.1 การดำเนินการควบคุม(Control Operation)

การดำเนินการควบคุมหมายถึง กระบวนการที่ใช้ควบคุมการทำงานของโปรแกรม หรือลักษณะการทำงานของโปรแกรมในซอฟต์แวร์ที่จะพัฒนา

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการ(programming operation)ของซอฟต์แวร์ ส่วนใหญ่จะเป็นการดำเนินการแบบตรงไปข้างหน้า(straight-line code) แต่ก็มีลักษณะการดำเนินการแบบเป็นเงื่อนไขแต่เป็นเงื่อนไขที่ไม่ซับซ้อน(non-nested structured) เช่น DO, CASE, IF-THEN-ELSE นอกจากนี้อาจมีการใช้โมดูลง่ายๆ เช่น การทำโปรซีเดอร์คอร์(procedure call) เป็นต้น
ต่ำ	การดำเนินการของซอฟต์แวร์จะมีโครงสร้างที่ซับซ้อน(nesting of structured programming) และโดยส่วนใหญ่จะเป็นแบบมีเงื่อนไข เช่น มีลูป WHILE ซ้อน IF-THEN-ELSE หรือ IF-THEN-ELSE ซ้อน IF-THEN-ELSE เป็นต้น
ปานกลาง	การดำเนินการของซอฟต์แวร์โดยส่วนใหญ่จะเป็นแบบมีเงื่อนไขที่ซับซ้อน จึงมีการใช้ตารางตัดสินใจ(decision table) หรือ มีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบ call back หรือแบบการส่งข้อความ(message passing) รวมทั้งมีการดำเนินการที่สนับสนุนการประมวลผลแบบกระจายบนเครือข่าย(distribute processing) เช่น ระบบclient/server
สูง	โครงสร้างของการดำเนินการมีความซับซ้อนมากซึ่งประกอบด้วยการดำเนินการที่เป็นแบบเงื่อนไขซ้อนเงื่อนไข หรือมีการใช้คิว(queue) และกองซ้อน(stack) เป็นตัวควบคุมลำดับการดำเนินการ หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน (homogeneous) คือ มีซีพียูหรือเวอร์ชันเดียวกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด(soft real time)
สูงมาก	การดำเนินการจะมีลักษณะเป็นแบบการเวียนบังเกิด(recursive) หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความความต่างกัน(heterogeneous) คือ มีซีพียูหรือเวอร์ชันต่างกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด (hard real time)
สูงที่สุด	การดำเนินการของซอฟต์แวร์จะต้องควบคุมการทำงานของฮาร์ดแวร์หรืออุปกรณ์โดยตรง (Microcode level control) หรือมีตารางการใช้ทรัพยากร(resource)ที่สามารถแก้ไขได้โดยอัตโนมัติตามความเหมาะสม หรือมีการประมวลผลแบบกระจาย(distribute processing) ที่ได้ผลลัพธ์แบบทันทีทันใด(hard real time)

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถามการดำเนินการควบคุม

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์ที่พัฒนาต้องเขียนโปรแกรมเป็นภาษาเครื่องเพื่อควบคุมการทำงานหรือใช้งานอุปกรณ์โดยตรง ใช่หรือไม่					
2) ซอฟต์แวร์ที่จะพัฒนานี้สามารถประมวลแบบผลแบบกระจาย(distribute processing) เช่น การประมวลผลแบบ client/sever ใช่หรือไม่					
3) ซอฟต์แวร์ที่พัฒนามีการประมวลผลแบบทันทีทันใด(real time) ใช่หรือไม่					
4) มีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน(homogeneous) คือ มียี่ห้อและเวอร์ชันเดียวกัน หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความความต่างกัน (heterogeneous) คือ มียี่ห้อและเวอร์ชันต่างกัน					
5) ซอฟต์แวร์ที่จะพัฒนามีการเขียนโปรแกรมลักษณะเป็นแบบการเวียนบังเกิด(recursion) ใช่หรือไม่					
6) คำสั่งดำเนินการมีการใช้ตารางตัดสินใจ(decision table) เพื่อช่วยควบคุมการดำเนินการใช่หรือไม่					
7) ซอฟต์แวร์ที่จะพัฒนามีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบcall back หรือแบบการส่งข้อความ(message passing) ใช่หรือไม่					
8) ซอฟต์แวร์ที่จะพัฒนามีส่วนของโปรแกรมเป็นเงื่อนไขซ้อนเงื่อนไข เช่น IFซ้อน IF IF ซ้อนลูป While หรือ ลูป For ซ้อน IF เป็นต้น ใช่หรือไม่					
9) ซอฟต์แวร์ที่จะพัฒนามีโครงสร้างที่ซับซ้อนจึงต้องมีการใช้คิว(Queue)หรือกองซ้อน (Stack) เพื่อจัดลำดับการทำงานของโปรแกรมแบบที่เป็นเงื่อนไข ใช่หรือไม่					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.3.2 การดำเนินการคำนวณ(Computation Operation)

การดำเนินการคำนวณ หมายถึง วิธีการหรือรูปแบบการคำนวณที่มีใช้ในซอฟต์แวร์ที่จะพัฒนา

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการคำนวณสามารถหาค่าจากสมการพีชคณิตหรือสมการเส้นตรง เช่น ทำการบวก การลบ การคูณ การหาร ตัวอย่างเช่น A=B+C*(D-E)
ต่ำ	การดำเนินการคำนวณสามารถหาค่าจากสมการที่มีความยากระดับกลาง(moderate-level expressions) คือ นอกจากทำการบวก การลบ การคูณ การหาร แล้วยังสามารถทำการยกกำลัง การหารากการคำนวณหาค่าตรีโกณมิติ ตัวอย่างเช่น D= SQRT(B^2 – 4*A*C)
ปานกลาง	การดำเนินการคำนวณมีการใช้รูทีน(routine) ที่เป็นมาตราฐานทางคณิตศาสตร์ และสถิติ เช่น การหาส่วนเบี่ยงเบนมาตราฐาน การหาค่า Z เป็นต้น นอกจากนี้ยังยังคำนวณหาค่าของเมตริกซ์(matrix)และเวกเตอร์(vector)ได้
สูง	การดำเนินการคำนวณสามารถวิเคราะห์เชิงตัวเลข(numerical analysis)ขั้นพื้นฐานได้ เช่น การหาคำตอบของสมการโดยวิธีการของนิวตัน(Newton's Method) หรือโดยวิธีการของ Guassian การแก้สมการเชิงอนุพันธ์ เป็นต้น
สูงมาก	การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างแน่นอน เช่น การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน (non-singular matrix equation) การหาค่าจากสมการเชิงอนุพันธ์ย่อย(partial differential equation) และการหารูปแบบของความสัมพันธ์จากข้อมูล 2 ชุด
สูงที่สุด	การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างไม่แน่นอน เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน(highly accurate analysis of noisy) การวิเคราะห์ข้อมูลแบบ stochastic

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถามการดำเนินการคำนวณ

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์ที่จะพัฒนาเกี่ยวข้องกับการวิเคราะห์โครงสร้างข้อมูลเชิงตัวเลขชนิดที่มีโครงสร้างที่ไม่แน่นอน ใช่หรือไม่ เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน การวิเคราะห์สภาพอากาศ เป็นต้น					

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถามการดำเนินการคำนวณ(ต่อ)

คำถาม	ระดับคะแนน				
	5	4	3	2	1
2) ซอฟต์แวร์ที่จะพัฒนาจะต้องวิเคราะห์โครงสร้างเชิงตัวเลขได้จากชุดข้อมูล ใช่หรือไม่ เช่น การวิเคราะห์จุดสูงสุดจุดต่ำสุดของข้อมูล การหาสมการแทนการกระจายของชุดข้อมูล การหาค่าจากสมการเชิงอนุพันธ์ช่วย การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน เป็นต้น					
3) ซอฟต์แวร์ที่จะพัฒนาสามารถวิเคราะห์เชิงตัวเลขขั้นพื้นฐานได้ ใช่หรือไม่ เช่น สามารถหาอนุพันธ์ การหาคำตอบของสมการโดยวิธีการของนิวตัน(Newton's Method) การหาพื้นที่ได้กราฟจากการอินติเกรท(Integrate) เป็นต้น					
4) ซอฟต์แวร์ที่จะพัฒนาต้องมีรูทีนที่เป็นมาตราฐานทางคณิตศาสตร์และสถิติ ใช่หรือไม่ เช่นการหาส่วนเบี่ยงเบนมาตราฐาน การหาค่า Z เป็นต้น					
5) ซอฟต์แวร์ที่จะพัฒนามีการหาคำตอบจากสมการพีชคณิตหรือสมการเส้นตรง เช่น การทำบวก ลบ คูณ หาร ใช่หรือไม่					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

1.3.3 การดำเนินการที่เกี่ยวกับอุปกรณ์(Device-dependent Operation)

การดำเนินการที่เกี่ยวกับอุปกรณ์ หมายถึง วิธีหรือกระบวนการที่ซอฟต์แวร์ที่จะพัฒนาใช้งานอุปกรณ์ เช่น การสั่งให้เครื่องพิมพ์เอกสารพิมพ์รายงาน การจัดเก็บข้อมูลลงในฮาร์ดดิส เป็นต้น

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์เป็นคำสั่งที่ใช้งานง่าย หรือเป็นภาษาระดับสูง เช่น การใช้คำสั่ง อ่าน(read) เขียน(write) ในภาษาพาสคาล(Pascal) ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล
ต่ำ	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์ สามารใช้คำสั่งที่เป็นพื้นฐานที่ภาษามีการเตรียมไว้ให้ เช่น คำสั่งเขียนข้อมูลลงแฟ้มข้อมูล(put) หรือ อ่านข้อมูลในแฟ้มข้อมูล(get) ในภาษาซี(C) โดยที่ผู้พัฒนาไม่จำเป็นต้องมีความรู้เกี่ยวกับอุปกรณ์ที่ใช้กับหรือแสดงผลข้อมูล
ปานกลาง	ผู้พัฒนาจะต้องมีความรู้อย่างเจาะจงเกี่ยวกับอุปกรณ์ที่เป็นตัวรับข้อมูลและตัวแสดงผล (input/output) โดยการใช้งานอุปกรณ์ต้องทำ 3 ขั้นตอนดังนี้คือ เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์นั้นและตรวจสอบความผิดพลาดในการทำงานของอุปกรณ์

ตารางอธิบายความแตกต่างของแต่ละระดับ(ต่อ)

ระดับ	เงื่อนไข
สูง	การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะกระทำในระดับกายภาพ (physical) เช่น การแปลงตำแหน่งของหน่วยความจำทางกายภาพ เพื่อช่วยในการค้นหาและการอ่านและการกำหนดรูปแบบการซ้อนทับ(overlap)ในหน่วยความจำของอุปกรณ์
สูงมาก	การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะมีรูทีน(routines) สำหรับจัดการเมื่อเกิดการผิดพลาดในขณะส่งและรับข้อมูล มีการจัดการเกี่ยวกับการส่งข้อมูลในสายการสื่อสาร หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบเข้มงวด(performance-intensive embedded system) เช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน
สูงที่สุด	มีการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed operations) หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบวิกฤต (performance-critical embedded systems) เช่น มีการบันทึกการทำงานของอุปกรณ์ทุกตัวตลอดเวลา

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถามการดำเนินการที่เกี่ยวกับอุปกรณ์

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์สามารถใช้งานอุปกรณ์โดยการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed operations) ใช่หรือไม่					
2) ซอฟต์แวร์ที่จะพัฒนามีระบบตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์ แบบเข้มงวด เช่นมีการบันทึกการทำงานของอุปกรณ์ทุกตัว หรือ แบบวิกฤตเช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน ใช่หรือไม่					
3) ซอฟต์แวร์ที่จะพัฒนามีรูทีนรองรับการขัดจังหวะ(Interrupt routines) เมื่อเกิดเหตุผิดพลาดขณะที่ซอฟต์แวร์กำลังใช้งานอุปกรณ์ ใช่หรือไม่					
4) ซอฟต์แวร์ที่จะพัฒนาจะต้องสามารถทำการแปลงตำแหน่งเชิงกายภาพ(address) ซึ่งเป็นที่เก็บข้อมูลจริงๆในหน่วยความจำสำรองหรือหน่วยความจำหลัก ใช่หรือไม่					
5) ซอฟต์แวร์ที่จะพัฒนาจะต้องสามารถจัดการกับอุปกรณ์ได้ เช่น เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์ ตรวจสอบความผิดพลาดของอุปกรณ์ เป็นต้น ใช่หรือไม่					
6) ใช้ภาษาระดับสูงในการใช้งานอุปกรณ์สูง เช่น การใช้คำสั่ง อ่าน(read) เขียน(write) ในภาษาพาสคาล(Pascal) ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล ใช่หรือไม่					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

1.3.4 การดำเนินการเกี่ยวกับการจัดการกับข้อมูล(Data Management Operation)

การดำเนินการเกี่ยวกับการจัดการกับข้อมูล หมายถึง ซอฟต์แวร์ที่จะพัฒนามีวิธีการจัดการกับการเก็บข้อมูลและการใช้งานข้อมูล

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลมีการใช้แถวลำดับ(array)ในหน่วยความจำหลัก เพื่อเก็บข้อมูล และสามารถใช้สอบถาม(query)และแก้ไข(update)ได้
ต่ำ	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ แฟ้มข้อมูลสามารถใช้สอบถามและแก้ไขได้
ปานกลาง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลได้หลายแฟ้มข้อมูลโดยจะเป็นแฟ้มส่งข้อมูลออกเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ จะไม่มีการดำเนินการกับแฟ้มข้อมูลที่เกิดในระหว่างการทำงาน เช่น log file
สูง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถทำทริกเกอร์(trigger)โดยใช้ข้อมูลเพียงชุดเดียว สามารถทำการปฏิวัติโครงสร้างของข้อมูลที่ซับซ้อน(complex data restructuring) เช่นการเปลี่ยนคีย์ การเปลี่ยนความสัมพันธ์ระหว่างแฟ้มข้อมูล เป็นต้น
สูงมาก	การดำเนินการเกี่ยวกับการจัดการฐานข้อมูลสามารถทำกับข้อมูลที่เก็บไว้ในหลายสถานที่ได้ สามารถทำทริกเกอร์(trigger)กับข้อมูลจำนวนมากๆได้ สามารถค้นหาข้อมูลในฐานข้อมูลได้
สูงที่สุด	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลเป็นแบบพลวัต(dynamic relational) คือสามารถเปลี่ยนแปลงโครงสร้างบางอย่างโดยอัตโนมัติตามสภาพของข้อมูลที่เก็บได้ มีโครงสร้างเชิงวัตถุ(object structures) มีการจัดการข้อมูลด้วยภาษาธรรมชาติ(natural language data management)

จุฬาลงกรณ์มหาวิทยาลัย

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถามการดำเนินการเกี่ยวกับการจัดการกับข้อมูล

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์ที่จะพัฒนาสามารถใช้งานข้อมูลโดยภาษาธรรมชาติ(Natural Language)ได้ ใช่หรือไม่					
2) ซอฟต์แวร์ที่จะพัฒนามีการใช้งานข้อมูลที่เก็บแบบฐานข้อมูล(database) คือ มีโครงสร้างของฐานข้อมูล ประกอบด้วยแฟ้มข้อมูลมากกว่า 1 แฟ้มและมีการกำหนดความสัมพันธ์ของแฟ้มข้อมูลในฐานข้อมูล ใช่หรือไม่					
3) ซอฟต์แวร์ที่จะพัฒนามีการจัดการกับฐานข้อมูลที่เป็นแบบกระจาย(distributed database) เช่น การจัดการกับสินค้าคงคลังของห้างสรรพสินค้าทุกๆสาขา ใช่หรือไม่					
4) ซอฟต์แวร์ที่จะพัฒนาสามารถทำทริกเกอร์(trigger) ได้ เช่น การแก้ไข บรรณาธิกร การเพิ่ม การลบ ข้อมูลในแฟ้มข้อมูลได้โดยการเขียนเป็นชุดคำสั่งที่มีเงื่อนไขไว้ เมื่อเงื่อนไขถูกตรวจสอบว่าเป็นจริง ชุดคำสั่งนั้นก็จะทำงานทันที ใช่หรือไม่					
5) ซอฟต์แวร์ที่จะพัฒนาสามารถแก้ไขหรือเปลี่ยนแปลงโครงสร้าง(restructuring)ของฐานข้อมูลได้ ใช่หรือไม่ เช่น เพิ่มจำนวนฟิลด์(field)ข้อมูล การเปลี่ยนแปลงความสัมพันธ์ของข้อมูล เป็นต้น					
6) ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลได้หลายแฟ้ม ใช่หรือไม่					
7) ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลเพียงแฟ้มเดียว ใช่หรือไม่					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

1.3.5 การดำเนินการส่วนต่อประสานกับผู้ใช้(User Interface Management Operation)

การปฏิบัติการส่วนต่อประสานกับผู้ใช้ หมายถึง วิธีหรือรูปแบบของซอฟต์แวร์ที่จะพัฒนาติดต่อกับผู้ใช้

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	มีโปรแกรมช่วยในการสร้าง(generators)รูปแบบการรับข้อมูลเข้า(input form) และการทำรายงาน (report) แบบง่าย เช่น มีwizardช่วย
ต่ำ	มีการใช้ตัวสร้างส่วนต่อประสานกับผู้ใช้(user interface)ที่ไม่ยุ่งยาก เช่น ใช้ตัวสร้าง GUI (graphic user interface builders)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ปานกลาง	มีการใช้งานส่วนต่อประสานกับผู้ใช้เป็นแบบวินโดว์ที่ไม่มีความซับซ้อนมาก เช่น มีการติดต่อแบบ windows95 เป็นต้น
สูง	มีการใช้งานส่วนต่อประสานกับผู้ใช้แบบวินโดว์ซึ่งสามารถใช้เสียงเป็นตัวรับและแสดงผล และยังใช้ติดต่อแบบมัลติมีเดียอย่างง่ายได้ด้วย
สูงมาก	มีการใช้ภาพ2 มิติ หรือ 3มิติที่มีความซับซ้อนไม่มาก มีการใช้ภาพที่เป็นพลวัต (dynamic graphic) และมีการใช้มัลติมีเดีย(multimedia)ในการติดต่อกับผู้ใช้
สูงที่สุด	มีการใช้มัลติมีเดียที่ซับซ้อน เช่น วีดิโอคอนเฟอร์เร็น (video conference) และมีการใช้ภาพเสมือนจริง(virtual reality) ในการติดต่อกับผู้ใช้

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถามการดำเนินการส่วนต่อประสานกับผู้ใช้

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้ภาพเสมือนจริง(virtual reality) หรือ วีดิโอคอนเฟอร์เร็น(video conference) ใช่หรือไม่					
2) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้รูปภาพ2มิติ หรือ 3มิติ ที่เคลื่อนไหวได้ ใช่หรือไม่					
3) ซอฟต์แวร์ที่จะพัฒนาสามารถใช้เสียงในการรับหรือแสดงผลได้ ใช่หรือไม่					
4) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้ โดยมีลักษณะการใช้ใกล้เคียงกับการใช้วินโดว์ ใช่หรือไม่ เช่น มีไอคอน(Icon) มีสกอร์บาร์(scroll bars) เป็นต้น					
5) ส่วนต่อประสานกับผู้ใช้แบบรูปภาพแบบง่ายและมีเครื่องมือช่วยสร้าง ใช่หรือไม่					
6) ท่านมีเครื่องมือที่ใช้สร้าง(generators)ส่วนที่ติดต่อประสานกับผู้ใช้แบบไม่มีความซับซ้อน ใช่หรือไม่ เช่น ใช้wizard เป็นต้น					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

1.4 ความต้องการที่จะนำผลิตภัณฑ์กลับมาใช้ใหม่ (Required Reusability : RUSE)

คำจำกัดความ

ตัวขับค่าใช้จ่ายนี้จะพิจารณาความพยายามสร้างซอฟต์แวร์ให้มีลักษณะที่เป็นองค์ประกอบ(component) เพื่อสามารถนำกลับไปใช้(Reuse) กับซอฟต์แวร์อื่นๆในอนาคต ซึ่งการที่จะนำบางส่วนของโปรแกรมกลับมาใช้ในลักษณะใดๆของซอฟต์แวร์ที่จะพัฒนานั้นก็ขึ้นอยู่กับการออกแบบซอฟต์แวร์ให้มีลักษณะเป็นฟังก์ชันหรือเป็นโมดุลที่เล็กที่สุดที่มีฟังก์ชันเดียว

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำ	ไม่มีการออกแบบสำหรับการนำโปรแกรมกลับมาใช้ใหม่ในอนาคต
ปานกลาง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโครงการ (Project)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปแกรมของ Microsoft Word6 มาใช้กับ Microsoft Word7 เป็นต้น
สูง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโปรแกรม (Program)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำบางส่วนของโปรแกรมของMicrosoft Excel มาใช้กับ Microsoft Word เป็นต้น
สูงมาก	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมที่ไม่ได้อยู่ในระบบ(Product Line)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของระบบซื้อ-ขาย มาใช้กับระบบธนาคาร เป็นต้น
สูงที่สุด	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมไปใช้ได้ในหลายๆระบบ เช่น นำส่วนของโปรแกรมของระบบบัญชี ระบบซื้อ-ขาย และระบบงานบุคคล มาใช้กับระบบธนาคาร เป็นต้น

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม RUSE

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมกลับมาใช้ใหม่ในอนาคต ใช่หรือไม่					
2) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบให้มีลักษณะเป็นฟังก์ชันหรือเป็นโมดุลที่เล็กที่สุดที่มีฟังก์ชั่นเดียว(Generic Design) เพื่อให้ซอฟต์แวร์ใดๆก็ได้สามารถนำบางส่วนกลับไปใช้ใหม่ได้ ใช่หรือไม่					
3) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมไปใช้กับระบบที่มีความใกล้เคียงกับระบบที่จะพัฒนาเท่านั้น ใช่หรือไม่					

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม RUSE(ต่อ)

คำถาม	ระดับคะแนน				
	5	4	3	2	1
4) ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโปรแกรม (Program)ที่มีความใกล้เคียงหรือเหมือนกับโปรแกรมที่จะพัฒนา ใช่หรือไม่ เช่น นำบางส่วนของโปรแกรมของMicrosoft Excel มาใช้กับ Microsoft Word เป็นต้น					
5) ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโครงการ (Project)ใหม่ที่มีความใกล้เคียงหรือเหมือนกับโครงการที่จะพัฒนา ใช่หรือไม่ เช่น นำส่วนของโปรแกรมของ Microsoft Word6 มาใช้กับ Microsoft7					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

1.5 ความต้องการเอกสารที่ตรงกับวงจรชีวิต(Documentation match to life-cycle needs: DOCU)

คำจำกัดความ

การกำหนดระดับของตัวขับค่าใช้จ่ายประเภทนี้จะอยู่ในรูปของการจัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์ได้ครบทุกขั้นตอนทั้งโครงการ ซึ่งขั้นตอนหรือรูปแบบของการจัดทำเอกสารก็ขึ้นอยู่กับผู้พัฒนาว่าใช้วงจรชีวิต(life-cycle)แบบใดในการพัฒนาซอฟต์แวร์ เช่น วอลเตอร์ฟอร์โมเดล(Waterfall Model) อินครีเมนต์ดิวิลอปต์โมเดล(Increment Development Model) สไปรรอลโมเดล(Spiral Model) โปรโตไทป์ปิงโมเดล(Prototyping Model) เป็นต้น ซึ่งการกำหนดระดับขึ้นกับเอกสารที่จัดทำครบทุกขั้นตอนและเพียงพอกับความต้องการใช้งานเพียงใด

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	ไม่มีการจัดทำเอกสารการพัฒนาซอฟต์แวร์
ต่ำ	เอกสารจัดทำไม่ครอบคลุมในหลายๆขั้นของวงจรชีวิต
ปานกลาง	เอกสารจัดทำได้ครบตามข้อกำหนดขั้นต่ำของวงจรชีวิต
สูง	เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่ใช้ภายในองค์กรแต่ไม่ได้เป็นข้อกำหนดไว้ในวงจรชีวิต เพื่อทำให้เกิดความสมบูรณ์มากขึ้น
สูงมาก	เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่มีความละเอียดเพื่อให้เกิดความสมบูรณ์มากที่สุด

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม DOCU

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1)ท่านต้องการใช้มีการจัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์ ใช่หรือไม่					
2)ท่านต้องการให้มีการจัดทำเอกสารครบตามที่วงจรชีวิตได้ระบุไว้ ใช่หรือไม่					
3)ท่านต้องการให้มีการจัดทำเอกสารที่นอกเหนือจากวงจรชีวิตระบุเพื่อใช้ในองค์กรด้วย ใช่หรือไม่ เช่นเอกสารที่ระบุถึงความสำคัญของบุคคลในองค์กรที่มีต่อซอฟต์แวร์ที่จะพัฒนา หรือข้อตกลงที่ใช้เฉพาะในองค์กร เป็นต้น					
4)ท่านต้องการให้เอกสารในส่วนที่เพิ่มเติมจากข้อกำหนดของวงจรชีวิตมีความละเอียดในทุกขั้นตอน ใช่หรือไม่					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

2)ปัจจัยกลุ่มแพลทฟอร์ม

ปัจจัยที่เกี่ยวข้องกับฮาร์ดแวร์ ซอฟต์แวร์ที่สนับสนุนการทำงานของซอฟต์แวร์ที่จะพัฒนา ซึ่งแบ่งออกเป็น 3 ประเภทคือ เวลาที่ใช้ในการประมวลผล ขนาดของหน่วยความจำ และการเปลี่ยนแพลทฟอร์ม

2.1.การจำกัดเวลาที่ใช้ในการประมวลผล (Execute Time Constraint: TIME)

คำจำกัดความ

เนื่องจากเวลาที่เข้ากระทำการ(Execution Time)ขึ้นอยู่กับซอฟต์แวร์ระบบ(Software System) การกำหนดระดับของตัวขับค่าใช้จ่ายประเภทนี้จึงคิดเป็นเปอร์เซ็นต์ของเวลาที่คาดว่าซอฟต์แวร์จะเข้ากระทำการ จากเวลาทั้งหมดที่ระบบมีให้กับซอฟต์แวร์

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ปานกลาง	เวลาที่ซอฟต์แวร์เข้ากระทำการน้อยกว่าหรือเท่ากับ 50% ของเวลาที่ซีพียูมีให้
สูง	เวลาที่ซอฟต์แวร์เข้ากระทำการน้อยกว่าหรือเท่ากับ 70% ของเวลาที่ซีพียูมีให้
สูงมาก	เวลาที่ซอฟต์แวร์เข้ากระทำการน้อยกว่าหรือเท่ากับ 85% ของเวลาที่ซีพียูมีให้
สูงที่สุด	เวลาที่ซอฟต์แวร์เข้ากระทำการน้อยกว่าหรือเท่ากับ 95% ของเวลาที่ซีพียูมีให้

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและถูกต้องของคำถาม TIME

คำถาม	ระดับคะแนน				
	5	4	3	2	1
เวลาที่ท่านคาดว่าซอฟต์แวร์จะใช้ในการกระทำการ(Execution Time)					
☐ ในกรณีที่ท่านทราบ					
(1) เวลาที่ระบบให้ซอฟต์แวร์เข้ากระทำการ.....ชั่วโมงต่อวัน					
(2) เวลาที่ซอฟต์แวร์เข้าไปใช้งานในซีพียูโดยประมาณ...ชั่วโมงต่อวัน					
☐ ในกรณีที่ท่านไม่ทราบ					
(1) ซอฟต์แวร์ที่พัฒนาเป็นซอฟต์แวร์ประเภทใด					
( ) การจัดการฐานข้อมูล(Database Management) เช่น MS Access, CA-Clipper, FoxPro, Oracle, Paradox Alpha Four เป็นต้น					
( ) การสื่อสาร(Communication) เช่น Comm Work, Crosstalk, SmartCom, Terminal Plus เป็นต้น					
( ) การประมวลคำ (Word Processing) เช่น AmiPro3.01, MS Word, WordPerfect, WordStar เป็นต้น					
( ) Spreadsheet เช่น CA-SuperCale, Lotus Improv, MS Excel, Quattro Pro เป็นต้น					
( ) รูปภาพ(Graphic) เช่น CorelDRAW, FreeHand, Illustrator, DesignWorks เป็นต้น					
( ) ซอฟต์แวร์ระบบ(System Software) เช่น OS/2, Win95/98/NT, UNIX, Linux เป็นต้น					
( ) Integrated Software เช่น MS Work, Lotus Work, ClarisWork, Framework XE, Legato เป็นต้น					
( ) แอปพลิเคชั่น(Applications) เช่น ระบบบัญชี ระบบบุคลากร เป็นต้น					
( ) ระบบอื่น ระบุ(.....)					
(2) ระบบของท่านสามารถทำงานแบบ					
( ) ผู้ใช้สามารถใช้งานได้พร้อมกันหลายคน					
( ) ผู้ใช้สามารถใช้งานได้ทีละคน					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

2.2.การจำกัดหน่วยความจำหลัก (Main Storage Constraint: STOR)

<u>คำจำกัดความ</u>

การกำหนดระดับให้กับตัวขับค่าใช้จ่ายประเภทนี้จะพิจารณาจากการที่ซอฟต์แวร์ใช้พื้นที่ในหน่วยความจำหลักมากน้อยเพียงใด ซึ่งถ้าซอฟต์แวร์ใช้หน่วยความจำมากซอฟต์แวร์ก็จะต้องมีการจัดการหน่วยความจำที่ดี การกำหนดระดับจะพิจารณาโดยคิดเป็นเปอร์เซ็นต์ของขนาดหน่วยความจำหลักที่คาดว่าจะใช้ จากขนาดของหน่วยความจำหลักที่ระบบมีให้

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ปานกลาง	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักน้อยกว่าหรือเท่ากับ 50% ของหน่วยความจำหลักที่สามารถใช้ได้
สูง	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักน้อยกว่าหรือเท่ากับ 70% ของหน่วยความจำหลักที่สามารถใช้ได้
สูงมาก	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักน้อยกว่าหรือเท่ากับ 85% ของหน่วยความจำหลักที่สามารถใช้ได้
สูงที่สุด	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักน้อยกว่าหรือเท่ากับ 95% ของหน่วยความจำหลักที่สามารถใช้ได้

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม STOR

คำถาม	ระดับคะแนน				
	5	4	3	2	1
ขนาดของหน่วยความจำหลักที่ท่านคาดว่าซอฟต์แวร์จะใช้ ☐ในกรณีที่ท่านทราบ (1) ขนาดของหน่วยความจำที่ระบบมีให้กับซอฟต์แวร์.....KB (2) ขนาดของหน่วยความจำที่คาดว่าซอฟต์แวร์ต้องการใช้.....KB					
☐ในกรณีที่ท่านไม่ทราบ (1) ขนาดของหน่วยความจำหลักที่ซอฟต์แวร์จองไว้ใช้งาน.....KB (2) ขนาดของหน่วยความจำที่คาดว่าจะใช้เก็บข้อมูลที่ใช้งานกับซอฟต์แวร์.....KB (3) ขนาดของหน่วยความจำที่ซอฟต์แวร์ใช้ร่วมกับซอฟต์แวร์อื่น.....KB (4) ขนาดของหน่วยความจำของเครื่องคอมพิวเตอร์ที่แนะนำ.....KB					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

2.3.การเปลี่ยนแปลงได้ง่ายของแพลทฟอร์ม (Platform Volatility: PVOL)

คำนิยามศัพท์

การเปลี่ยนแปลงได้ง่ายของแพลทฟอร์มหมายถึง ซอฟต์แวร์ที่จะพัฒนาต้องถูกออกแบบหรือพัฒนาให้มีความคล่องตัวสูงในการเปลี่ยนหรือปรับปรุงให้เข้ากับแพลทฟอร์มใหม่ ซึ่งถ้าซอฟต์แวร์ที่จะพัฒนาต้องทำการเปลี่ยนบ่อยๆ ก็ทำให้เกิดความความยากในการพัฒนา หรือ ใช้เวลามากในการพัฒนาเพื่อให้ซอฟต์แวร์สามารถทำงานบนหลายๆแพลทฟอร์มได้ เป็นต้น แพลทฟอร์มในที่นี้จะมีความหมายรวมทั้งฮาร์ดแวร์และซอฟต์แวร์(OS,DBMS etc) ตัวอย่างเช่น ถ้าซอฟต์แวร์ที่จะพัฒนาเป็นระบบปฏิบัติการ แพลทฟอร์มคือ อุปกรณ์คอมพิวเตอร์ ถ้าซอฟต์แวร์ที่จะพัฒนาเป็นระบบจัดการฐานข้อมูล(database management) แพลทฟอร์มคือฮาร์ดแวร์และระบบปฏิบัติการ ถ้าซอฟต์แวร์ที่จะพัฒนาคือระบบบัญชี แพลทฟอร์มคือ ระบบปฏิบัติการ ระบบจัดการฐานข้อมูล และฮาร์ดแวร์ ถ้าซอฟต์แวร์ที่จะพัฒนาคือตัวค้นหาข้อความบนเครือข่าย(network text browser) แพลทฟอร์มคือ ฮาร์ดแวร์ ระบบปฏิบัติการ และคลังเก็บข้อมูลแบบกระจาย (distribute information repositories) นอกจากนี้แพลทฟอร์มยังรวมไปถึงตัวแปลงภาษา(compiler) ด้วย

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำ	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 12 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 1 เดือน
ปานกลาง	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 6 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 2 สัปดาห์
สูง	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 1 สัปดาห์
สูงมาก	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 สัปดาห์ หรือทำการปรับปรุงบางส่วนทุกๆ 2 วัน

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม PVOL

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ท่านมีนโยบายที่จะทำการเปลี่ยนแพลทฟอร์มของซอฟต์แวร์ที่จะพัฒนาเป็นประจำในช่วงระยะเวลาใดเวลาหนึ่ง ใช่หรือไม่ เช่น ใช้เวลาในการเปลี่ยนเวอร์ชั่นของแพลทฟอร์มทุกๆ 6 เดือน หรือ ใช้เวลาในการเปลี่ยนชนิดของแพลทฟอร์มทุกๆ 12 เดือน เป็นต้น					
2) โดยเฉลี่ยจะใช้งานแพลทฟอร์มเป็นเวลาประมาณ.....เดือนก่อนที่จะทำการเปลี่ยนแปลงในแต่ละครั้ง					
3) ท่านจะต้องทำการปรับปรุงแพลทฟอร์มเพียงบางส่วนเป็นประจำในช่วงระยะเวลาใดเวลาหนึ่ง เพื่อให้ซอฟต์แวร์สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น หรือสามารถรองรับการใช้งานได้ ใช่หรือไม่ เช่น การขยายความสามารถในการใช้งานของผู้ให้บริการทางอินเตอร์เน็ต(Internet Service Provider) โดยการจัดการกับพื้นที่ในหน่วยความจำสำรองทุกสัปดาห์ เป็นต้น					
4) โดยเฉลี่ยใช้งานแพลทฟอร์มเป็นเวลาประมาณ.....สัปดาห์ก่อนที่จะทำการเปลี่ยนแปลงในแต่ละครั้ง					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

3)ปัจจัยกลุ่มบุคลากร

ปัจจัยที่เกี่ยวข้องกับบุคลากรหรือทีมที่จะพัฒนาซอฟต์แวร์ซึ่งแบ่งออกเป็น 6 ประเภท คือ ความสามารถในการวิเคราะห์ ความสามารถของโปรแกรมเมอร์ ประสบการณ์การใช้แอปพลิเคชั่น ประสบการณ์การใช้แพลทฟอร์ม ประสบการณ์การใช้เครื่องมือ และความต่อเนื่องของบุคลากรที่พัฒนาซอฟต์แวร์

3.1.ความสามารถในการวิเคราะห์ (Analyst Capability: ACAP)

คำจำกัดความ

ตัวขับค่าใช้จ่ายประเภทนี้จะพิจารณาจากความสามารถของบุคลากรในการวิเคราะห์ระบบงานตามที่ร้องขอ (request) และการออกแบบซอฟต์แวร์อย่างรายละเอียด(Detailed Design) เช่น การออกแบบเกี่ยวกับข้อมูล (Data Design) การออกแบบเกี่ยวกับโครงสร้าง(Architectural Design) การออกแบบเกี่ยวกับกระบวนการ (Procedural Design) การออกแบบเกี่ยวกับการเชื่อมต่อ (Interface Design) เป็นต้น ปัจจัยหลักที่ใช้ในการแบ่งระดับความสามารถคือ ความสามารถ(ability)ในการวิเคราะห์และออกแบบ ประสิทธิภาพ(efficiency) และความละเอียด(thoroughness)ในการวิเคราะห์และออกแบบ และความสามารถในการสื่อสารกับผู้อื่น ใน

การกำหนดระดับจะไม่นำประสบการณ์ในการวิเคราะห์มาพิจารณาด้วยเนื่องจากได้แยกออกเป็นอีกหนึ่งตัวขับค่าใช้จ่าย การกำหนดระดับของความสามารถในการวิเคราะห์จะพิจารณาความสามารถโดยเฉลี่ยของนักวิเคราะห์ระบบในทีมว่ามีความสามารถอยู่ในตำแหน่งเปอร์เซ็นต์ไทล์ที่เท่าใด

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25
ต่ำ	เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45
ปานกลาง	เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65
สูง	เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85
สูงมาก	เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม ACAP

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) จำนวนนักวิเคราะห์ระบบในทีมที่พัฒนาซอฟต์แวร์.....คน					
2) ท่านคิดว่านักวิเคราะห์ระบบแต่ละท่านในทีมมีความสามารถอยู่ในระดับใดบ้าง (มีความสามารถมากที่สุด = 5, มีความสามารถมาก = 4, มีความสามารถปานกลาง = 3, มีความสามารถน้อย = 2, มีความสามารถน้อยที่สุด = 1) นักวิเคราะห์ระบบคนที่ 1 มีความสามารถอยู่ในระดับ..... นักวิเคราะห์ระบบคนที่ 2 มีความสามารถอยู่ในระดับ..... นักวิเคราะห์ระบบคนที่ 3 มีความสามารถอยู่ในระดับ..... ฯลฯ					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

3.2.ความสามารถของโปรแกรมเมอร์ (Programmer Capability: PCAP)

คำจำกัดความ

แม้ว่ามีการให้ความสำคัญกับการวิเคราะห์ระบบอย่างมาก แต่เครื่องมือในปัจจุบันก็ได้พัฒนามีจนมีความซับซ้อนมากขึ้น ดังนั้นการใช้งานเครื่องมือเหล่านี้ก็ขึ้นกับความสามารถของโปรแกรมเมอร์ การจะกำหนดระดับให้นั้นควรจะพิจารณาบนพื้นฐานของความสามารถโดยรวมของโปรแกรมเมอร์ในทีมที่พัฒนามากกว่าที่จะพิจารณาโปรแกรมเมอร์เพียงคนเดียว ปัจจัยร่วมที่มีส่วนช่วยในการกำหนดระดับคือ ความสามารถในการ

พัฒนาซอฟต์แวร์ ประสิทธิภาพของซอฟต์แวร์ ความสมบูรณ์ของซอฟต์แวร์ที่พัฒนา และความสามารถในการติดต่อสื่อสารกับผู้อื่น ตัวขับค่าใช้จ่ายประเภทนี้จะไม่นำประสบการณ์ในการเขียนโปรแกรมมาพิจารณาด้วย เนื่องจากได้แยกออกเป็นอีกหนึ่งตัวขับค่าใช้จ่าย การกำหนดระดับจะพิจารณาจากความสามารถเฉลี่ยในการเขียนโปรแกรมของทีมโปรแกรมเมอร์ว่าอยู่ในตำแหน่งของเปอร์เซ็นต์ไทล์ที่เท่าใด

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25
ต่ำ	เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45
ปานกลาง	เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65
สูง	เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85
สูงมาก	เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม PCAP

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) จำนวนโปรแกรมเมอร์ในทีมที่พัฒนาซอฟต์แวร์.....คน					
2) ท่านคิดว่าโปรแกรมเมอร์แต่ละท่านในทีมมีความสามารถอยู่ในระดับใด ใน5 ระดับนี้ (มีความสามารถมากที่สุด = 5, มีความสามารถมาก = 4, มีความสามารถปานกลาง = 3,มีความสามารถน้อย = 2, มีความสามารถน้อยที่สุด =1) โปรแกรมเมอร์คนที่ 1 มีความสามารถอยู่ในระดับ..... โปรแกรมเมอร์คนที่ 2 มีความสามารถอยู่ในระดับ..... โปรแกรมเมอร์คนที่ 3 มีความสามารถอยู่ในระดับ..... ฯลฯ					

<u>ข้อเสนอแนะอื่นๆ</u>

.....

.....

.....

3.3.ประสบการณ์การพัฒนาแอปพลิเคชั่น (Application Experience: AEXP)

<u>คำจำกัดความ</u>

การกำหนดระดับของตัวขับค่าใช้จ่ายประเภทนี้ขึ้นอยู่กับประสบการณ์ในการพัฒนาซอฟต์แวร์(แอปพลิเคชั่น)ชนิดเดียวกับซอฟต์แวร์ที่จะพัฒนา ซึ่งในตัวขับค่าใช้จ่ายนี้จะพิจารณาจากระยะเวลาเฉลี่ยที่ทีมพัฒนาเคยใช้พัฒนาซอฟต์แวร์ประเภทเดียวกับซอฟต์แวร์ที่จะพัฒนา

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ3 ปี
สูงมาก	มากกว่า 3 ปี

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม AEXP

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน					
2) ประสบการณ์ในการพัฒนาซอฟต์แวร์ชนิดเดียวกับซอฟต์แวร์ที่จะพัฒนาของแต่ละท่าน บุคลากรคนที่ 1 มีประสบการณ์.....เดือน บุคลากรคนที่ 2 มีประสบการณ์.....เดือน บุคลากรคนที่ 3 มีประสบการณ์.....เดือน ฯลฯ					

<u>ข้อเสนอแนะอื่นๆ</u>

.....

.....

.....

3.4.ประสบการณ์การใช้แพลทฟอร์ม(Platform Experience: PEXP)

<u>คำจำกัดความ</u>

ตัวขับค่าใช้จ่ายนี้จะพิจารณาจากการทีมที่พัฒนามีประสบการณ์ในการใช้แพลทฟอร์มได้เป็นอย่างดี ซึ่งแพลทฟอร์มในที่นี้รวมทั้งฮาร์ดแวร์และซอฟต์แวร์ เช่น มีการรวมรูปภาพ(graphic)กับตัวเชื่อมประสาน(user interface) มีการใช้ฐานข้อมูล(database) มีการใช้ระบบเครือข่าย(networking) เป็นต้น ซึ่งในตัวขับค่าใช้จ่ายนี้จะพิจารณาจากระยะเวลาที่ทีมพัฒนาเคยใช้แพลทฟอร์มประเภทเดียวกับแพลฟอร์มที่จะใช้ในซอฟต์แวร์ที่จะพัฒนา

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม PEXP

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน					
2) ประสบการณ์ในการใช้แพลทฟอร์มชนิดเดียวกับแพลทฟอร์มที่จะใช้พัฒนาซอฟต์แวร์ของแต่ละท่าน บุคลากรคนที่ 1 มีประสบการณ์.....เดือน บุคลากรคนที่ 2 มีประสบการณ์.....เดือน บุคลากรคนที่ 3 มีประสบการณ์.....เดือน ฯลฯ					

<u>ข้อเสนอแนะอื่นๆ</u>

.....

.....

.....

3.5.ประสบการณ์การใช้ภาษาโปรแกรมและเครื่องมือ(Language and Tool Experience: LTEX)

<u>คำจำกัดความ</u>

ตัวขับค่าใช้จ่ายนี้เป็นการวัดประสบการณ์ในการใช้ภาษาโปรแกรม(programming language) และใช้เครื่องมือในการพัฒนาซอฟต์แวร์ การพัฒนาซอฟต์แวร์ยังรวมถึงการใช้งานเครื่องมือพัฒนาซอฟต์แวร์ในการวิเคราะห์และออกแบบ การจัดการโครงแบบ(configuration management) การจัดทำเอกสารการพัฒนาซอฟต์แวร์ การจัดการไลบรารี่(library management) ของโปรแกรมภาษา การกำหนดรูปแบบการเขียนโปรแกรม(program style)ที่เป็นมาตราฐานเดียวกัน และการตรวจสอบความสอดคล้อง(consistency checking)ของโปรแกรม เป็นต้น นอกจากนี้การมีประสบการณ์ในการเขียนโปรแกรมในภาษาใดภาษาหนึ่งจะมีผลต่อระยะเวลาที่ใช้ในการพัฒนาด้วย

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม LTEX

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน					
2) ประสบการณ์ในการใช้ภาษาโปรแกรม(Language Programming)ชนิดเดียวกับที่จะใช้พัฒนาซอฟต์แวร์ของโปรแกรมเมอร์แต่ละท่าน บุคลากรคนที่ 1 มีประสบการณ์.....เดือน บุคลากรคนที่ 2 มีประสบการณ์.....เดือน บุคลากรคนที่ 3 มีประสบการณ์.....เดือน ฯลฯ					
3) ประสบการณ์ในการใช้เครื่องมือ(Tool)ชนิดเดียวกับที่จะใช้พัฒนาซอฟต์แวร์ของบุคลากรแต่ละท่าน บุคลากรคนที่ 1 มีประสบการณ์.....เดือน บุคลากรคนที่ 2 มีประสบการณ์.....เดือน บุคลากรคนที่ 3 มีประสบการณ์.....เดือนฯลฯ					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

.....

3.6.ความต่อเนื่องของบุคลากร(Personnel Continuity: PCON)

คำจำกัดความ

ความต่อเนื่องของบุคลากรหมายถึงการหมุนเวียนของบุคลากรที่อยู่ในทีมพัฒนาซอฟต์แวร์ตั้งแต่เริ่มพัฒนาจนกระทั่งเสร็จสมบูรณ์ โดยถ้ามีการหมุนเวียนเพียงเล็กน้อย ซอฟต์แวร์ก็น่าจะมีความต่อเนื่องในการพัฒนาสูง บุคลากรในที่นี้รวมถึงบุคคลในทุกตำแหน่งที่ร่วมอยู่ในทีมพัฒนา เช่น ผู้จัดการโครงการ ผู้ช่วยผู้จัดการโครงการ นักวิเคราะห์ระบบ โปรแกรมเมอร์ พนักงานทั่วไป เลขานุการ เป็นต้น

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 48%ต่อปี
ต่ำ	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 24%ต่อปี แต่น้อยกว่า48%ต่อปี
ปานกลาง	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 12%ต่อปี แต่น้อยกว่า24%ต่อปี
สูง	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 6%ต่อปี แต่น้อยกว่า12%ต่อปี
สูงมาก	บุคลากรมีการเปลี่ยนแปลงน้อยกว่า 6%ต่อปี

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม PCON

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) จำนวนบุคลากรที่พัฒนาเมื่อเริ่มการพัฒนาซอฟต์แวร์.....คน					
2) ท่านมีนโยบายที่จะรับบุคลากรเข้าทีมพัฒนาเพิ่มอีก.....คน					
3) ท่านมีนโยบายให้บุคลากรลาออกได้.....คน					
4) บุคลากรที่รับเข้ามาใหม่จะทำงานแทนบุคลากรคนเก่าที่ลาออกไป...คน					
5) ซอฟต์แวร์มีกำหนดเวลาของการพัฒนาประมาณ.....ปี					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

.....

4)ปัจจัยกลุ่มโครงการ

ปัจจัยที่เป็นภาพรวมของซอฟต์แวร์ที่ต้องการพัฒนา ซึ่งจะแบ่งออกเป็น 3 ประเภท คือการใช้เครื่องมือ สถานที่ที่จะใช้พัฒนาซอฟต์แวร์ และกำหนดการการพัฒนา

4.1.การใช้เครื่องมือซอฟต์แวร์ (Use of software tool: TOOL)

คำจำกัดความ

ความสามารถหรือประสิทธิภาพของเครื่องมือซอฟต์แวร์ได้พัฒนาจนเป็นที่ยอมรับว่ามีผลต่อการพัฒนาโครงการ เนื่องจากเครื่องมือในสมัยแรกๆสามารถทำได้เพียงการบรรณาธิกร(edit)และเขียน(code)โปรแกรม แต่ต่อมาเครื่องมือได้มีการพัฒนาจนสามารถนำมาใช้ในทุกขั้นของวงจรชีวิตของการพัฒนาได้ เช่นขั้นตอนการวิเคราะห์ (Analysis) ขั้นตอนออกแบบ(Design) และขั้นตอนการทำให้เกิดผล(Implementation)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	เครื่องมือสามารถสามารถเขียน(Code) บรรณาธิกร(Edit) และตรวจสอบความผิดพลาดในการเขียนโปรแกรมได้(Debug)
ต่ำ	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำฟ้อนเอนด์เคส (Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงแรกๆ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา (Requirements Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase) หรือ ช่วยในขั้นตอนการออกแบบ(Design Phase) หรือแบบเครื่องมือเคสที่ทำแบล็กเอนด์เคส (Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงหลัง เช่น ช่วยในขั้นตอนของการทำให้เกิดผล(Implementation Phase) ช่วยในขั้นตอนของการรวม(Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา(Maintenance Phase)
ปานกลาง	เครื่องมือที่ใช้พัฒนาซอฟต์แวร์สามารถใช้ได้ในทุกขั้นตอนของวงจรชีวิต ในขั้นพื้นฐานได้ นอกจากนี้ยังสามารถทำการรวมบางขั้นตอนของวงจรชีวิตได้
สูง	เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นของวงจรชีวิตและยังสามารถทำการรวมทุกขั้นของวงจรชีวิตได้
สูงมาก	เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นตามรูปแบบของวงจรชีวิต) นอกจากนี้ยังสามารถทำการรวมกัน(Integrated)โดยระเบียบวิธีการ(Method) กระบวนการ(Process) หรือการนำกลับมาใช้

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม TOOL

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นภาษาโปรแกรม เช่น ภาษา C ,Visual Basic, Pascal, Java เป็นต้น ใช่หรือไม่					
2) เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นเครื่องมือเคส (Computer Aided Software Engineering :CASE) ใช่หรือไม่					
3) เครื่องมือที่ใช้นอกจากจะเป็นแบบเครื่องมือเคสที่ช่วยในการพัฒนาซอฟต์แวร์ได้ทุกขั้นตอนของวงจรชีวิต(SDLC)แล้วยังสามารถนำไปใช้รวม(Integrate)กับซอฟต์แวร์อื่นๆ ได้ ใช่หรือไม่					
4) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ช่วยในหลายขั้นตอนของวงจรชีวิต(SDLC) เช่น ช่วยในการเก็บรวบรวมข้อมูล(Required Phase)และนำข้อมูลที่ได้มาวิเคราะห์และออกแบบการพัฒนาซอฟต์แวร์(Design Phase) หรือ ช่วยออกแบบซอฟต์แวร์(Design Phase)แล้วนำไปสร้างโปรแกรม(Code Generator) เป็นต้น ใช่หรือไม่					
5) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำฟ้อนเอนด์เคส (Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงแรกๆ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา(Required Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase) หรือ ช่วยในขั้นตอนการออกแบบ(Design Phase) ใช่หรือไม่					
6) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำแบล็กฟ้อนเอนด์เคส (Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงหลัง เช่น ช่วยในขั้นตอนของการทำให้เกิดผล(Implementation Phase) ช่วยในขั้นตอนของการรวม (Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา(Maintenance Phase) ใช่หรือไม่					
7) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสแบบการรวม(Integrated CASE) ซึ่งสามารถทำการรวม(Integrated) แบบกระบวนการ(Process) วิธีการ(Method) หรือการใช้ใหม่ (Reuse) ใช่หรือไม่					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

.....

4.2.การพัฒนาในหลายสถานที่ (Multisite Development: SITE)

ทำจำกัดความ

ตัวขับค่าใช้จ่ายประเภทนี้ได้ถูกพิ่มเข้ามาในแบบจำลองโคโคโม2 เนื่องจากการวิจัยพบว่าสถานที่ที่ใช้พัฒนาและอุปกรณ์ที่ใช้ในการสื่อสารระหว่างบุคลากรในที่มพัฒนา มีผลต่อการพัฒนาซอฟต์แวร์ เช่น สถานที่ที่ห่างไกลกันหรืออุปกรณ์การสื่อสารที่ไม่ทันสมัยจะทำให้ไม่สามารถแก้ไขปัญหาได้ทันท่วงที เนื่องจากความล่าช้าในการติดต่อ

ตารางอธิบายความแตกต่างของแต่ละระดับเกี่ยวกับสถานที่

ระดับ	เงื่อนไขของสถานที่
ต่ำมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีทั้งในประเทศและต่างประเทศ
ต่ำ	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด และ มีบริษัทที่พัฒนาหลายบริษัท
ปานกลาง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด หรือ มีบริษัทที่พัฒนาหลายบริษัท หรือ ตั้งอยู่ในจังหวัดที่ไม่สำคัญเพียงจังหวัดเดียว เช่น หนองบัวลำพู ปัตตานี เป็นต้น
สูง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในกรุงเทพฯ หรือจังหวัดสำคัญๆ(กรุงเทพฯ เชียงใหม่ นครราชสีมา ภูเก็ต สงขลา เป็นต้น)
สูงมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีหลายตึก หรือ เป็นหมู่ตึก
สูงที่สุด	สถานที่ที่พัฒนาซอฟต์แวร์จะอยู่ในบริเวณเดียวกัน เช่น อยู่ในตึกเดียวกัน อยู่ในชั้นเดียวกัน เป็นต้น

ตารางอธิบายความแตกต่างของแต่ละระดับการสื่อสารระหว่างการพัฒนา

ระดับ	เงื่อนไขของการสนับสนุนด้านการสื่อสาร
ต่ำมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์รวม หรือ จดหมาย
ต่ำ	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์ส่วนตัวหรือแฟกส่วนตัว
ปานกลาง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณแคบ เช่น ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น
สูง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น
สูงมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์
สูงที่สุด	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ มัลติมีเดียที่สามารถโต้ตอบได้

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม SITE

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) ซอฟต์แวร์จะถูกพัฒนาภายในประเทศเท่านั้น ใช่หรือไม่					
2) ซอฟต์แวร์จะถูกพัฒนาภายในจังหวัดเดียวกัน ใช่หรือไม่					
3) ซอฟต์แวร์จะถูกพัฒนาภายในบริษัทเดียวหลายบริษัท ใช่หรือไม่					
4) ซอฟต์แวร์จะถูกพัฒนาในกรุงเทพฯหรือจังหวัดสำคัญๆ ใช่หรือไม่					
5) ซอฟต์แวร์จะถูกพัฒนาภายในอาคารเดียวกันหรือหลายอาคารที่อยู่ในบริเวณใกล้เคียงกัน ใช่หรือไม่					
6) ซอฟต์แวร์จะถูกพัฒนาภายในบริเวณเดียวกัน ห้องเดียวกัน หรือชั้นเดียวกัน ใช่หรือไม่					
7) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ คือ มัลติมีเดียที่สามารถได้ตอบได้ ใช่หรือไม่					
8) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์ ใช่หรือไม่					
9) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น ใช่หรือไม่					
10) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณแคบ ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น ใช่หรือไม่					
11) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้โทรศัพท์ส่วนตัว หรือแฟกส่วนตัว ใช่หรือไม่					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

4.3.กำหนดเวลาของการพัฒนาที่ต้องการ (Required Development Schedule: SCED)

ข้อจำกัดความ

การกำหนดระดับจะขึ้นอยู่กับข้อจำกัดของทีมที่พัฒนา ดังนั้นจึงจะพิจารณาเป็นทีมที่พัฒนา โดยจะเทียบเป็นเปอร์เซ็นต์ของการขยายเวลาการทำงานออกไปหรือการเร่งเวลาการทำงานให้เร็วขึ้นเมื่อเทียบกับเวลาการทำงานปกติ ซึ่งการเร่งตารางการทำงานมีแนวโน้มว่าจะใช้ความพยายามสูงในเฟส(phase)ท้ายๆของการพัฒนา เนื่องจากอาจมีปัญหาหรือข้อสงสัยหลายอย่างที่ไม่ได้ถูกพิจารณาหรือแก้ไขในเฟสแรกๆ ดังนั้นจึงต้องมาทำในเฟสท้ายๆ และการขยายเวลาการทำงานออกไปมีแนวโน้มว่าจะใช้ความพยายามในเฟสแรกๆ

มาก เนื่องจากใช้ในการวางแผนการพัฒนาซอฟต์แวร์(Planning) การกำหนดคุณลักษณะ(Specifications)ของซอฟต์แวร์ที่จะพัฒนา และการวางแผนการตรวจสอบความถูกต้อง(Validation)ของซอฟต์แวร์

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	เวลาในการพัฒนาน้อยกว่าหรือเท่ากับ 75% ของเวลาปกติ
ต่ำ	เวลาในการพัฒนามากกว่า 75%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ 85%
ปานกลาง	เวลาในการพัฒนามากกว่า 85%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ100%
สูง	เวลาในการพัฒนามากกว่า 100%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ130%
สูงมาก	เวลาในการพัฒนามากกว่า 130%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ160%

ให้ผู้เชี่ยวชาญกำหนดน้ำหนักของความครอบคลุมและสอดคล้องของคำถาม SCED

คำถาม	ระดับคะแนน				
	5	4	3	2	1
1) หลังจากที่ท่านได้เก็บรวบรวมความต้องการ(Requirement) ต่างๆจากผู้ใช้แล้ว ท่านคาดว่าจะใช้เวลาในการพัฒนาซอฟต์แวร์ประมาณ .....เดือน					
2) เวลาที่ท่านบวกเผื่อไว้(กรณีใดๆ)ประมาณ.....เดือน					

ข้อเสนอแนะอื่นๆ

.....

.....

.....

.....

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบสอบถามรอบที่ 2

เรื่อง การสร้างชุดคำถามเพื่อใช้ในการประมาณค่าใช้จ่ายของการพัฒนาซอฟต์แวร์ โดยใช้แบบจำลองไดไดโม2 ตามทัศนะของผู้เชี่ยวชาญ

วัตถุประสงค์

เพื่อให้ผู้เชี่ยวชาญยืนยันความคิดเห็นเกี่ยวกับชุดคำถามที่ได้แสดงไว้ในแบบสอบถามรอบที่ 1

ขั้นตอนและการดำเนินการ

1. ระดับคะแนนมีความหมายดังนี้

5 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับมากที่สุด

4 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับมาก

3 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับปานกลาง

2 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับน้อย

1 หมายถึง ชุดคำถามมีความครอบคลุมและสอดคล้องกับส่วนที่เป็นคำจำกัดความและส่วนที่เป็นตารางอธิบายความแตกต่างแต่ละระดับน้อยที่สุด

2. ความหมายของสัญลักษณ์มีดังนี้

\+ หมายถึง ค่ามัธยฐานของระดับคะแนนจากคำตอบในรอบที่ 1

⌊⌋ หมายถึง ขอบข่ายของพิสัยระหว่างควอไทล์ที่ 1 และ ควอไทล์ที่ 3 ของระดับคะแนน จากคำตอบในรอบที่ 1

△ หมายถึง คำตอบของท่านในรอบที่ 1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่าง

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1)ซอฟต์แวร์ที่พัฒนาจะถูกนำไปใช้ในด้านที่เกี่ยวข้องกับชีวิตมนุษย์หรือในด้านที่มีผลต่อชีวิตมนุษย์ ใช่หรือไม่	↿	+ ⇂	△ ✓			มีซอฟต์แวร์ประเภทนี้น้อยมาก

จากตัวอย่าง หมายความว่า คำถามข้อที่ 1 ในรอบที่แล้วคำตอบท่านอยู่ในระดับคะแนนที่ 3 (△) และในรอบนี้ ท่านยืนยันคำตอบเดิม ในช่องระดับคะแนนที่ 3 (✓) ซึ่งอยู่นอกช่วงพิสัยระหว่างระหว่างควอไทล์ที่ 1 และ ควอไทล์ที่ 3 (↿⇂) ท่านจึงให้เหตุผลในช่องเหตุผล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบตอบถามรอบที่ 2

1 ปัจจัยกลุ่มผลิตภัณฑ์

ปัจจัยที่เกี่ยวข้องกับคุณลักษณะของซอฟต์แวร์ที่ต้องการจะพัฒนา ซึ่งแบ่งออกเป็น 5 ประเภท ดังนี้ ความน่าเชื่อถือของซอฟต์แวร์ ขนาดของซอฟต์แวร์ต่อขนาดของข้อมูล ความซับซ้อนของซอฟต์แวร์ การนำซอฟต์แวร์กลับมาใช้ และการจัดทำเอกสารประกอบการพัฒนา

1.1.ความน่าเชื่อถือของซอฟต์แวร์ที่ต้องการ (Required Software Reliability: RELY)

คำจำกัดความ

ความน่าเชื่อถือของซอฟต์แวร์หมายถึงซอฟต์แวร์ต้องสามารถทำงานได้ในช่วงเวลาที่กำหนดอย่างต่อเนื่องและถูกต้อง โดยไม่มีเหตุผิดปกติต่างๆ เช่น การคำนวณผิดพลาด การไม่ทำงานตามขั้นตอน การหยุดทำงานโดยที่ยังไม่จบการทำงานหรือไม่ถูกสั่งให้หยุด เป็นต้น โดยการพิจารณาความน่าเชื่อถือของผลิตภัณฑ์ซอฟต์แวร์ตามแนวทางของโคโคโม2 จะพิจารณาจากความเสียหายที่เกิดขึ้นหลังจากซอฟต์แวร์ไม่สามารถทำงานได้ เพื่อประเมินระดับของความพยายามที่จะใช้ในการพัฒนาซอฟต์แวร์

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำที่สุด	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความไม่สะดวกในการทำงานแต่สามารถดำเนินงานต่อไปได้
ต่ำ	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายเพียงเล็กน้อยในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม
ปานกลาง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายปานกลางในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม
สูง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความสูญเสียทางการเงินอย่างมาก
สูงมาก	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความเสี่ยงต่อชีวิตมนุษย์

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท RELY

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์ที่พัฒนาจะถูกนำไปใช้ในด้านที่อาจเกิดความเสี่ยงกับชีวิตมนุษย์ หรือมีผลกระทบต่อชีวิตมนุษย์ ใช่หรือไม่		+ ↑				
2) ซอฟต์แวร์ที่จะพัฒนาเป็นซอฟต์แวร์ที่เกี่ยวข้องกับรายรับ-รายจ่ายขององค์กรเป็นอย่างมาก ซึ่งเมื่อซอฟต์แวร์ไม่สามารถทำงานได้จะกระทบต่อรายรับ-รายจ่ายขององค์กร ใช่หรือไม่		+ ↑				

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท RELY(ต่อ)

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
3) ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติ ท่านจำเป็นต้องดำเนินการให้มีการแก้ไขทันที ใช่หรือไม่		+				
4) ท่านสามารถดำเนินงานในลักษณะนั้นต่อได้ ถึงแม้ว่าซอฟต์แวร์จะไม่สามารถทำงานได้ก็ตาม ใช่หรือไม่ เช่นการใช้บุคลากรทำงานแทน เป็นต้น		+				

1.2.ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ (Data Base Size : DATA)

<u>คำจำกัดความ</u>

ตัวขับค่าใช้จ่ายประเภทนี้จะใช้สมการในการคำนวณคือ D/P

โดย D คือ ขนาดของฐานข้อมูล(ไบต์)

P คือ ขนาดของซอฟต์แวร์(จำนวนบรรทัดของโปรแกรม)

ซึ่งจะพบว่าขนาดของฐานข้อมูลมีความสำคัญสำหรับการพิจารณา เนื่องจากถ้าขนาดของฐานข้อมูลมีขนาดใหญ่มากก็จะทำให้ได้ค่าจาก D/P มาก ซึ่งหมายถึงว่าข้อมูลที่ใช้ในการทดสอบซอฟต์แวร์ก็ต้องมีขนาดใหญ่มากขึ้นด้วย

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำ	D/P น้อยกว่า 10
ปานกลาง	D/P มากกว่าเท่ากับ 10 แต่น้อยกว่า 100
สูง	D/P มากกว่าเท่ากับ 100 แต่น้อยกว่า 1000
สูงมาก	D/P มากกว่า 1000

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท_DATA

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ขนาดของหน่วยความจำสำรองที่ใช้เก็บข้อมูล(กิโลไบต์:KB)		+				
☐ในกรณีที่ท่านทราบ						
ขนาดของหน่วยความจำสำรองที่ใช้เก็บข้อมูล						
ประมาณ.....KB	↳	↴				
☐ในกรณีที่ท่านไม่ทราบ						
ให้ท่านประมาณจำนวนแฟ้มข้อมูล(Files) จำนวนรายการ (Transactions) จำนวนเขตข้อมูล(Fields) และจำนวนอักษร						
Files จำนวนRecord ความยาวของRecord						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
ฯลฯ						

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท DATA(ต่อ)

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
2) ขนาดของซอฟต์แวร์ที่จะพัฒนา(จำนวนบรรทัดของโปรแกรม: SLOC)		+				
☐ในกรณีที่ท่านทราบ ขนาดของซอฟต์แวร์ที่จะพัฒนามีจำนวนบรรทัดของโปรแกรมประมาณ.....บรรทัด		↑	↑			
☐ในกรณีที่ท่านไม่ทราบ(ประมาณโดยวิธีของFunction Point)						
(1) ภาษาที่ใช้พัฒนาซอฟต์แวร์จัดอยู่ในประเภทใด						
( ) Ada ( ) ANSI/QUICK/Turbo Basic ( ) AI						
( ) Basic Assembly ( ) Macro Assembly ( ) APL						
( ) Basic – Compiled ( ) Fourth-Generator						
( ) Basic – Interpreted ( ) ANSI Cobol 85 ( ) C						
( ) C++ ( ) Fortan77 ( ) Forth						
( ) High-Level ( )Jovial ( ) Lisp						
( ) Modula2 ( ) Object-Oriented ( ) Procedural Language						
( ) Pascal ( ) Program Generator ( ) Prolog						
( ) Report Generator ( ) Spreadsheet ( ) Shell APL						
( ) Query Language						
(2) ประเภทของแฟ้มของข้อมูลที่จะใช้กับซอฟต์แวร์						
(2.1)จำนวนชุดข้อมูลที่รับมาจากภายนอก (External Input: EI).....ชุด						
(2.2)จำนวนชุดข้อมูลที่ส่งออกไปภายนอก (External Output: EO) .....ชุด						
(2.3)จำนวนชุดข้อมูลภายในเชิงตรรกะ (Internal Logical File: ILF) .....ชุด						
(2.4)จำนวนชุดข้อมูลที่ต่อประสานกับภายนอก (External Interface File: EIF) .....ชุด						
(2.5)จำนวนชุดข้อมูลที่เป็นคำถามจากภายนอก (External Inquiry: EQ) .....ชุด						

1.3 ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์(Complexity :CPLX)

ข้อจำกัดความ

ในโคโคโม2จะกำหนดระดับของความซับซ้อนของโปรแกรมที่จะพัฒนา โดยจะแบ่งหัวข้อการพิจารณาออกเป็น 5 หัวข้อย่อยดังนี้

1.3.1 การดำเนินการควบคุม(Control Operation)

การดำเนินการควบคุมหมายถึง กระบวนการที่ใช้ควบคุมการทำงานของโปรแกรม หรือลักษณะการทำงานของโปรแกรมในซอฟต์แวร์ที่จะพัฒนา

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการ(programming operation)ของซอฟต์แวร์ ส่วนใหญ่จะเป็นการดำเนินการแบบตรงไปข้างหน้า (straight-line code) แต่ก็มีลักษณะการดำเนินการแบบเป็นเงื่อนไขแต่เป็นเงื่อนไขที่ไม่ซับซ้อน(non-nested structured) เช่น DO, CASE, IF-THEN-ELSE นอกจากนี้อาจมีการใช้โมดุลง่ายๆ เช่น การทำโปรซีเดอร์คอร์(procedure call) เป็นต้น
ต่ำ	การดำเนินการของซอฟต์แวร์จะมีโครงสร้างที่ซับซ้อน(nesting of structured programming) และโดยส่วนใหญ่จะเป็นแบบมีเงื่อนไข เช่น มีลูป WHILE ซ้อน IF-THEN-ELSE หรือ IF-THEN-ELSE ซ้อน IF-THEN-ELSE เป็นต้น
ปานกลาง	การดำเนินการของซอฟต์แวร์โดยส่วนใหญ่จะเป็นแบบมีเงื่อนไขที่ซับซ้อน จึงมีการใช้ตารางตัดสินใจ(decision table) หรือ มีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบ call back หรือแบบการส่งข้อความ(message passing) รวมทั้งมีการดำเนินการที่สนับสนุนการประมวลผลแบบกระจายบนเครือข่าย(distribute processing) เช่น ระบบclient/server
สูง	โครงสร้างของการดำเนินการมีความซับซ้อนมากซึ่งประกอบด้วยการดำเนินการที่เป็นแบบเงื่อนไขซ้อนเงื่อนไข หรือมีการใช้คิว(queue) และกองซ้อน(stack) เป็นตัวควบคุมลำดับการดำเนินการ หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน (homogeneous) คือ มีขี่ห้อหรือเวอร์ชันเดียวกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด(soft real time)
สูงมาก	การดำเนินการจะมีลักษณะเป็นแบบการเวียนบังเกิด(recursive) หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความความต่างกัน(heterogeneous) คือ มีขี่ห้อหรือเวอร์ชันต่างกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด (hard real time)
สูงที่สุด	การดำเนินการของซอฟต์แวร์จะต้องควบคุมการทำงานของฮาร์ดแวร์หรืออุปกรณ์โดยตรง (Microcode level control) หรือมีตารางการใช้ทรัพยากร(resource)ที่สามารถแก้ไขได้โดยอัตโนมัติตามความเหมาะสม หรือมีการประมวลผลแบบกระจาย(distribute processing) ที่ได้ผลลัพธ์แบบทันทีทันใด(hard real time)

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท การดำเนินการควบคุม

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์ที่พัฒนาต้องเขียนโปรแกรมเป็นภาษาเครื่องเพื่อควบคุมการทำงานหรือใช้งานอุปกรณ์โดยตรง ใช่หรือไม่		+				
2) ซอฟต์แวร์ที่จะพัฒนานี้สามารถประมวลแบบผลแบบกระจาย (distribute processing) เช่น การประมวลผลแบบ client/sever ใช่หรือไม่		+				
3) ซอฟต์แวร์ที่พัฒนามีการประมวลผลแบบทันทีทันใด(real time) ใช่หรือไม่		+				
4) ซอฟต์แวร์ที่พัฒนามีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน (homogeneous) คือ มียี่ห้อและเวอร์ชันเดียวกัน ใช่หรือไม่		+				
5) ซอฟต์แวร์ที่จะพัฒนามีการเขียนโปรแกรมลักษณะเป็นแบบการเวียนบังเกิด(recursion) ใช่หรือไม่		+				
6) ซอฟต์แวร์มีการจัดการกับเงื่อนไขจำนวนมากๆในแต่ละโปรซิเดอร์(Procedure) ใช่หรือไม่			+			
7) ซอฟต์แวร์ที่จะพัฒนามีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบcall back หรือแบบการส่งข้อความ(message passing) ใช่หรือไม่		+				
8) ซอฟต์แวร์ที่จะพัฒนามีส่วนของโปรแกรมเป็นลูปเงื่อนไขที่ซับซ้อน เช่น ลูป While ซ้อนกัน เป็นต้น ใช่หรือไม่		+				
9) ซอฟต์แวร์ที่จะพัฒนามีโครงสร้างที่ซับซ้อนจึงต้องมีการใช้ คิว (Queue) หรือกองซ้อน(Stack) เพื่อจัดลำดับการทำงานของโปรแกรม ใช่หรือไม่		+				

1.3.2 การดำเนินการคำนวณ(Computation Operation)

การดำเนินการคำนวณ หมายถึง วิธีการหรือรูปแบบการคำนวณที่มีใช้ในซอฟต์แวร์ที่จะพัฒนา

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการคำนวณสามารถหาค่าจากสมการพีชคณิตหรือสมการเส้นตรง เช่น ทำการบวก การลบ การคูณ การหาร ตัวอย่างเช่น A=B+C*(D-E)
ต่ำ	การดำเนินการคำนวณสามารถหาค่าจากสมการที่มีความยากระดับกลาง(moderate-level expressions) คือ นอกจากทำการบวก การลบ การคูณ การหาร แล้วยังสามารถทำการยกกำลัง การหารากการคำนวณหาค่าตรีโกณมิติ ตัวอย่างเช่น D= SQRT(B^2 – 4*A*C)
ปานกลาง	การดำเนินการคำนวณมีการใช้รูทีน(routine) ที่เป็นมาตราฐานทางคณิตศาสตร์ และสถิติ เช่น การหาส่วนเบี่ยงเบนมาตราฐาน การหาค่า Z เป็นต้น นอกจากนี้ยังคำนวณหาค่าของเมตริกซ์(matrix)และเวกเตอร์(vector)ได้
สูง	การดำเนินการคำนวณสามารถวิเคราะห์เชิงตัวเลข(numerical analysis)ขั้นพื้นฐานได้ เช่น การหาคำตอบของสมการโดยวิธีการของนิวตัน(Newton's Method) หรือโดยวิธีการของ Guassian การแก้สมการเชิงอนุพันธ์ เป็นต้น
สูงมาก	การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างแน่นอน เช่น การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน (non-singular matrix equation) การหาค่าจากสมการเชิงอนุพันธ์ย่อย(partial differential equation) และการหารูปแบบของความสัมพันธ์จากข้อมูล 2 ชุด
สูงที่สุด	การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างไม่แน่นอน เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน(highly accurate analysis of noisy) การวิเคราะห์ข้อมูลแบบ stochastic

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท การดำเนินการคำนวณ

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์ที่จะพัฒนาเกี่ยวข้องกับการวิเคราะห์โครงสร้างข้อมูลเชิงตัวเลขชนิดที่มีโครงสร้างที่ไม่แน่นอน ใช่หรือไม่ เช่น การวิเคราะห์ระดับความสูงของเยื่องรบกวน การวิเคราะห์สภาพอากาศ เป็นต้น		+ ↑↓				
2) ซอฟต์แวร์ที่จะพัฒนาจะต้องวิเคราะห์โครงสร้างเชิงตัวเลขได้จากชุดข้อมูล ใช่หรือไม่ เช่น การวิเคราะห์จุดสูงสุดจุดต่ำสุดของข้อมูล การหาสมการแทนการกระจายของชุดข้อมูล การหาค่าจากสมการเชิงอนุพันธ์ย่อย การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน เป็นต้น		+ ↑	↓			
3) ซอฟต์แวร์ที่จะพัฒนาสามารถวิเคราะห์เชิงตัวเลขขั้นพื้นฐานได้ ใช่หรือไม่ เช่น สามารถหาอนุพันธ์ การหาคำตอบของสมการโดยวิธีการของนิวตัน(Newton's Method) การหาพื้นที่ใต้กราฟจากการอินติเกรท(Integrate) เป็นต้น		+ ↑	↓			
4) ซอฟต์แวร์ที่จะพัฒนาต้องมีรูทีนที่เป็นมาตราฐานทางคณิตศาสตร์และสถิติ ใช่หรือไม่ เช่นการหาส่วนเบี่ยงเบนมาตราฐาน การหาค่าZ เป็นต้น	↑	+	↓			
5) ซอฟต์แวร์ที่จะพัฒนามีการหาคำตอบจากสมการพีชคณิตหรือสมการเส้นตรง เช่น การทำบวก ลบ คูณ หาร ใช่หรือไม่	↑	+ ↓				

1.3.3 การดำเนินการที่เกี่ยวกับอุปกรณ์(Device-dependent Operation)

การดำเนินการที่เกี่ยวกับอุปกรณ์ หมายถึง วิธีหรือกระบวนการที่ซอฟต์แวร์ที่จะพัฒนาใช้งานอุปกรณ์ เช่น การสั่งให้เครื่องพิมพ์เอกสารพิมพ์รายงาน การจัดเก็บข้อมูลลงในฮาร์ดดิส เป็นต้น

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์เป็นคำสั่งที่ใช้งานง่าย หรือเป็นภาษาระดับสูง เช่น การใช้คำสั่ง อ่าน(read) เขียน(write) ในภาษาพาสคาล(Pascal) ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล
ต่ำ	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์ สามารถใช้คำสั่งที่เป็นพื้นฐานที่ภาษามีการเตรียมไว้ให้ เช่น คำสั่งเขียนข้อมูลลงแฟ้มข้อมูล(put) หรือ อ่านข้อมูลในแฟ้มข้อมูล(get) ในภาษาซี(C) โดยที่ผู้พัฒนาไม่จำเป็นต้องมีความรู้เกี่ยวกับอุปกรณ์ที่ใช้

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ปานกลาง	ผู้พัฒนาจะต้องมีความรู้อย่างเจาะจงเกี่ยวกับอุปกรณ์ที่เป็นตัวรับข้อมูลและตัวแสดงผล (input/output) โดยการใช้งานอุปกรณ์ต้องทำ 3 ขั้นตอนดังนี้คือ เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์นั้นและตรวจสอบความผิดพลาดในการทำงานของอุปกรณ์
สูง	การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะกระทำในระดับกายภาพ (physical) เช่น การแปลงตำแหน่งของหน่วยความจำทางกายภาพ เพื่อช่วยในการค้นหาและการอ่านและการกำหนดรูปแบบการซ้อนทับ(overlap)ในหน่วยความจำของอุปกรณ์
สูงมาก	การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะมีรูทีน(routines) สำหรับจัดการเมื่อเกิดการผิดพลาดในขณะส่งและรับข้อมูล มีการจัดการเกี่ยวกับการส่งข้อมูลในสายการสื่อสาร หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบเข้มงวด(performance-intensive embedded system) เช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน
สูงที่สุด	มีการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed operations) หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบวิกฤต (performance-critical embedded systems) เช่น มีการบันทึกการทำงานของอุปกรณ์ทุกชนิดตลอดเวลา

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท ถามการดำเนินการที่เกี่ยวกับอุปกรณ์

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์สามารถใช้งานอุปกรณ์โดยการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed) ใช่หรือไม่	↓	+ ↑				
2) ซอฟต์แวร์ที่จะพัฒนามีระบบตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์ แบบเข้มงวด เช่นมีการบันทึกการทำงานของอุปกรณ์ทุกตัว หรือ แบบวิกฤตเช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน ใช่หรือไม่	↓	+ ↑				
3) ซอฟต์แวร์ที่จะพัฒนามีรูทีนรองรับการขัดจังหวะ(Interrupt routines) เมื่อเกิดเหตุผิดพลาดขณะที่ซอฟต์แวร์กำลังใช้งานอุปกรณ์ ใช่หรือไม่	↓	+ ↑				
4) ซอฟต์แวร์ที่จะพัฒนาจะต้องดำเนินการกับอุปกรณ์ที่เป็นตัวรับข้อมูลหรือตัวแสดงผลข้อมูล ในระดับกายภาพ เช่น การแปลงตำแหน่งเชิงกายภาพ(address) ซึ่งเป็นที่เก็บข้อมูลจริงๆในหน่วยความจำสำรองหรือหน่วยความจำหลัก ใช่หรือไม่		+ ↓	↑			

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท ถามการดำเนินการที่เกี่ยวกับอุปกรณ์(ต่อ)

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
5) ซอฟต์แวร์ที่จะพัฒนาจะต้องสามารถจัดการกับอุปกรณ์ได้ เช่น เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์ ตรวจสอบความผิดพลาดของอุปกรณ์ เป็นต้น ใช่หรือไม่		+			-	
6) ใช้ภาษาระดับสูงในการใช้งานอุปกรณ์ เช่น การใช้คำสั่ง อ่าน (read) เขียน(write) ในภาษาPascal ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล ใช่หรือไม่		+				

1.3.4 การดำเนินการเกี่ยวกับการจัดการกับข้อมูล(Data Management Operation)

การดำเนินการเกี่ยวกับการจัดการกับข้อมูล หมายถึง ซอฟต์แวร์ที่จะพัฒนามีวิธีการจัดการกับการเก็บข้อมูลและการใช้งานข้อมูล

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลมีการใช้แถวลำดับ(array)ในหน่วยความจำหลัก เพื่อเก็บข้อมูล และสามารถใช้สอบถาม(query)และแก้ไข(update)ได้
ต่ำ	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ แฟ้มข้อมูลสามารถใช้สอบถามและแก้ไขได้
ปานกลาง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลได้หลายแฟ้มข้อมูล โดยจะเป็นแฟ้มส่งข้อมูลออกเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ จะไม่มีการดำเนินการกับแฟ้มข้อมูลที่เกิดในระหว่างการทำงาน เช่น log file .
สูง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถทำทริกเกอร์(trigger)โดยใช้ข้อมูลเพียงชุดเดียว สามารถทำการปฏิวัติโครงสร้างของข้อมูลที่ซับซ้อน(complex data restructuring) เช่นการเปลี่ยนคีย์ การเปลี่ยนความสัมพันธ์ระหว่างแฟ้มข้อมูล เป็นต้น
สูงมาก	การดำเนินการเกี่ยวกับการจัดการฐานข้อมูลสามารถทำกับข้อมูลที่เก็บไว้ในหลายสถานที่ได้ สามารถทำทริกเกอร์(trigger)กับข้อมูลจำนวนมากๆได้ สามารถค้นหาข้อมูลในฐานข้อมูลได้
สูงที่สุด	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลเป็นแบบพลวัต(dynamic relational) คือสามารถเปลี่ยนแปลงโครงสร้างบางอย่างโดยอัตโนมัติตามสภาพของข้อมูลที่เก็บได้ มีโครงสร้างเชิงวัตถุ(object structures) มีการจัดการข้อมูลด้วยภาษาธรรมชาติ(natural language data management)

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท ฉากการดำเนินการที่เกี่ยวกับการจัดการกับข้อมูล

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์ที่จะพัฒนาสามารถใช้งานข้อมูลโดยภาษาธรรมชาติ (Natural Language)ได้ ใช่หรือไม่		+	+		-	
2) ซอฟต์แวร์ที่จะพัฒนามีการใช้งานข้อมูลที่เก็บแบบฐานข้อมูล (database) คือ มีโครงสร้างของฐานข้อมูล ประกอบด้วยแฟ้มข้อมูลมากกว่า 1 แฟ้มและมีการกำหนดความสัมพันธ์ของแฟ้มข้อมูลในฐานข้อมูล ใช่หรือไม่	+	+				
3) ซอฟต์แวร์ที่จะพัฒนามีการจัดการกับฐานข้อมูลที่เป็นแบบกระจาย(distributed database) ใช่หรือไม่ เช่น การจัดการกับสินค้าคงคลังของห้างสรรพสินค้าทุกๆสาขา		+				
4) ซอฟต์แวร์ที่จะพัฒนาสามารถทำทริกเกอร์(trigger) ได้ เช่น การแก้ไข บรรณาธิกร การเพิ่ม การลบ ข้อมูลในแฟ้มข้อมูลได้โดยการเขียนเป็นชุดคำสั่งที่มีเงื่อนไขไว้ เมื่อเงื่อนไขถูกตรวจสอบว่าเป็นจริง ชุดคำสั่งนั้นก็จะทำงานทันที ใช่หรือไม่		+				
5) ซอฟต์แวร์ที่จะพัฒนาสามารถแก้ไขหรือเปลี่ยนแปลงโครงสร้าง (restructuring)ของฐานข้อมูลได้ ใช่หรือไม่ เช่น เพิ่มจำนวนฟิลด์ (field)ข้อมูล การเปลี่ยนแปลงความสัมพันธ์ของข้อมูล เป็นต้น		+				
6) ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลได้ ใช่หรือไม่		+				
7) ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลเพียงแฟ้มเดียว ใช่หรือไม่		+				

1.3.5 การดำเนินการส่วนต่อประสานกับผู้ใช้(User Interface Management Operation)

การปฏิบัติการส่วนต่อประสานกับผู้ใช้ หมายถึง วิธีหรือรูปแบบของซอฟต์แวร์ที่จะพัฒนาติดต่อกับผู้ใช้

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	มีโปรแกรมช่วยในการสร้าง(generators)รูปแบบการรับข้อมูลเข้า(input form) และการทำรายงาน (report) แบบง่าย เช่น มีwizardช่วย
ต่ำ	มีการใช้ตัวสร้างส่วนต่อประสานกับผู้ใช้(user interface)ที่ไม่ซับซ้อน เช่น ใช้ตัวสร้างGUI (graphic user interface builders)
ปานกลาง	มีการใช้งานส่วนต่อประสานกับผู้ใช้เป็นแบบวินโดว์ที่ไม่มีความซับซ้อนมาก เช่น มีการติดต่อแบบ windows95 เป็นต้น
สูง	มีการใช้งานส่วนต่อประสานกับผู้ใช้แบบวินโดว์ซึ่งสามารถใช้เสียงเป็นตัวรับและแสดงผล และยังใช้ติดต่อแบบมัลติมีเดียอย่างง่ายได้ด้วย
สูงมาก	มีการใช้ภาพ2 มิติ หรือ 3มิติที่มีความซับซ้อนไม่มาก มีการใช้ภาพที่เป็นพลวัต(dynamic graphic) และมีการใช้มัลติมีเดีย(multimedia)ในการติดต่อกับผู้ใช้
สูงที่สุด	มีการใช้มัลติมีเดียที่ซับซ้อน เช่น วีดิโอคอนเฟอร์เร็น (video conference) และมีการใช้ภาพเสมือนจริง(virtual reality) ในการติดต่อกับผู้ใช้

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท การดำเนินการส่วนต่อประสานกับผู้ใช้

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้ภาพเสมือนจริง(virtual reality) หรือ วีดิโอคอนเฟอร์เร็น(video conference), ใช่หรือไม่		+ ↑	↓			
2) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้รูปภาพ2มิติ หรือ 3มิติ ที่เคลื่อนไหวได้ ใช่หรือไม่	↑	+ ↓				
3) ซอฟต์แวร์ที่จะพัฒนาสามารถใช้เสียงในการรับหรือแสดงผลได้ ใช่หรือไม่		+ ↑	↓			
4) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้ โดยมีลักษณะการใช้ใกล้เคียงกับการใช้วินโดว์ ใช่หรือไม่ เช่น มีไอคอน(Icon) มีทูบาร์(toolbars) เป็นต้น		+ ↑	↓			
5) ส่วนต่อประสานกับผู้ใช้แบบรูปภาพแบบง่ายและมีเครื่องมือช่วยสร้างที่เป็นแบบvisual programming ใช่หรือไม่	+ ↑	↓				

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท การดำเนินการส่วนต่อประสานกับผู้ใช้(ต่อ)

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
6) ท่านมีเครื่องมือที่ใช้สร้าง(generators)ส่วนที่ติดต่อประสานกับผู้ใช้แบบไม่มีความซับซ้อน ใช่หรือไม่ เช่น ใช้wizardสร้าง เป็นต้น		+			-	

1.4 ความต้องการที่จะนำผลิตภัณฑ์กลับมาใช้ใหม่ (Required Reusability : RUSE)

คำจำกัดความ

ตัวขับค่าใช้จ่ายนี้จะพิจารณาความพยายามสร้างซอฟต์แวร์ให้มีลักษณะที่เป็นองค์ประกอบ(component) เพื่อสามารถนำกลับไปใช้(Reuse) กับซอฟต์แวร์อื่นๆในอนาคต ซึ่งการที่จะนำบางส่วนของโปรแกรมกลับมาใช้ในลักษณะใดๆ ของซอฟต์แวร์ที่จะพัฒนานั้นก็ขึ้นอยู่กับ การออกแบบซอฟต์แวร์ให้มีลักษณะเป็นฟังก์ชันหรือเป็นโมดูลที่เล็กที่สุดที่มีฟังก์ชันเดียว

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำ	ไม่มีการออกแบบสำหรับการนำโปรแกรมกลับมาใช้ใหม่ในอนาคต
ปานกลาง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโครงการ (Project)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของ Microsoft Word6 มาใช้กับ Microsoft Word7 เป็นต้น
สูง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโปรแกรม (Program)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำบางส่วนของโปรแกรมของMicrosoft Excel มาใช้กับ Microsoft Word เป็นต้น
สูงมาก	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมที่ไม่ได้อยู่ในระบบ(Product Line)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของระบบซื้อ-ขาย มาใช้กับระบบธนาคาร เป็นต้น
สูงที่สุด	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมไปใช้ได้ในหลายๆระบบ เช่น นำส่วนของโปรแกรมของระบบบัญชี ระบบซื้อ-ขาย และระบบงานบุคคล มาใช้กับระบบธนาคาร เป็นต้น

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท RUSE

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมกลับมาใช้ใหม่ในอนาคต ใช่หรือไม่	↓	+ ↓				
2) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบให้มีลักษณะเป็นฟังก์ชันหรือเป็นโมดูลที่เล็กที่สุดที่มีฟังก์ชันเดียว(Generic Design) เพื่อให้ซอฟต์แวร์ใดๆก็ได้สามารถนำบางส่วนกลับไปใช้ใหม่ได้ ใช่หรือไม่	↓	+ ↓				
3) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมไปใช้กับระบบที่มีความใกล้เคียงกับระบบที่จะพัฒนาเท่านั้น ใช่หรือไม่	+ ↓	↓				
4) ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโปรแกรม(Program)ที่มีความใกล้เคียงหรือเหมือนกับโปรแกรมที่จะพัฒนา ใช่หรือไม่ เช่น นำบางส่วนของโปรแกรมของ Microsoft Excel มาใช้กับ Microsoft Word เป็นต้น	↓	+	↓			
5)ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโครงการ(Project)ที่มีลักษณะต่างไปจากโครงการที่จะพัฒนาใช่หรือไม่ เช่น นำส่วนของโปรแกรมของ Microsoft Word มาใช้กับ Microsoft Access	↓	+	↓			

1.5 ความต้องการเอกสารที่ตรงกับวงจรชีวิต(Documentation match to life-cycle needs: DOCU)

คำจำกัดความ

การกำหนดระดับของตัวขับค่าใช้จ่ายประเภทนี้จะอยู่ในรูปของการจัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์ได้ครบทุกขั้นตอนทั้งโครงการ ซึ่งขั้นตอนหรือรูปแบบของการจัดทำเอกสารก็ขึ้นอยู่กับผู้พัฒนาว่าใช้วงจรชีวิต(life-cycle)แบบใดในการพัฒนาซอฟต์แวร์ เช่น วอลเตอร์ฟอร์โมเดล(Waterfall Model) อินครีเมนต์ดิวอลป์โมเดล(Increment Development Model) สไปรรอลโมเดล(Spiral Model) โปรโตไทป์ปิงโมเดล(Prototyping Model) เป็นต้น ซึ่งการกำหนดระดับขึ้นกับเอกสารที่จัดทำครบทุกขั้นตอนและเพียงพอกับความต้องการใช้งานเพียงใด

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	ไม่มีการจัดทำเอกสารการพัฒนาซอฟต์แวร์
ต่ำ	เอกสารจัดทำไม่ครอบคลุมในหลายๆขั้นของวงจรชีวิต
ปานกลาง	เอกสารจัดทำได้ครบตามข้อกำหนดขั้นต่ำของวงจรชีวิต
สูง	เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่ใช้ภายในองค์กรแต่ไม่ได้เป็นข้อกำหนดไว้ในวงจรชีวิต เพื่อทำให้เกิดความสมบูรณ์มากขึ้น
สูงมาก	เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่มีความละเอียดเพื่อให้เกิดความสมบูรณ์มากที่สุด

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท DOCU

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ท่านต้องการใช้มีการจัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์ ใช่หรือไม่	+ ↳	↴				
2) ท่านต้องการให้มีการจัดทำเอกสารครบตามที่วงจรชีวิตได้ระบุไว้ ใช่หรือไม่	+ ↳	↴				
3) ท่านต้องการให้มีการจัดทำเอกสารที่นอกเหนือจากวงจรชีวิตระบุเพื่อใช้ในองค์กรด้วย ใช่หรือไม่ เช่นเอกสารที่ระบุถึงความสำคัญของบุคคลในองก์กรที่มีต่อซอฟต์แวร์ที่จะพัฒนา หรือข้อตกลงที่ใช้เฉพาะในองค์กร เป็นต้น		+ ↳	↴			
4) ท่านต้องการให้มีการจัดทำเอกสารที่นอกเหนือจากข้อกำหนดขั้นต่ำของวงจรชีวิต และมีความละเอียดในทุกขั้นตอน ใช่หรือไม่		+ ↳↴				

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2)ปัจจัยกลุ่มแพลทฟอร์ม

ปัจจัยที่เกี่ยวข้องกับฮาร์ดแวร์ ซอฟต์แวร์ที่สนับสนุนการทำงานของซอฟต์แวร์ที่จะพัฒนา ซึ่งแบ่งออกเป็น 3 ประเภทคือ เวลาที่ใช้ในการประมวลผล ขนาดของหน่วยความจำ และการเปลี่ยนแพลทฟอร์ม

2.1.การจำกัดเวลาที่ใช้ในการประมวลผล (Execute Time Constraint: TIME)

คำจำกัดความ

เนื่องจากเวลาที่เข้ากระทำการ(Execution Time)ขึ้นอยู่กับซอฟต์แวร์ระบบ(Software System) การกำหนดระดับของตัวขับค่าใช้จ่ายประเภทนี้จึงคิดเป็นเปอร์เซ็นต์ของเวลาที่คาดว่าซอฟต์แวร์จะเข้ากระทำการจากเวลาทั้งหมดที่ระบบมีให้กับซอฟต์แวร์

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ปานกลาง	เวลาที่ซอฟต์แวร์เข้ากระทำการน้อยกว่าหรือเท่ากับ 50% ของเวลาที่ซีพียูมีให้
สูง	เวลาที่ซอฟต์แวร์เข้ากระทำการมากกว่า 50% ของเวลาที่ซีพียูมีให้ แต่น้อยกว่าหรือเท่ากับ 70% ของเวลาที่ซีพียูมีให้
สูงมาก	เวลาที่ซอฟต์แวร์เข้ากระทำการมากกว่า 70% ของเวลาที่ซีพียูมีให้แต่น้อยกว่าหรือเท่ากับ 85% ของเวลาที่ซีพียูมีให้
สูงที่สุด	เวลาที่ซอฟต์แวร์เข้ากระทำการมากกว่า 85% ของเวลาที่ซีพียูมีให้ น้อยกว่าหรือเท่ากับ 95% ของเวลาที่ซีพียูมีให้

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท TIME

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
เวลาที่ท่านคาดว่าซอฟต์แวร์จะใช้ในการกระทำการ(Execution Time) ☐ในกรณีที่ท่านทราบ (1) เวลาที่ระบบให้ซอฟต์แวร์เข้ากระทำการ.....ชั่วโมงต่อวัน (2) เวลาที่ซอฟต์แวร์เข้าไปใช้งานในซีพียูโดยประมาณ.....ชั่วโมงต่อวัน		+				

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท TIME(ต่อ)

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
☐ในกรณีที่ท่านไม่ทราบ (1) ซอฟต์แวร์ที่พัฒนาเป็นซอฟต์แวร์ประเภทใด ( ) การจัดการฐานข้อมูล(Database Management) เช่น MS Access, CA-Clipper, FoxPro, Oracle, Paradox Alpha Four เป็นต้น ( ) การสื่อสาร(Communication) เช่น Comm Work, Crosstalk, SmartCom, Terminal Plus เป็นต้น ( ) การประมวลคำ (Word Processing) เช่น AmiPro3.01, MS Word, WordPerfect, WordStar เป็นต้น ( ) Spreadsheet เช่น CA-SuperCale, Lotus Improv, MS Excel, Quattro Pro เป็นต้น ( ) รูปภาพ(Graphic) เช่น CorelDRAW, FreeHand, Illustrator, DesignWorks เป็นต้น ( ) ซอฟต์แวร์ระบบ(System Software) เช่น OS/2, Win95/98/NT, UNIX, Linux เป็นต้น ( ) Integrated Software เช่น MS Work, Lotus Work, ClarisWork, Framework XE, Legato เป็นต้น ( ) แอปพลิเคชั่น(Applications) เช่น ระบบบัญชี ระบบบุคลากร เป็นต้น ( ) ระบบอื่น ระบุ(.....) (2) ระบบของท่านสามารถทำงานแบบ ( ) ผู้ใช้สามารถใช้งานได้พร้อมกันหลายคน ( ) ผู้ใช้สามารถใช้งานได้ทีละคน						

2.2.การจำกัดหน่วยความจำหลัก (Main Storage Constraint: STOR)

<u>คำจำกัดความ</u>

การกำหนดระดับให้กับตัวขับค่าใช้จ่ายประเภทนี้จะพิจารณาจากการที่ซอฟต์แวร์ใช้พื้นที่ในหน่วยความจำหลักมากน้อยเพียงใด ซึ่งถ้าซอฟต์แวร์ใช้หน่วยความจำมากซอฟต์แวร์ก็จะต้องมีการจัดการหน่วยความจำที่ดี การกำหนดระดับจะพิจารณาโดยคิดเป็นเปอร์เซ็นต์ของขนาดหน่วยความจำหลักที่คาดว่าจะใช้ จากขนาดของหน่วยความจำหลักที่ระบบมีให้

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ปานกลาง	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักน้อยกว่าหรือเท่ากับ 50% ของหน่วยความจำหลักที่สามารถใช้ได้
สูง	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักมากกว่า 50% ของหน่วยความจำหลักที่สามารถใช้ได้ แต่น้อยกว่าหรือเท่ากับ 70% ของหน่วยความจำหลักที่สามารถใช้ได้
สูงมาก	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักมากกว่า 70% ของหน่วยความจำหลักที่สามารถใช้ได้ แต่น้อยกว่าหรือเท่ากับ 85% ของหน่วยความจำหลักที่สามารถใช้ได้
สูงที่สุด	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักมากกว่า 85% ของหน่วยความจำหลักที่สามารถใช้ได้ แต่น้อยกว่าหรือเท่ากับ 95% ของหน่วยความจำหลักที่สามารถใช้ได้

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท STOR

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
ขนาดของหน่วยความจำหลักที่ท่านคาดว่าซอฟต์แวร์จะใช้ ☐ในกรณีที่ท่านทราบ (1) ขนาดของหน่วยความจำที่ระบบมีให้กับซอฟต์แวร์.....KB (2) ขนาดของหน่วยความจำที่คาดว่าซอฟต์แวร์ต้องการใช้.....KB ☐ในกรณีที่ท่านไม่ทราบ (1)ขนาดของหน่วยความจำหลักที่ซอฟต์แวร์จองไว้ใช้งาน.....KB (2) ขนาดของหน่วยความจำที่คาดว่าจะใช้เก็บข้อมูลที่ใช้งานกับซอฟต์แวร์.....KB (3) ขนาดของหน่วยความจำที่ซอฟต์แวร์ใช้ร่วมกับซอฟต์แวร์อื่นๆ.....KB (4) ขนาดของหน่วยความจำของเครื่องคอมพิวเตอร์ที่แนะนำ.....KB	↓	+ ↑				

2.3.การเปลี่ยนแปลงได้ง่ายของแพลทฟอร์ม (Platform Volatility: PVOL)

<u>คำจำกัดความ</u>

การเปลี่ยนแปลงได้ง่ายของแพลทฟอร์มหมายถึง ซอฟต์แวร์ที่จะพัฒนาต้องถูกออกแบบหรือพัฒนาให้มีความคล่องตัวสูงในการเปลี่ยนหรือปรับปรุงให้เข้ากับแพลทฟอร์มใหม่ ซึ่งถ้าซอฟต์แวร์ที่จะพัฒนาต้องทำการเปลี่ยนบ่อยๆ ก็ทำให้เกิดความความยากในการพัฒนา หรือ ใช้เวลามากในการพัฒนาเพื่อให้ซอฟต์แวร์สามารถทำงานบนหลายๆแพลทฟอร์มได้ เป็นต้น แพลทฟอร์มในที่นี้จะมีความหมายรวมทั้งฮาร์ดแวร์และซอฟต์แวร์(OS,DBMS etc) ตัวอย่างเช่น ถ้าซอฟต์แวร์ที่จะพัฒนาเป็นระบบปฏิบัติการ แพลทฟอร์มคืออุปกรณ์คอมพิวเตอร์ ถ้าซอฟต์แวร์ที่จะพัฒนาเป็นระบบจัดการฐานข้อมูล(database management) แพลทฟอร์มคือฮาร์ดแวร์และระบบปฏิบัติการ ถ้าซอฟต์แวร์ที่จะพัฒนาคือระบบบัญชี แพลทฟอร์มคือระบบปฏิบัติการ ระบบจัดการฐานข้อมูล และฮาร์ดแวร์ ถ้าซอฟต์แวร์ที่จะพัฒนาคือตัวค้นหาข้อความบนเครือข่าย(network text browser) แพลทฟอร์มคือ ฮาร์ดแวร์ ระบบปฏิบัติการ และคลังเก็บข้อมูลแบบกระจาย (distribute information repositories) นอกจากนี้แพลทฟอร์มยังรวมไปถึงตัวแปลงภาษา(compiler)

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำ	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 12 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 1 เดือน
ปานกลาง	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 6 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 2 สัปดาห์
สูง	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 1 สัปดาห์
สูงมาก	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 สัปดาห์ หรือทำการปรับปรุงบางส่วนทุกๆ 2 วัน

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท PVOL

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ท่านมีนโยบายที่จะทำการเปลี่ยนแพลทฟอร์มของซอฟต์แวร์ที่จะพัฒนาเป็นประจำในช่วงระยะเวลาใดเวลาหนึ่ง ใช่หรือไม่ เช่น ใช้เวลาในการเปลี่ยนเวอร์ชั่นของแพลทฟอร์มทุกๆ 6 เดือน หรือ ใช้เวลาในการเปลี่ยนชนิดของแพลทฟอร์มทุกๆ 12 เดือน เป็นต้น		+ ↑	↓			
2) โดยเฉลี่ยจะใช้งานแพลทฟอร์มเป็นเวลาประมาณ.....เดือนก่อนที่จะทำการเปลี่ยนแปลงในแต่ละครั้ง	↑	+ ↓				

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท PVOL(ต่อ)

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
3) ท่านจะต้องทำการปรับปรุงแพลทฟอร์มเพียงบางส่วนเป็นประจำในช่วงระยะเวลาใดเวลาหนึ่ง เพื่อให้ซอฟต์แวร์สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น หรือสามารถรองรับการใช้งานได้ ใช่หรือไม่ เช่น การขยายความสามารถในการใช้งานของผู้ให้บริการทางอินเตอร์เน็ต(Internet Service Provider) โดยการจัดการกับพื้นที่ในหน่วยความจำสำรองทุกสัปดาห์ เป็นต้น		+				
4) โดยเฉลี่ยใช้งานแพลทฟอร์มเป็นเวลาประมาณ..... สัปดาห์ก่อนที่จะทำการเปลี่ยนแปลงในแต่ละครั้ง		+				

3)ปัจจัยกลุ่มบุคลากร

ปัจจัยที่เกี่ยวข้องกับบุคลากรหรือทีมที่จะพัฒนาซอฟต์แวร์ซึ่งแบ่งออกเป็น 6 ประเภท คือ ความสามารถในการวิเคราะห์ ความสามารถของโปรแกรมเมอร์ ประสบการณ์การใช้แอปพลิเคชั่น ประสบการณ์การใช้แพลทฟอร์ม ประสบการณ์การใช้เครื่องมือ และความต่อเนื่องของบุคลากรที่พัฒนาซอฟต์แวร์

3.1.ความสามารถในการวิเคราะห์ (Analyst Capability: ACAP)

คำจำกัดความ

ตัวขับค่าใช้จ่ายประเภทนี้จะพิจารณาจากความสามารถของบุคลากรในการวิเคราะห์ระบบงานตามที่ร้องขอ (request) และการออกแบบซอฟต์แวร์อย่างรายละเอียด(Detailed Design) เช่น การออกแบบเกี่ยวกับข้อมูล (Data Design) การออกแบบเกี่ยวกับโครงสร้าง(Architectural Design) การออกแบบเกี่ยวกับกระบวนการ (Procedural Design) การออกแบบเกี่ยวกับการเชื่อมต่อ (Interface Design) เป็นต้น ปัจจัยหลักที่ใช้ในการแบ่งระดับความสามารถคือ ความสามารถ(ability)ในการวิเคราะห์และออกแบบ ประสิทธิภาพ(efficiency) และความละเอียด(thoroughness)ในการวิเคราะห์และออกแบบ และความสามารถในการสื่อสารกับผู้อื่น ในการกำหนดระดับจะไม่นำประสบการณ์ในการวิเคราะห์มาพิจารณาด้วยเนื่องจากได้แยกออกเป็นอีกหนึ่งตัวขับค่าใช้จ่าย การกำหนดระดับของความสามารถในการวิเคราะห์จะพิจารณาความสามารถโดยเฉลี่ยของนักวิเคราะห์ระบบในทีมว่ามีความสามารถอยู่ในตำแหน่งเปอร์เซ็นต์ไทล์ที่เท่าใด

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25
ต่ำ	เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45
ปานกลาง	เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65
สูง	เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85
สูงมาก	เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท ACAP

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1)จำนวนนักวิเคราะห์ระบบในทีมที่พัฒนาซอฟต์แวร์.....คน		+				
2)ท่านคิดว่านักวิเคราะห์ระบบแต่ละท่านในทีมมีความสามารถอยู่ในระดับใดบ้าง (มีความสามารถมากที่สุด = 5, มีความสามารถมาก = 4, มีความสามารถปานกลาง = 3, มีความสามารถน้อย = 2, มีความสามารถน้อยที่สุด = 1) นักวิเคราะห์ระบบคนที่ 1 มีความสามารถอยู่ในระดับ..... นักวิเคราะห์ระบบคนที่ 2 มีความสามารถอยู่ในระดับ..... นักวิเคราะห์ระบบคนที่ 3 มีความสามารถอยู่ในระดับ..... ฯลฯ		+				

3.2.ความสามารถของโปรแกรมเมอร์ (Programmer Capability: PCAP)

<u>คำจำกัดความ</u>

แม้ว่ามีการให้ความสำคัญกับการวิเคราะห์ระบบอย่างมาก แต่เครื่องมือในปัจจุบันก็ได้พัฒนามีจนมีความซับซ้อนมากขึ้น ดังนั้นการใช้งานเครื่องมือเหล่านี้ก็ขึ้นกับความสามารถของโปรแกรมเมอร์ การจะกำหนดระดับให้นั้นควรจะพิจารณาบนพื้นฐานของความสามารถโดยรวมของโปรแกรมเมอร์ในทีมที่พัฒนามากกว่าที่จะพิจารณาโปรแกรมเมอร์เพียงคนเดียว ปัจจัยร่วมที่มีส่วนช่วยในการกำหนดระดับคือ ความสามารถในการพัฒนาซอฟต์แวร์ ประสิทธิภาพของซอฟต์แวร์ ความสมบูรณ์ของซอฟต์แวร์ที่พัฒนา และความสามารถในการติดต่อสื่อสารกับผู้อื่น ตัวขับค่าใช้จ่ายประเภทนี้จะไม่นำประสบการณ์ในการเขียนโปรแกรมมาพิจารณาด้วย เนื่องจากได้แยกออกเป็นอีกหนึ่งตัวขับค่าใช้จ่าย การกำหนดระดับจะพิจารณาจากความสามารถเฉลี่ยในการเขียนโปรแกรมของทีมโปรแกรมเมอร์ว่าอยู่ในตำแหน่งของเปอร์เซ็นต์ไทล์ที่เท่าใด

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25
ต่ำ	เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45
ปานกลาง	เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65
สูง	เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85
สูงมาก	เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท PCAP

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) จำนวนโปรแกรมเมอร์ในทีมที่พัฒนาซอฟต์แวร์.....คน		+			-	
2) ท่านคิดว่าโปรแกรมเมอร์แต่ละท่านในทีมมีความสามารถอยู่ในระดับใด ใน5 ระดับนี้ (มีความสามารถมากที่สุด = 5, มีความสามารถมาก = 4, มีความสามารถปานกลาง = 3,มีความสามารถน้อย = 2, มีความสามารถน้อยที่สุด =1) โปรแกรมเมอร์คนที่ 1 มีความสามารถอยู่ในระดับ..... โปรแกรมเมอร์คนที่ 2 มีความสามารถอยู่ในระดับ..... โปรแกรมเมอร์คนที่ 3 มีความสามารถอยู่ในระดับ..... ฯลฯ	+					

3.3.ประสบการณ์การพัฒนาแอปพลิเคชั่น (Application Experience: AEXP)

<u>คำจำกัดความ</u>

การกำหนดระดับของตัวขับค่าใช้จ่ายประเภทนี้ขึ้นอยู่กับประสบการณ์ในการพัฒนาซอฟต์แวร์(แอปพลิเคชั่น)ชนิดเดียวกับซอฟต์แวร์ที่จะพัฒนา ซึ่งในตัวขับค่าใช้จ่ายนี้จะพิจารณาจากระยะเวลาเฉลี่ยที่ทีมพัฒนาเคยใช้พัฒนาซอฟต์แวร์ประเภทเดียวกับซอฟต์แวร์ที่จะพัฒนา

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท AEXP

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน	+ ↓	↓			-	
2) ประสบการณ์ในการพัฒนาซอฟต์แวร์ชนิดเดียวกับซอฟต์แวร์ที่จะพัฒนาของแต่ละท่าน บุคลากรคนที่ 1 มีประสบการณ์.....เดือน บุคลากรคนที่ 2 มีประสบการณ์.....เดือน บุคลากรคนที่ 3 มีประสบการณ์.....เดือน ฯลฯ	+ ↓	↓				

3.4ประสบการณ์การใช้แพลทฟอร์ม(Platform Experience: PEXP)

<u>คำจำกัดความ</u>

ตัวขับค่าใช้จ่ายนี้จะพิจารณาจากการทีมที่พัฒนามีประสบการณ์ในการใช้แพลทฟอร์มได้เป็นอย่างดี ซึ่งแพลทฟอร์มในที่นี้รวมทั้งฮาร์ดแวร์และซอฟต์แวร์ เช่น มีการรวมรูปภาพ(graphic)กับตัวเชื่อมประสาน(user interface) มีการใช้ฐานข้อมูล มีการใช้ระบบเครือข่าย(networking) เป็นต้น ซึ่งในตัวขับค่าใช้จ่ายนี้จะพิจารณาจากระยะเวลาที่ทีมพัฒนาเคยใช้แพลทฟอร์มประเภทเดียวกับแพลฟอร์มที่จะใช้ในซอฟต์แวร์ที่จะพัฒนา

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท PEXP

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน	↓	+ ↓				
3) ประสบการณ์ในการใช้แพลทฟอร์มชนิดเดียวกับแพลทฟอร์มที่จะใช้พัฒนาซอฟต์แวร์ของแต่ละท่าน บุคลากรคนที่ 1 มีประสบการณ์.....เดือน บุคลากรคนที่ 2 มีประสบการณ์.....เดือน บุคลากรคนที่ 3 มีประสบการณ์.....เดือน ฯลฯ	↓	+ ↓				

3.5.ประสบการณ์การใช้ภาษาโปรแกรมและเครื่องมือ(Language and Tool Experience: LTEX)

คำจำกัดความ

ตัวขับค่าใช้จ่ายนี้เป็นการวัดประสบการณ์ในการใช้ภาษาโปรแกรม(programming language) และใช้เครื่องมือในการพัฒนาซอฟต์แวร์ การพัฒนาซอฟต์แวร์ยังรวมถึงการใช้งานเครื่องมือพัฒนาซอฟต์แวร์ในการวิเคราะห์และออกแบบ การจัดการโครงแบบ(configuration management) การจัดทำเอกสารการพัฒนาซอฟต์แวร์ การจัดการไลบรารี่(library management) ของโปรแกรมภาษา การกำหนดรูปแบบการเขียนโปรแกรม(program style)ที่เป็นมาตราฐานเดียวกัน และการตรวจสอบความสอดคล้อง(consistency checking)ของโปรแกรม เป็นต้น นอกจากนี้การมีประสบการณ์ในการเขียนโปรแกรมในภาษาใดภาษาหนึ่งจะมีผลต่อระยะเวลาที่ใช้ในการพัฒนาด้วย

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท LTEX

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน		+				
2) ประสบการณ์ในการใช้ภาษาโปรแกรม(Language Programming) ชนิดเดียวกับที่จะใช้พัฒนาซอฟต์แวร์และการใช้เครื่องมือของโปรแกรมเมอร์แต่ละท่าน บุคลากรคนที่ 1 มีประสบการณ์.....เดือน.....เดือน บุคลากรคนที่ 2 มีประสบการณ์.....เดือน.....เดือน บุคลากรคนที่ 3 มีประสบการณ์.....เดือน.....เดือน ฯลฯ		+				

3.6.ความต่อเนื่องของบุคลากร(Personnel Continuity: PCON)

<u>คำจำกัดความ</u>

ความต่อเนื่องของบุคลากรหมายถึงการหมุนเวียนของบุคลากรที่อยู่ในทีมพัฒนาซอฟต์แวร์ตั้งแต่เริ่มพัฒนาจนกระทั่งเสร็จสมบูรณ์ โดยถ้ามีการหมุนเวียนเพียงเล็กน้อย ซอฟต์แวร์ก็น่าจะมีความต่อเนื่องในการพัฒนาสูง บุคลากรในที่นี้รวมถึง บุคคลในทุกตำแหน่งที่ร่วมอยู่ในทีมพัฒนา เช่น ผู้จัดการโครงการ ผู้ช่วยผู้จัดการโครงการ นักวิเคราะห์ระบบ โปรแกรมเมอร์ พนักงานทั่วไป เลขานุการ เป็นต้น

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 48%ต่อปี
ต่ำ	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 24%ต่อปี แต่น้อยกว่า48%ต่อปี
ปานกลาง	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 12%ต่อปี แต่น้อยกว่า24%ต่อปี
สูง	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 6%ต่อปี แต่น้อยกว่า12%ต่อปี
สูงมาก	บุคลากรมีการเปลี่ยนแปลงน้อยกว่า 6%ต่อปี

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท PCON

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) จำนวนบุคลากรที่พัฒนาเมื่อเริ่มการพัฒนาซอฟต์แวร์.....คน	↓	+ ↓				
2) ท่านมีนโยบายที่จะรับบุคลากรเข้าทีมพัฒนาเพิ่มอีก.....คน		+ ↓	↓			
3) ท่านมีนโยบายให้บุคลากรลาออกได้.....คน		↓	+ ↓			
4) บุคลากรที่รับเข้ามาใหม่จะทำงานแทนบุคลากรคนเก่าที่ ลาออกไป.....คน		+ ↓	↓			
5) ซอฟต์แวร์มีกำหนดเวลาของการพัฒนาประมาณ.....ปี	↓	+	↓			

4)ปัจจัยกลุ่มโครงการ

ปัจจัยที่เป็นภาพรวมของซอฟต์แวร์ที่ต้องการพัฒนา ซึ่งจะแบ่งออกเป็น 3 ประเภท คือการใช้เครื่องมือ สถานที่ที่จะใช้พัฒนาซอฟต์แวร์ และกำหนดการการพัฒนา

4.1.การใช้เครื่องมือซอฟต์แวร์ (Use of software tool: TOOL)

คำจำกัดความ

ความสามารถหรือประสิทธิภาพของเครื่องมือซอฟต์แวร์ได้พัฒนาจนเป็นที่ยอมรับว่ามีผลต่อการพัฒนาโครงการ เนื่องจากเครื่องมือในสมัยแรกๆสามารถทำได้เพียงการบรรณาธิกร(edit)และเขียน(code)โปรแกรม แต่ต่อมาเครื่องมือได้มีการพัฒนาจนสามารถนำมาใช้ในทุกขั้นของวงจรชีวิตของการพัฒนาได้ เช่นขั้นตอนการวิเคราะห์ (Analysis) ขั้นตอนออกแบบ(Design) และขั้นตอนการทำให้เกิดผล(Implementation)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	เครื่องมือสามารถสามารถเขียน(Code) บรรณาธิกร(Edit) และตรวจสอบความผิดพลาดในการเขียนโปรแกรมได้(Debug)
ต่ำ	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำฟอนแอนด์เคส (Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงแรกๆ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา (Requirements Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase) หรือ ช่วยในขั้นตอนการออกแบบ(Design Phase) หรือแบบเครื่องมือเคสที่ทำแบ็กเอนด์เคส (Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงหลัง เช่น ช่วยในขั้นตอนของการทำให้เกิดผล(Implementation Phase) ช่วยในขั้นตอนของการรวม(Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา(Maintenance Phase)
ปานกลาง	เครื่องมือที่ใช้พัฒนาซอฟต์แวร์สามารถใช้ได้ในทุกขั้นตอนของวงจรชีวิตในขั้นพื้นฐานได้ นอกจากนี้ยังสามารถทำการรวมบางขั้นตอนของวงจรชีวิตได้
สูง	เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นของวงจรชีวิตและยังสามารถทำการรวมทุกขั้นของวงจรชีวิตได้
สูงมาก	เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นตามรูปแบบของวงจรชีวิตนอกจากนี้ยังสามารถทำการรวมกัน(Integrated)โดยระเบียบวิธีการ(Method) กระบวนการ(Process) หรือการนำกลับมาใช้

กำหนดระดับคะแนนของชุดคำถามจากตัวชับค่าใช้จ่ายประเภท TOOL

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นภาษาโปรแกรม เช่น ภาษา C,Visual Basic, Pascal, Java เป็นต้น ใช่หรือไม่		+				
2) เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นเครื่องมือ เคส (Computer Aided Software Engineering :CASE) ใช่หรือไม่		+				
3) เครื่องมือที่ใช้นอกจากจะเป็นแบบเครื่องมือเคสที่ช่วยในการพัฒนาซอฟต์แวร์ได้ทุกขั้นตอนของวงจรชีวิต(SDLC)แล้วยังสามารถนำไปใช้รวม(Integrate)กับซอฟต์แวร์อื่นๆได้ ใช่หรือไม่		+				

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท TOOL(ต่อ)

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
4) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ช่วยในหลายขั้นตอนของวงจรชีวิต(SDLC) เช่น ช่วยในการเก็บรวบรวมข้อมูล(Required Phase) และนำข้อมูลที่ได้มาวิเคราะห์และออกแบบการพัฒนาซอฟต์แวร์ (Design Phase) หรือ ช่วยออกแบบซอฟต์แวร์(Design Phase)แล้วนำไปสร้างโปรแกรม(Code Generator) เป็นต้น ใช่หรือไม่	↳	+ ↴				
5) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำส่วนฟรอนเอนด์เคส (Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต (SDLC)ในช่วงแรกๆ ใช่หรือไม่ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา(Required Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase)		+ ↳	↴			
6) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำแบบฉีกเอนด์เคส (Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงหลัง ใช่หรือไม่ เช่น ช่วยในขั้นตอนของการทำให้เกิดผล(Implementation Phase) ช่วยในขั้นตอนของการรวม (Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา (Maintenance Phase)		+ ↳	↴			
7) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสแบบการรวม(Integrated CASE) ซึ่งสามารถทำการรวม(Integrated) แบบกระบวนการ (Process) วิธีการ(Method) หรือการใช้ใหม่(Reuse) ใช่หรือไม่		+ ↳	↴			

4.2.การพัฒนาในหลายสถานที่ (Multisite Development: SITE)

<u>คำจำกัดความ</u>

ตัวขับค่าใช้จ่ายประเภทนี้ได้ถูกเพิ่มเข้ามาในแบบจำลองโคโคโม2 เนื่องจากการวิจัยพบว่าสถานที่ที่ใช้พัฒนาและอุปกรณ์ที่ใช้ในการสื่อสารระหว่างบุคลากรในทีมพัฒนา มีผลต่อการพัฒนาซอฟต์แวร์ เช่น สถานที่ที่ห่างไกลกันหรืออุปกรณ์การสื่อสารที่ไม่ทันสมัยจะทำให้ไม่สามารถแก้ไขปัญหาได้ทันท่วงที เนื่องจากความล่าช้าในการติดต่อ

ตารางอธิบายความแตกต่างของแต่ละระดับเกี่ยวกับสถานที่

ระดับ	เงื่อนไขของสถานที่
ต่ำมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีทั้งในประเทศและต่างประเทศ
ต่ำ	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด และ มีบริษัทที่พัฒนาหลายบริษัท
ปานกลาง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด หรือ มีบริษัทที่พัฒนาหลายบริษัท หรือ ตั้งอยู่ในจังหวัดที่ไม่สำคัญเพียงจังหวัดเดียว เช่น หนองบัวลำภู ปัตตานี เป็นต้น
สูง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในกรุงเทพฯ หรือจังหวัดสำคัญๆ(กรุงเทพฯ เชียงใหม่ นครราชสีมา ภูเก็ต สงขลา เป็นต้น.)
สูงมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีหลายตึก หรือ เป็นหมู่ตึก
สูงที่สุด	สถานที่ที่พัฒนาซอฟต์แวร์จะอยู่ในบริเวณเดียวกัน เช่น อยู่ในตึกเดียวกัน อยู่ในชั้นเดียวกัน เป็นต้น

ตารางอธิบายความแตกต่างของแต่ละระดับการสื่อสารระหว่างการพัฒนา

ระดับ	เงื่อนไขของการสนับสนุนด้านการสื่อสาร
ต่ำมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์รวม หรือ จดหมาย
ต่ำ	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์ส่วนตัวหรือแฟกส่วนตัว
ปานกลาง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณแคบ เช่น ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น
สูง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น
สูงมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์
สูงที่สุด	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ มัลติมีเดียที่สามารถโต้ตอบได้

จุฬาลงกรณ์มหาวิทยาลัย

กำหนดระดับคะแนนของชุดคำถามจากตัวขับค่าใช้จ่ายประเภท SITE

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) ซอฟต์แวร์จะถูกพัฒนาภายในประเทศเท่านั้น ใช่หรือไม่	↓	+ ↑				
2) ซอฟต์แวร์จะถูกพัฒนาภายในจังหวัดเดียวกัน ใช่หรือไม่		+ ↓	↑			
3) ซอฟต์แวร์จะถูกพัฒนาภายในบริษัทเดียว ใช่หรือไม่	↓	+ ↑				
4) ซอฟต์แวร์จะถูกพัฒนาในกรุงเทพฯหรือจังหวัดสำคัญๆ ใช่หรือไม่	↓	+ ↑				
5) ซอฟต์แวร์จะถูกพัฒนาภายในอาคารเดียวกัน ใช่หรือไม่		+ ↓	↑			
6) ซอฟต์แวร์จะถูกพัฒนาภายในบริเวณเดียวกัน ห้องเดียวกัน หรือชั้นเดียวกัน ใช่หรือไม่		+ ↓	↑			
7) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ คือ มัลติมีเดียที่สามารถโต้ตอบได้ ใช่หรือไม่		+ ↓	↑			
8) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์ ใช่หรือไม่	↓	+	↑			
9) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น ใช่หรือไม่		+ ↓	↑			
10) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณแคบ ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น ใช่หรือไม่	↓	+ ↑				
11) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้โทรศัพท์ส่วนตัว หรือแฟกส่วนตัว ใช่หรือไม่	↓	+ ↑				

4.3.กำหนดเวลาของการพัฒนาที่ต้องการ (Required Development Schedule: SCED)

<u>คำจำกัดความ</u>

การกำหนดระดับจะขึ้นอยู่กับข้อจำกัดของทีมที่พัฒนา ดังนั้นจึงจะพิจารณาเป็นทีมที่พัฒนา โดยจะเทียบเป็นเปอร์เซ็นต์ของการขยายเวลาการทำงานออกไปหรือการเร่งเวลาการทำงานให้เร็วขึ้นเมื่อเทียบกับเวลาการทำงานปกติ ซึ่งการเร่งตารางการทำงานมีแนวโน้มว่าจะใช้ความพยายามสูงในเฟส(phase)ท้ายๆของการพัฒนา เนื่องจากอาจมีปัญหาหรือข้อสงสัยหลายอย่างที่ไม่ได้ถูกพิจารณาหรือแก้ไขในเฟสแรกๆ ดังนั้นจึงต้องมาทำในเฟสท้ายๆ และการขยายเวลาการทำงานออกไปมีแนวโน้มว่าจะใช้ความพยายามในเฟสแรกๆมาก เนื่องจากใช้ในการวางแผนการพัฒนาซอฟต์แวร์(Planning) การกำหนดคุณลักษณะ(Specifications)ของซอฟต์แวร์ที่จะพัฒนา และการวางแผนการตรวจสอบความถูกต้อง(Validation)ของซอฟต์แวร์

<u>ตารางอธิบายความแตกต่างของแต่ละระดับ</u>

ระดับ	เงื่อนไข
ต่ำมาก	เวลาในการพัฒนาน้อยกว่าหรือเท่ากับ 75% ของเวลาปกติ
ต่ำ	เวลาในการพัฒนามากกว่า 75%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ 85%
ปานกลาง	เวลาในการพัฒนามากกว่า 85%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ100%
สูง	เวลาในการพัฒนามากกว่า 100%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ130%
สูงมาก	เวลาในการพัฒนามากกว่า 130%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ160%

กำหนดระดับคะแนนของชุดคำถามจากตัวชี้วัดค่าใช้จ่ายประเภท SCED

คำถาม	ระดับคะแนน					เหตุผล
	5	4	3	2	1	
1) หลังจากที่ท่านได้เก็บรวบรวมความต้องการ(Requirement) ต่างๆ จากผู้ใช้แล้ว ท่านคาดว่าจะใช้เวลาในการพัฒนาซอฟต์แวร์ประมาณ .....เดือน	+ ↓	↓				
2) เวลาที่ท่านบวกเผื่อไว้(กรณีใดๆ)ประมาณ..... เดือน	+ ↓	↓				

แบบสอบถามรอบที่ 3

แบบสอบถามฉบับนี้เป็นเครื่องมือที่ใช้ในการเก็บข้อมูลรอบที่ 3 แต่มีวัตถุประสงค์ต่างจากแบบสอบถามรอบที่ 1 และ 2 คือในแบบสอบถามรอบนี้ผู้วิจัยต้องการให้ผู้เชี่ยวชาญพิจารณาคำตอบของคำถาม ที่ได้จากแบบสอบถามรอบที่ 1 และ 2 แล้วคัดเลือกเฉพาะ คำถามที่มีลักษณะปลายปิด ซึ่งมีทั้งหมด 6 ตัวขับค่าใช้จ่าย ในแบบสอบถามรอบนี้ ผู้วิจัยต้องการเก็บคำตอบของคำถามแต่ละข้อและผลสรุปของระดับที่เหมาะสม โดยผู้เชี่ยวชาญจะต้องพิจารณาคำตอบและระดับ(ดังตัวอย่าง) โดยอาศัยคำจำกัดความและตารางอธิบายความแตกต่างแต่ละระดับของแต่ละตัวขับค่าใช้จ่ายเป็นเกณฑ์ในการแบ่งระดับ

ขั้นตอนและการดำเนินการ

1. ให้ผู้เชี่ยวชาญอ่านคำจำกัดความและตารางอธิบายความแตกต่างแต่ละระดับของตัวขับค่าใช้จ่ายทั้ง 11 ตัว
2. จากนั้นให้ผู้เชี่ยวชาญ พิจารณาชุดคำตอบที่ผู้วิจัยสร้างขึ้น ซึ่งชุดคำตอบที่สร้างขึ้นนี้แบ่งได้เป็น 2 กลุ่ม คือ คำตอบที่มีลักษณะ ใช่ หรือ ไม่ใช่ และ คำตอบที่มีลักษณะแบ่งกลุ่ม ตามตารางอธิบายระดับของโคโคโม2 คือ ต่ำมาก ต่ำ ปานกลาง สูง สูงมาก และสูงที่สุด เมื่อผู้เชี่ยวชาญพิจารณาคำตอบแต่ละข้อคำถามแล้วเห็นด้วยกับระดับที่ผู้วิจัยจัดให้กับคำตอบ ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายลง(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายลง (✗)ในช่องความคิดเห็น
3. ถ้าท่านต้องการสร้างชุดข้อมูลใหม่ ให้เขียนลงในส่วนข้อเสนอแนะเพิ่มเติม

ตัวอย่าง

RELY	คำถาม ข้อที่1	คำถาม ข้อที่2	คำถาม ข้อที่3	คำถาม ข้อที่4	ระดับ	ความคิดเห็น
ชุดคำตอบที่1	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	✓
ชุดคำตอบที่2	ไม่ใช่	ใช่	ใช่	ใช่	สูง	✓
ชุดคำตอบที่3	ไม่ใช่	ไม่ใช่	ใช่	?	ปานกลาง	✓
ชุดคำตอบที่4	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	✓

จากตัวอย่างข้างต้น เป็นตัวขับค่าใช้จ่ายประเภทความเชื่อถือได้ของซอฟต์แวร์(RELY)

1. คำถามทั้ง 4 ข้อ ของ RELY อยู่ด้านหลัง
2. ชุดคำตอบที่ 1 จะมีคำตอบ 4 คำตอบ และถูกจัดอยู่ในกลุ่ม สูงมาก ในชุดคำตอบที่ 2 และ 3 ก็มีลักษณะใกล้เคียงกัน แต่ถูกจัดอยู่ในกลุ่ม สูง และ ปานกลางตามลำดับ ส่วนในชุดคำตอบที่ 2 คำตอบข้อที่ 4 คือ ? หมายถึง คำตอบของคำถามข้อที่ 4 จะเป็นถูกหรือผิดก็ได้ ก็จะถูกจัดอยู่ในกลุ่ม สูง เหมือนกัน
3. คำตอบที่เป็นไปได้ในแต่ละข้อมี 3 คำตอบ คือ

ใช่ ถ้าคำตอบเป็น ไม่ใช่ ถ้าคำตอบไม่ใช่ และ ? คำตอบอาจใช่หรือไม่ใช่ ก็ได้

1.ความน่าเชื่อถือของซอฟต์แวร์ที่ต้องการ (Required Software Reliability: RELY)

คำจำกัดความ

ความน่าเชื่อถือของซอฟต์แวร์หมายถึงซอฟต์แวร์ต้องสามารถทำงานได้ในช่วงเวลาที่กำหนดอย่างต่อเนื่องและถูกต้อง โดยไม่มีเหตุผิดปกติต่างๆ เช่น การคำนวณผิดพลาด การไม่ทำงานตามขั้นตอน การหยุดทำงานโดยที่ยังไม่จบการทำงานหรือไม่ถูกสั่งให้หยุด เป็นต้น โดยการพิจารณาความน่าเชื่อถือของผลิตภัณฑ์ซอฟต์แวร์ตามแนวทางของโคโคโม2 จะพิจารณาจากความเสียหายที่เกิดขึ้นหลังจากซอฟต์แวร์ไม่สามารถทำงานได้ เพื่อประเมินระดับของความพยายามที่จะใช้ในการพัฒนาซอฟต์แวร์

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำที่สุด	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความไม่สะดวกในการทำงานแต่สามารถดำเนินงานต่อไปได้
ต่ำ	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายเพียงเล็กน้อยในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม
ปานกลาง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายปานกลางในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม
สูง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความสูญเสียทางการเงินอย่างมาก
สูงมาก	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความเสี่ยงต่อชีวิตมนุษย์

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท RELY

ข้อที่	คำถาม
1	ซอฟต์แวร์ที่พัฒนาจะถูกนำไปใช้ในด้านที่อาจเกิดความเสี่ยงกับชีวิตมนุษย์ หรือมีผลกระทบต่อชีวิตมนุษย์ ใช่หรือไม่
2	ซอฟต์แวร์ที่จะพัฒนาเป็นซอฟต์แวร์ที่เกี่ยวข้องกับรายรับ-รายจ่ายขององค์กรเป็นอย่างมาก ซึ่งเมื่อซอฟต์แวร์ไม่สามารถทำงานได้จะกระทบต่อรายรับ-รายจ่ายขององค์กร ใช่หรือไม่
3	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติ ท่านจำเป็นต้องดำเนินการให้มีการแก้ไขทันที ใช่หรือไม่
4	ท่านสามารถดำเนินงานในลักษณะนั้นต่อได้ ถึงแม้ว่าซอฟต์แวร์จะไม่สามารถทำงานได้ก็ตาม ใช่หรือไม่ เช่นการใช้บุคลากรทำงานแทน เป็นต้น

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย RELY	คำถาม				ระดับ	ความคิดเห็น
	ข้อที่1	ข้อที่2	ข้อที่3	ข้อที่4		
ชุดคำตอบที่ 1	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 2	ใช่	ใช่	ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 3	ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 4	ใช่	ไม่ใช่	ไม่ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 5	ไม่ใช่	ใช่	ใช่	ใช่	สูง	
ชุดคำตอบที่ 6	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 7	ไม่ใช่	ใช่	ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 8	ไม่ใช่	ใช่	ไม่ใช่	ใช่	สูง	
ชุดคำตอบที่ 9	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่10	ไม่ใช่	ไม่ใช่	ใช่	ใช่	ปานกลาง	
ชุดคำตอบที่11	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่12	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ต่ำมาก	

2) ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์(Complexity: CPLX) การพิจารณาความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์จะพิจารณา 5 อย่างด้วยกัน

2.1 การดำเนินการควบคุม(Control Operation)

การดำเนินการควบคุมหมายถึง กระบวนการที่ใช้ควบคุมการทำงานของโปรแกรม หรือลักษณะการทำงานของโปรแกรมในซอฟต์แวร์ที่จะพัฒนา

ตารางแสดงความแตกต่างของแต่ละระดับของ Control Operation

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการ(programming operation)ของซอฟต์แวร์ ส่วนใหญ่จะเป็นการดำเนินการแบบตรงไปข้างหน้า (straight-line code) แต่ก็มีลักษณะการดำเนินการแบบเป็นเงื่อนไขแต่เป็นเงื่อนไขที่ไม่ซับซ้อน(non-nested structured) เช่น DO, CASE, IF-THEN-ELSE นอกจากนี้อาจมีการใช้โมดุลง่ายๆ เช่น การทำโปรซีเดอร์คอร์(procedure call) เป็นต้น

ตารางแสดงความแตกต่างของแต่ละระดับของ Control Operation(ต่อ)

ระดับ	เงื่อนไข
ต่ำ	การดำเนินการของซอฟต์แวร์จะมีโครงสร้างที่ซับซ้อน(nesting of structured programming) และโดยส่วนใหญ่จะเป็นแบบมีเงื่อนไข เช่น มีลูป WHILE ซ้อน IF-THEN-ELSE หรือ IF-THEN-ELSE ซ้อน IF-THEN-ELSE เป็นต้น
ปานกลาง	การดำเนินการของซอฟต์แวร์โดยส่วนใหญ่จะเป็นแบบมีเงื่อนไขที่ซับซ้อน จึงมีการใช้ตารางตัดสินใจ(decision table) หรือ มีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบ call back หรือแบบการส่งข้อความ(message passing) รวมทั้งมีการดำเนินการที่สนับสนุนการประมวลผลแบบกระจายบนเครือข่าย(distribute processing) เช่น ระบบclient/server
สูง	โครงสร้างของการดำเนินการมีความซับซ้อนมากซึ่งประกอบด้วยการดำเนินการที่เป็นแบบเงื่อนไขซ้อนเงื่อนไข หรือมีการใช้คิว(queue) และกองซ้อน(stack) เป็นตัวควบคุมลำดับการดำเนินการ หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน (homogeneous) คือ มียี่ห้อหรือเวอร์ชันเดียวกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด(soft real time)
สูงมาก	การดำเนินการจะมีลักษณะเป็นแบบการเวียนบังเกิด(recursive) หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความความต่างกัน(heterogeneous) คือ มียี่ห้อหรือเวอร์ชันต่างกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด (hard real time)
สูงที่สุด	การดำเนินการของซอฟต์แวร์จะต้องควบคุมการทำงานของฮาร์ดแวร์หรืออุปกรณ์โดยตรง (Microcode level control) หรือมีตารางการใช้ทรัพยากร(resource)ที่สามารถแก้ไขได้โดยอัตโนมัติตามความเหมาะสม หรือมีการประมวลผลแบบกระจาย(distribute processing) ที่ได้ผลลัพธ์แบบทันทีทันใด(hard real time)

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท Control Operation

ข้อที่	คำถาม
1	ซอฟต์แวร์ที่พัฒนาต้องเขียนโปรแกรมเป็นภาษาเครื่องเพื่อควบคุมการทำงานหรือใช้งานอุปกรณ์โดยตรง ใช่หรือไม่
2	ซอฟต์แวร์ที่จะพัฒนานี้สามารถประมวลแบบผลแบบกระจาย(distribute processing) เช่น การประมวลผลแบบ client/sever ใช่หรือไม่
3	ซอฟต์แวร์ที่พัฒนามีการประมวลผลแบบทันทีทันใด(real time) ใช่หรือไม่
4	ซอฟต์แวร์ที่พัฒนามีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน(homogeneous) คือ มียี่ห้อและเวอร์ชันเดียวกัน ใช่หรือไม่
5	ซอฟต์แวร์ที่จะพัฒนามีการเขียนโปรแกรมลักษณะเป็นแบบการเวียนบังเกิด(recursion) ใช่หรือไม่
6	ซอฟต์แวร์มีการจัดการกับเงื่อนไขจำนวนมากๆในแต่ละโปรซิเดอร์(Procedure) ใช่หรือไม่

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท Control Operation(ต่อ)

ข้อที่	คำถาม
7	ซอฟต์แวร์ที่จะพัฒนามีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบcall back หรือแบบการส่งข้อความ(message passing) ใช่หรือไม่
8	ซอฟต์แวร์ที่จะพัฒนามีโครงสร้างที่ซับซ้อนจึงต้องมีการใช้คิว(Queue) หรือกองซ้อน(Stack) เพื่อจัดลำดับการทำงานของโปรแกรมแบบที่เป็นเงื่อนไข ใช่หรือไม่
9	ซอฟต์แวร์ที่จะพัฒนามีส่วนของโปรแกรมเป็นลูปที่ซับซ้อนเงื่อนไข เช่น ลูป While ซ้อนกัน เป็นต้น ใช่หรือไม่

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย CPLX1	คำถาม									ระดับ	ความคิดเห็น
	ข้อที่ 1	ข้อที่ 2	ข้อที่ 3	ข้อที่ 4	ข้อที่ 5	ข้อที่ 6	ข้อที่ 7	ข้อที่ 8	ข้อที่ 9		
ชุดคำตอบที่ 1	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงที่สุด	
ชุดคำตอบที่ 2	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 3	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 4	ไม่ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	สูง	
ชุดคำตอบที่ 5	ไม่ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ใช่	สูง	
ชุดคำตอบที่ 6	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ใช่	ปานกลาง	
ชุดคำตอบที่ 7	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่?	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 8	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่ 9	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำมาก	

2.2 การดำเนินการคำนวณ(Computation Operation)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการคำนวณสามารถหาค่าจากสมการพีชคณิตหรือสมการเส้นตรง เช่น ทำการบวก การลบ การคูณ การหาร ตัวอย่างเช่น A=B+C*(D-E)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำ	การดำเนินการคำนวณสามารถหาค่าจากสมการที่มีความซากระดับกลาง(moderate-level expressions) คือ นอกจากทำการบวก การลบ การคูณ การหาร แล้วยังสามารถทำการยกกำลัง การหารากการคำนวณหาค่าตรีโกณมิติ ตัวอย่างเช่น D= SQRT(B^2 – 4*A*C)
ปานกลาง	การดำเนินการคำนวณมีการใช้รูทีน(routine) ที่เป็นมาตราฐานทางคณิตศาสตร์ และสถิติ เช่นการหาส่วนเบี่ยงเบนมาตราฐาน การหาค่า Z เป็นต้น นอกจากนี้ยังคำนวณหาค่าของเมตริกซ์(matrix)และเวกเตอร์(vector)ได้
สูง	การดำเนินการคำนวณสามารถวิเคราะห์เชิงตัวเลข(numerical analysis)ขั้นพื้นฐานได้ เช่น การหาคำตอบของสมการ โดยวิธีการของนิวตัน(Newton's Method) หรือ โดยวิธีการของ Guassian การแก้สมการเชิงอนุพันธ์ เป็นต้น
สูงมาก	การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างแน่นอน เช่น การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน (non-singular matrix equation) การหาค่าจากสมการเชิงอนุพันธ์ย่อย(partial differential equation) และการหารูปแบบของความสัมพันธ์จากข้อมูล 2 ชุด
สูงที่สุด	การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างไม่แน่นอน เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน(highly accurate analysis of noisy) การวิเคราะห์ข้อมูลแบบ stochastic

ตารางแสดงคำถามของตัวขับคำใช้จ่ายประเภท Computation Operation

ข้อที่	คำถาม
1	ซอฟต์แวร์ที่จะพัฒนาเกี่ยวข้องกับการวิเคราะห์โครงสร้างข้อมูลเชิงตัวเลขชนิดที่มีโครงสร้างที่ไม่แน่นอน ใช่หรือไม่ เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน การวิเคราะห์สภาพอากาศ เป็นต้น
2	ซอฟต์แวร์ที่จะพัฒนาจะต้องวิเคราะห์โครงสร้างเชิงตัวเลขได้จากชุดข้อมูล ใช่หรือไม่ เช่น การวิเคราะห์จุดสูงสุดจุดต่ำสุดของข้อมูล การหาสมการแทนการกระจายของชุดข้อมูล การหาค่าจากสมการเชิงอนุพันธ์ย่อย การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน เป็นต้น
3	ซอฟต์แวร์ที่จะพัฒนาสามารถวิเคราะห์เชิงตัวเลขขั้นพื้นฐานได้ ใช่หรือไม่ เช่น สามารถหาอนุพันธ์ การหาคำตอบของสมการ โดยวิธีการของนิวตัน(Newton's Method) การหาพื้นที่ได้กราฟจากการอินติเกรท(Integrate) เป็นต้น
4	ซอฟต์แวร์ที่จะพัฒนาต้องมีรูทีนที่เป็นมาตราฐานทางคณิตศาสตร์และสถิติ ใช่หรือไม่ เช่นการหาส่วนเบี่ยงเบนมาตราฐาน การหาค่า Z เป็นต้น
5	ซอฟต์แวร์ที่จะพัฒนามีการหาคำตอบจากสมการพีชคณิตหรือสมการเส้นตรง เช่น การทำบวก ลบ คูณ หาร ใช่หรือไม่

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวนับค่าใช้จ่าย CPLX2	คำถาม					ระดับ	ความคิดเห็น
	ข้อที่ 1	ข้อที่ 2	ข้อที่ 3	ข้อที่ 4	ข้อที่ 5		
ชุดคำตอบที่ 1	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงที่สุด	
ชุดคำตอบที่ 2	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 3	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 4	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 5	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำมาก	
ชุดคำตอบที่ 6	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ต่ำ	

2.3 การดำเนินการที่เกี่ยวกับอุปกรณ์(Device-dependent Operation)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์เป็นคำสั่งที่ใช้งานง่าย หรือเป็นภาษาระดับสูง เช่น การใช้คำสั่ง อ่าน(read) เขียน(write) ในภาษาพาสคาล(Pascal) ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล
ต่ำ	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์ สามารถใช้คำสั่งที่เป็นพื้นฐานที่ภาษามีการเตรียมไว้ให้ เช่น คำสั่งเขียนข้อมูลลงแฟ้มข้อมูล(put) หรือ อ่านข้อมูลในแฟ้มข้อมูล(get) ในภาษาซี(C) โดยที่ผู้พัฒนาไม่จำเป็นต้องมีความรู้เกี่ยวกับอุปกรณ์ที่ใช้
ปานกลาง	ผู้พัฒนาจะต้องมีความรู้อย่างเจาะจงเกี่ยวกับอุปกรณ์ที่เป็นตัวรับข้อมูลและตัวแสดงผล (input/output) โดยการใช้งานอุปกรณ์ต้องทำ 3 ขั้นตอนดังนี้คือ เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์นั้นและตรวจสอบความผิดพลาดในการทำงานของอุปกรณ์
สูง	การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะกระทำในระดับกายภาพ (physical) เช่น การแปลงตำแหน่งของหน่วยความจำทางกายภาพ เพื่อช่วยในการค้นหาและการอ่าน และการกำหนดรูปแบบการซ้อนทับ(overlap)ในหน่วยความจำของอุปกรณ์
สูงมาก	การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะมีรูทีน(routines) สำหรับจัดการเมื่อเกิดการผิดพลาดในขณะส่งและรับข้อมูล มีการจัดการเกี่ยวกับการส่งข้อมูลในสายการสื่อสาร หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบเข้มงวด(performance-intensive embedded system) เช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
สูงที่สุด	มีการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed operations) หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบวิกฤต (performance-critical embedded systems) เช่น มีการบันทึกการทำงานของอุปกรณ์ทุกชนิดตลอดเวลา

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท Device-dependent Operation

ข้อที่	คำถาม
1	ซอฟต์แวร์สามารถใช้งานอุปกรณ์โดยการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed) ใช่หรือไม่
2	ซอฟต์แวร์ที่จะพัฒนามีระบบตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์ แบบเข้มงวด เช่นมีการบันทึกการทำงานของอุปกรณ์ทุกตัว หรือ แบบวิกฤตเช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน ใช่หรือไม่
3	ซอฟต์แวร์ที่จะพัฒนามีรูทีนรองรับการขัดจังหวะ(Interrupt routines) เมื่อเกิดเหตุผิดพลาดขณะที่ซอฟต์แวร์กำลังใช้งานอุปกรณ์ ใช่หรือไม่
4	ซอฟต์แวร์ที่จะพัฒนาจะต้องดำเนินการกับอุปกรณ์ที่เป็นตัวรับข้อมูลหรือตัวแสดงผลข้อมูล ในระดับกายภาพ เช่น การแปลงตำแหน่งเชิงกายภาพ(address) ซึ่งเป็นที่เก็บข้อมูลจริงๆในหน่วยความจำสำรองหรือหน่วยความจำหลัก ใช่หรือไม่
5	ซอฟต์แวร์ที่จะพัฒนาจะต้องสามารถจัดการกับอุปกรณ์ได้ เช่น เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์ ตรวจสอบความผิดพลาดของอุปกรณ์ เป็นต้น ใช่หรือไม่
6	ใช้ภาษาระดับสูงในการใช้งานอุปกรณ์ เช่น การใช้คำสั่ง อ่าน(read) เขียน(write) ในภาษาพาสคาล (Pascal) ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล ใช่หรือไม่

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย CPLX3	คำถาม						ระดับ	ความคิดเห็น
	ข้อที่ 1	ข้อที่ 2	ข้อที่ 3	ข้อที่ 4	ข้อที่ 5	ข้อที่ 6		
ชุดข้อมูลที่ 1	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงที่สุด	
ชุดข้อมูลที่ 2	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 3	ไม่ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 4	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูง	

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น(ต่อ)

ตัวขับค่าใช้จ่าย CPLX3	คำถาม						ระดับ	ความคิดเห็น
	ข้อที่ 1	ข้อที่ 2	ข้อที่ 3	ข้อที่ 4	ข้อที่ 5	ข้อที่ 6		
ชุดคำตอบที่ 5	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 6	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ต่ำมาก	
ชุดคำตอบที่ 7	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	

2.4 การดำเนินการเกี่ยวกับการจัดการกับข้อมูล(Data Management Operation)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลมีการใช้แถวลำดับ(array)ในหน่วยความจำหลักเพื่อเก็บข้อมูล และสามารถใช้สอบถาม(query)และแก้ไข(update)ได้
ต่ำ	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ แฟ้มข้อมูลสามารถใช้สอบถามและแก้ไขได้
ปานกลาง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลได้หลายแฟ้มข้อมูลโดยจะเป็นแฟ้มส่งข้อมูลออกเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ จะไม่มีการดำเนินการกับแฟ้มข้อมูลที่เกิดในระหว่างการทำงาน เช่น log file
สูง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถทำทริกเกอร์(trigger)โดยใช้ข้อมูลเพียงชุดเดียว สามารถทำการปฏิวัติโครงสร้างของข้อมูลที่ซับซ้อน(complex data restructuring) เช่น การเปลี่ยนคีย์ การเปลี่ยนความสัมพันธ์ระหว่างแฟ้มข้อมูล เป็นต้น
สูงมาก	การดำเนินการเกี่ยวกับการจัดการฐานข้อมูลสามารถทำกับข้อมูลที่เก็บไว้ในหลายสถานที่ได้ สามารถทำทริกเกอร์(trigger)กับข้อมูลจำนวนมากๆได้ สามารถค้นหาข้อมูลในฐานข้อมูลได้
สูงที่สุด	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลเป็นแบบพลวัต(dynamic relational) คือสามารถเปลี่ยนแปลงโครงสร้างบางอย่างโดยอัตโนมัติตามสภาพของข้อมูลที่เก็บได้ มีโครงสร้างเชิงวัตถุ(object structures) มีการจัดการข้อมูลด้วยภาษาธรรมชาติ(natural language data management)

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท Data Management Operation

ข้อที่	คำถาม
1	ซอฟต์แวร์ที่จะพัฒนาสามารถใช้งานข้อมูลโดยภาษาธรรมชาติ(Natural Language)ได้ ใช่หรือไม่
2	ซอฟต์แวร์ที่จะพัฒนามีการใช้งานข้อมูลที่เก็บแบบฐานข้อมูล(database) คือ มีโครงสร้างของฐานข้อมูล ประกอบด้วยแฟ้มข้อมูลมากกว่า 1 แฟ้มและมีการกำหนดความสัมพันธ์ของแฟ้มข้อมูลในฐานข้อมูล ใช่หรือไม่
3	ซอฟต์แวร์ที่จะพัฒนามีการจัดการกับฐานข้อมูลที่เป็นแบบกระจาย(distributed database) ใช่หรือไม่ เช่น การจัดการกับสินค้าคงคลังของห้างสรรพสินค้าทุกๆสาขา
4	ซอฟต์แวร์ที่จะพัฒนาสามารถทำทริกเกอร์(trigger) ได้ เช่น การแก้ไข บรรณาธิกร การเพิ่ม การลบข้อมูลในแฟ้มข้อมูลได้โดยการเขียนเป็นชุดคำสั่งที่มีเงื่อนไขไว้ เมื่อเงื่อนไขถูกตรวจสอบว่าเป็นจริง ชุดคำสั่งนั้นก็จะทำงานทันที ใช่หรือไม่
5	ซอฟต์แวร์ที่จะพัฒนาสามารถแก้ไขหรือเปลี่ยนแปลงโครงสร้าง(restructuring)ของฐานข้อมูลได้ ใช่หรือไม่ เช่น เพิ่มจำนวนฟิลด์(field)ข้อมูล การเปลี่ยนแปลงความสัมพันธ์ของข้อมูล เป็นต้น
6	ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลได้ ใช่หรือไม่
7	ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลเพียงแฟ้มเดียว ใช่หรือไม่

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย CPLX4	คำถาม							ระดับ	ความคิดเห็น
	ข้อที่ 1	ข้อที่ 2	ข้อที่ 3	ข้อที่ 4	ข้อที่ 5	ข้อที่ 6	ข้อที่ 7		
ชุดคำตอบที่ 1	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงที่สุด	
ชุดคำตอบที่ 2	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 3	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 4	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 5	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 6	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 7	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ใช่	ต่ำ	
ชุดคำตอบที่ 8	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำมาก	

2.5 การดำเนินการส่วนต่อประสานกับผู้ใช้(User Interface Management Operation)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	มีโปรแกรมช่วยในการสร้าง(generators)รูปแบบการรับข้อมูลเข้า(input form) และการทำรายงาน (report) แบบง่าย เช่น มีwizardช่วย
ต่ำ	มีการใช้ตัวสร้างส่วนต่อประสานกับผู้ใช้(user interface)ที่ไม่ซับซ้อน เช่น ใช้ตัวสร้าง GUI (graphic user interface builders)
ปานกลาง	มีการใช้งานส่วนต่อประสานกับผู้ใช้เป็นแบบวินโดว์ที่ไม่มีความซับซ้อนมาก เช่น มีการติดต่อแบบ windows95 เป็นต้น.
สูง	มีการใช้งานส่วนต่อประสานกับผู้ใช้แบบวินโดว์ซึ่งสามารถใช้เสียงเป็นตัวรับและแสดงผล และยังใช้ติดต่อแบบมัลติมีเดียอย่างง่ายได้ด้วย
สูงมาก	มีการใช้ภาพ2 มิติ หรือ 3มิติที่มีความซับซ้อนไม่มาก มีการใช้ภาพที่เป็นพลวัต (dynamic graphic) และมีการใช้มัลติมีเดีย(multimedia)ในการติดต่อกับผู้ใช้
สูงที่สุด	มีการใช้มัลติมีเดียที่ซับซ้อน เช่น วีดิโอคอนเฟอร์เร็น (video conference) และมีการใช้ภาพเสมือนจริง(virtual reality) ในการติดต่อกับผู้ใช้

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท User Interface Management Operation

ข้อที่	คำถาม
1	ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้ภาพเสมือนจริง(virtual reality) หรือ วีดิโอคอนเฟอร์เร็น(video conference) ใช่หรือไม่
2	ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้รูปภาพ2มิติ หรือ 3มิติ ที่เคลื่อนไหวได้ ใช่หรือไม่
3	ซอฟต์แวร์ที่จะพัฒนาสามารถใช้เสียงในการรับหรือแสดงผลได้ ใช่หรือไม่
4	ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้ โดยมีลักษณะการใช้ใกล้เคียงกับการใช้วินโดว์ ใช่หรือไม่ เช่น มีไอคอน(Icon) มีทูบาร์(toolbars) เป็นต้น
5	ส่วนต่อประสานกับผู้ใช้แบบรูปภาพแบบง่ายและมีเครื่องมือช่วยสร้าง ใช่หรือไม่
6	ท่านมีเครื่องมือที่ใช้สร้าง(generators)ส่วนที่ติดต่อประสานกับผู้ใช้แบบไม่มีความซับซ้อน ใช่หรือไม่ เช่น ใช้wizardสร้าง เป็นต้น

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย CPLX5	คำถาม						ระดับ	ความคิดเห็น
	ข้อที่ 1	ข้อที่ 2	ข้อที่ 3	ข้อที่ 4	ข้อที่ 5	ข้อที่ 6		
ชุดคำตอบที่ 1	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงที่สุด	
ชุดคำตอบที่ 2	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 3	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 4	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 5	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่ 6	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่ 7	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ต่ำมาก	

3. ความต้องการที่จะนำผลิตภัณฑ์กลับมาใช้ใหม่ (Required Reusability : RUSE)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำ	ไม่มีการออกแบบสำหรับการนำโปรแกรมกลับมาใช้ใหม่ในอนาคต
ปานกลาง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโครงการ (Project)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของ Microsoft Word6 มาใช้กับ Microsoft Word7 เป็นต้น
สูง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโปรแกรม (Program)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำบางส่วนของโปรแกรมของMicrosoft Excel มาใช้กับ Microsoft Word เป็นต้น
สูงมาก	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมที่ไม่ได้อยู่ในระบบ(Product Line)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของระบบซื้อ-ขาย มาใช้กับระบบธนาคาร เป็นต้น
สูงที่สุด	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมไปใช้ได้ในหลายๆระบบ เช่น นำส่วนของโปรแกรมของระบบบัญชี ระบบซื้อ-ขาย และระบบงานบุคคล มาใช้กับระบบธนาคาร เป็นต้น

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท RUSE

ข้อที่	คำถาม
1	ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมกลับมาใช้ใหม่ในอนาคต ใช่หรือไม่
2	ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบให้มีลักษณะเป็นฟังก์ชันหรือเป็นโมดุลที่เล็กที่สุดที่มีฟังก์ชันเดียว(Generic Design) เพื่อให้ซอฟต์แวร์ใดๆก็ได้สามารถนำบางส่วนกลับไปใช้ใหม่ได้ ใช่หรือไม่
3	ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมไปใช้กับระบบที่มีความใกล้เคียงกับระบบที่จะพัฒนาเท่านั้น ใช่หรือไม่
4	ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโปรแกรม(Program)ที่มีความใกล้เคียงหรือเหมือนกับโปรแกรมที่จะพัฒนา ใช่หรือไม่ เช่น นำบางส่วนของโปรแกรมของMicrosoft Excel มาใช้กับ Microsoft Word เป็นต้น
5	ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโครงการ(Project)ใหม่ที่มีความใกล้เคียงหรือเหมือนกับโครงการที่จะพัฒนา ใช่หรือไม่ เช่น นำส่วนของโปแกรมของ Microsoft Word6 มาใช้กับ Microsoft7

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย RUSE	คำถาม					ระดับ	ความคิดเห็น
	ข้อที่1	ข้อที่2	ข้อที่3	ข้อที่4	ข้อที่5		
ชุดคำตอบที่ 1	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	สูงที่สุด	
ชุดคำตอบที่ 2	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	สูงที่สุด	
ชุดคำตอบที่ 3	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 4	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 5	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 6	ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 7	ใช่	ไม่ใช่	ใช่	ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 8	ใช่	ไม่ใช่	ใช่	ใช่	ใช่	ปานกลาง	
ชุดคำตอบที่ 9	ใช่	ไม่ใช่	ใช่	ใช่	ใช่	ปานกลาง	
ชุดคำตอบที่ 10	ใช่	ไม่ใช่	ใช่	ใช่	ไม่ใช่	ต่ำ	

4. ความต้องการเอกสารที่ตรงกับวงจรชีวิต(Documentation match to life-cycle needs: DOCU)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	ไม่มีการจัดทำเอกสารการพัฒนาซอฟต์แวร์
ต่ำ	เอกสารจัดทำไม่ครอบคลุมในหลายๆขั้นของวงจรชีวิต
ปานกลาง	เอกสารจัดทำได้ครบตามข้อกำหนดขั้นต่ำของวงจรชีวิต
สูง	เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่ใช้ภายในองค์กรแต่ไม่ได้เป็นข้อกำหนดไว้ในวงจรชีวิต เพื่อทำให้เกิดความสมบูรณ์มากขึ้น
สูงมาก	เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่มีความละเอียดเพื่อให้เกิดความสมบูรณ์มากที่สุด

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท DOCU

ลำดับที่	คำถาม
1	ท่านต้องการใช้มีการจัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์ ใช่หรือไม่
2	ท่านต้องการให้มีการจัดทำเอกสารครบตามที่วงจรชีวิตได้ระบุไว้ ใช่หรือไม่
3	ท่านต้องการให้มีการจัดทำเอกสารที่นอกเหนือจากวงจรชีวิตระบุเพื่อใช้ในองค์กรด้วย ใช่หรือไม่ เช่น เอกสารที่ระบุถึงความสำคัญของบุคคลในองค์กรที่มีต่อซอฟต์แวร์ที่จะพัฒนา หรือข้อตกลงที่ใช้เฉพาะในองค์กร เป็นต้น
4	ท่านต้องการให้เอกสารในส่วนที่เพิ่มเติมจากข้อกำหนดของวงจรชีวิตมีความละเอียดในทุกขั้นตอน ใช่หรือไม่

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย DOCU	คำถาม				ระดับ	ความคิดเห็น
	ข้อที่1	ข้อที่2	ข้อที่3	ข้อที่4		
ชุดคำตอบที่ 1	ใช่	ใช่	ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 2	ใช่	ใช่	ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 3	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 4	ใช่	ไม่ใช่	ใช่	ใช่	ต่ำ	
ชุดคำตอบที่ 5	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำมาก	
ชุดคำตอบที่ 6	ไม่ใช่	ใช่	ใช่	ใช่	ต่ำมาก	

5. การใช้เครื่องมือซอฟต์แวร์ (Use of software tool: TOOL)

ตารางอธิบายความแตกต่างของแต่ละระดับ

ระดับ	เงื่อนไข
ต่ำมาก	เครื่องมือสามารถสามารถเขียน(Code) บรรณาธิกร(Edit) และตรวจสอบความผิดพลาดในการเขียนโปรแกรมได้(Debug)
ต่ำ	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำฟอนเอนด์เคส (Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงแรกๆ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา(Requirements Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase)หรือ ช่วยในขั้นตอนการออกแบบ(Design Phase) หรือแบบเครื่องมือเคสที่ทำแบบอีกเอนด์เคส(Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงหลัง เช่น ช่วยในขั้นตอนของการทำให้เกิดผล (Implementation Phase) ช่วยในขั้นตอนของการรวม(Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา(Maintenance Phase)
ปานกลาง	เครื่องมือที่ใช้พัฒนาซอฟต์แวร์สามารถใช้ได้ในทุกขั้นตอนของวงจรชีวิตในขั้นพื้นฐานได้ นอกจากนี้ยังสามารถทำการรวมบางขั้นตอนของวงจรชีวิตได้
สูง	เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นของวงจรชีวิตและยังสามารถทำการรวมทุกขั้นของวงจรชีวิตได้
สูงมาก	เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นตามรูปแบบของวงจรชีวิตนอกจากนี้ยังสามารถทำการรวมกัน(Integrated)โดยระเบียบวิธีการ(Method) กระบวนการ (Process) หรือการนำกลับมาใช้

ตารางแสดงคำถามของตัวชี้วัดค่าใช้จ่ายประเภท TOOL

ข้อที่	คำถาม
1	เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นภาษาโปรแกรม เช่น ภาษา C,Visual Basic, Pascal, Java เป็นต้น ใช่หรือไม่
2	เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นเครื่องมือเคส (Computer Aided Software Engineering : CASE) ใช่หรือไม่
3	เครื่องมือที่ใช้นอกจากจะเป็นแบบเครื่องมือเคสที่ช่วยในการพัฒนาซอฟต์แวร์ได้ทุกขั้นตอนของวงจรชีวิต(SDLC)แล้วยังสามารถนำไปใช้รวม(Integrate)กับซอฟต์แวร์อื่นๆได้ ใช่หรือไม่

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท TOOL

ข้อที่	คำถาม
4	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ช่วยในหลายขั้นตอนของวงจรชีวิต(SDLC) เช่น ช่วยในการเก็บรวบรวมข้อมูล(Required Phase)และนำข้อมูลที่ได้มาวิเคราะห์และออกแบบการพัฒนาซอฟต์แวร์ (Design Phase) หรือ ช่วยออกแบบซอฟต์แวร์(Design Phase)แล้วนำไปสร้างโปรแกรม(Code Generator) เป็นต้น ใช่หรือไม่
5	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำฟ้อนแอนด์เคส (Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงแรกๆ ใช่หรือไม่ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา(Required Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase)
6	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำแบบอีกแอนด์เคส (Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงหลัง ใช่หรือไม่ เช่น ช่วยในขั้นตอนของการทำให้เกิดผล(Implementation Phase) ช่วยในขั้นตอนของการรวม(Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา(Maintenance Phase)
7	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสแบบการรวม(Integrated CASE) ซึ่งสามารถทำการรวม (Integrated) แบบกระบวนการ(Process) วิธีการ(Method) หรือการใช้ใหม่(Reuse) ใช่หรือไม่

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายถูก(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายถูก (✗)ในช่องความคิดเห็น

ตัวขับค่าใช้จ่าย TOOL	คำถาม							ระดับ	ความคิดเห็น
	ข้อที่1	ข้อที่2	ข้อที่3	ข้อที่4	ข้อที่5	ข้อที่6	ข้อที่7		
ชุดคำตอบที่ 1	ไม่ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 2	ไม่ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 3	ไม่ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 4	ไม่ใช่	ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 5	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่ 6	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่ 7	ไม่ใช่	ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่ 8	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำมาก	
ชุดคำตอบที่ 9	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ต่ำมาก	

6. การพัฒนาในหลายสถานที่ (Multisite Development: SITE)

ตารางอธิบายความแตกต่างของแต่ละระดับเกี่ยวกับสถานที่

ระดับ	เงื่อนไขของสถานที่
ต่ำมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีทั้งในประเทศและต่างประเทศ
ต่ำ	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด และ มีบริษัทที่พัฒนาหลายบริษัท
ปานกลาง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด หรือ มีบริษัทที่พัฒนาหลายบริษัท หรือ ตั้งอยู่ในจังหวัดที่ไม่สำคัญเพียงจังหวัดเดียว เช่น หนองบัวลำภู ปัตตานี เป็นต้น
สูง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในกรุงเทพฯ หรือจังหวัดสำคัญๆ(กรุงเทพฯ เชียงใหม่ นครราชสีมา ภูเก็ต สงขลา ..)
สูงมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีหลายตึก หรือ เป็นหมู่ตึก
สูงที่สุด	สถานที่ที่พัฒนาซอฟต์แวร์จะอยู่ในบริเวณเดียวกัน เช่น อยู่ในตึกเดียวกัน อยู่ในชั้นเดียวกัน เป็นต้น

ตารางอธิบายความแตกต่างของแต่ละระดับการสื่อสารระหว่างการพัฒนา

ระดับ	เงื่อนไขของการสนับสนุนด้านการสื่อสาร
ต่ำมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์รวม หรือ จดหมาย
ต่ำ	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์ส่วนตัวหรือแฟกส่วนตัว
ปานกลาง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณแคบ เช่น ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น
สูง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น
สูงมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์
สูงที่สุด	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ มัลติมีเดียที่สามารถโต้ตอบได้

จุฬาลงกรณ์มหาวิทยาลัย

ตารางแสดงคำถามของตัวขับค่าใช้จ่ายประเภท SITE

ข้อที่	คำถาม
1	ซอฟต์แวร์จะถูกพัฒนาภายในประเทศท่านั้น ใช่หรือไม่
2	ซอฟต์แวร์จะถูกพัฒนาภายในจังหวัดเดียวกัน ใช่หรือไม่
3	ซอฟต์แวร์จะถูกพัฒนาภายในบริษัทเดียว ใช่หรือไม่
4	ซอฟต์แวร์จะถูกพัฒนาในกรุงเทพฯหรือจังหวัดสำคัญๆ ใช่หรือไม่
5	ซอฟต์แวร์จะถูกพัฒนาภายในอาคารเดียวกัน ใช่หรือไม่
6	ซอฟต์แวร์จะถูกพัฒนาภายในบริเวณเดียวกัน ห้องเดียวกัน หรือชั้นเดียวกัน ใช่หรือไม่
7	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ คือ มัลติมีเดียที่สามารถโต้ตอบได้ ใช่หรือไม่
8	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์ ใช่หรือไม่
9	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น ใช่หรือไม่
10	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณแคบ ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น ใช่หรือไม่
11	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้โทรศัพท์ส่วนตัว หรือแฟกส่วนตัว ใช่หรือไม่

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ถ้าท่านเห็นด้วย ขอให้ผู้เชี่ยวชาญทำเครื่องหมายอง(✓)ในช่องความคิดเห็น หรือ ถ้าท่านไม่เห็นด้วยให้ทำเครื่องหมายอง(✕)ในช่องความคิดเห็น

ตัวชับค่าใช้จ่าย SITE	คำถาม											ระดับ	ความคิดเห็น
	ข้อที่1	ข้อที่2	ข้อที่3	ข้อที่4	ข้อที่5	ข้อที่6	ข้อที่7	ข้อที่8	ข้อที่9	ข้อที่10	ข้อที่11		
ชุดคำตอบที่ 1	ใช่	ไม่ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงที่สุด	
ชุดคำตอบที่ 2	ใช่	ไม่ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงที่สุด	
ชุดคำตอบที่ 3	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	สูงที่สุด	
ชุดคำตอบที่ 4	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	สูงที่สุด	
ชุดคำตอบที่ 5	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ใช่	ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 6	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ใช่	สูงมาก	
ชุดคำตอบที่ 7	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูงมาก	
ชุดคำตอบที่ 8	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 9	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 10	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	สูง	
ชุดคำตอบที่ 11	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ใช่	ปานกลาง	
ชุดคำตอบที่ 12	ใช่	?	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ปานกลาง	
ชุดคำตอบที่ 13	ใช่	ใช่	ใช่	ใช่	ใช่	ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ปานกลาง	
ชุดคำตอบที่ 14	ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ไม่ใช่	ไม่ใช่	ต่ำ	
ชุดคำตอบที่ 15	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ไม่ใช่	ใช่	ต่ำมาก	

ข้อเสนอแนะเพิ่มเติม

ตัวขับค่าใช้จ่าย	คำถาม											ระดับ	ความคิดเห็น
	ข้อที่1	ข้อที่2	ข้อที่3	ข้อที่4	ข้อที่5	ข้อที่6	ข้อที่7	ข้อที่8	ข้อที่9	ข้อที่10	ข้อที่11		

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

ตารางแสดงข้อมูลที่ได้จากแบบสอบถาม

ตารางที่ ข.1 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 1

คำถาม		ความถี่ของแต่ระดับคะแนน					มัธยฐาน	ควอไทล์ที่ 3	ควอไทล์ที่ 1
		1	2	3	4	5			
RELY	1	0	1	2	8	2	4	4.00	4.00
	2	0	1	1	7	4	4	5.00	4.00
	3	0	0	1	9	3	4	4.00	4.00
	4	0	0	3	6	4	4	5.00	4.00
	5	1	3	4	3	2	3	5.00	3.00
	6	1	0	5	6	1	4	4.00	3.00
	7	0	1	4	6	2	4	4.00	3.00
DATA	1	0	1	2	6	4	4	5.00	4.00
	2	0	0	6	4	3	4	4.00	3.00
CPLX	1	1	0	4	3	5	4	4.00	3.00
	2	0	1	0	6	6	4	5.00	4.00
	3	0	0	1	5	7	5	5.00	4.00
	4	0	1	3	6	3	4	4.00	3.00
	5	0	2	4	5	2	4	4.00	3.00
	6	2	2	4	4	1	3	4.00	3.00
	7	0	2	3	6	2	4	4.00	3.00
	8	2	0	2	5	4	4	5.00	3.00
	9	2	1	3	5	2	4	4.00	3.00
	10	1	1	1	6	4	4	4.00	4.00
	11	0	1	3	6	3	4	4.00	3.00
	12	0	2	3	5	3	4	4.00	3.00
	13	0	0	4	5	4	4	5.00	3.00
	14	0	0	3	6	4	4	5.00	4.00

ตารางที่ ข.1 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 1(ต่อ)

คำถาม	ความถี่ของแต่ระดับคะแนน					มัธยฐาน	ควอไทล์ที่ 3	ควอไทล์ที่ 1
	1	2	3	4	5			
CPLX 15	1	0	2	5	5	4	5.00	4.00
16	0	1	2	6	4	4	5.00	4.00
17	0	3	0	4	6	4	5.00	4.00
18	0	1	3	7	2	4	4.00	3.00
19	0	1	3	6	3	4	4.00	3.00
20	1	1	6	4	1	4	4.00	3.00
21	1	0	5	3	3	4	4.00	3.00
22	0	0	1	3	9	5	5.00	4.00
23	0	2	1	5	5	4	5.00	4.00
24	1	0	4	7	1	4	4.00	3.00
25	0	1	2	4	6	4	5.00	4.00
26	1	1	2	3	6	4	5.00	3.00
27	3	1	2	5	2	4	4.00	2.00
28	0	1	2	5	5	4	5.00	4.00
29	1	1	3	5	3	4	4.00	3.00
30	1	1	1	6	4	4	5.00	4.00
31	0	1	4	5	3	4	4.00	3.00
32	1	1	3	5	3	4	4.00	3.00
33	0	1	2	3	7	5	5.00	4.00
RUSE 1	1	0	1	6	5	4	5.00	4.00
2	0	1	1	5	6	4	5.00	4.00
3	0	0	2	4	7	5	5.00	4.00
4	0	2	3	3	5	4	5.00	3.00
5	1	1	3	3	5	4	5.00	3.00
DOCU 1	0	1	0	4	8	5	5.00	4.00
2	0	0	2	4	7	5	5.00	4.00
3	0	0	4	6	3	4	4.00	3.00
4	0	0	2	8	3	4	4.00	4.00

ตารางที่ ข.1 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 1(ต่อ)

คำถาม	ความถี่ของแต่ระดับคะแนน					มัธยฐาน	ควอไทล์ที่ 3	ควอไทล์ที่ 1
	1	2	3	4	5			
TIME	0	0	4	6	3	4	4.00	3.00
STOR	0	0	3	5	5	4	5.00	4.00
PVOL 1	1	2	2	6	2	4	4.00	3.00
2	0	0	3	6	4	4	5.00	4.00
3	0	1	4	4	4	4	5.00	3.00
4	0	2	3	5	3	4	4.00	3.00
ACAP 1	0	1	3	6	3	4	4.00	3.00
2	0	0	2	7	4	4	5.00	4.00
PCAP 1	0	0	3	5	5	4	5.00	4.00
2	0	0	2	4	7	5	5.00	4.00
AEXP 1	0	0	3	3	7	5	5.00	4.00
2	0	0	1	4	8	5	5.00	4.00
PEXP 1	0	0	3	4	6	4	5.00	4.00
2	0	0	1	6	6	4	5.00	4.00
LTEX 1	0	0	4	5	4	4	5.00	3.00
2	0	0	3	5	5	4	5.00	4.00
3	0	1	1	5	6	4	5.00	4.00
PCON 1	0	0	2	6	5	4	4.00	4.00
2	1	1	4	4	3	4	4.00	3.00
3	0	0	6	4	3	4	4.00	3.00
4	1	0	5	5	2	4	4.00	3.00
5	0	0	4	6	3	4	5.00	3.00

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.1 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 1(ต่อ)

คำถาม	ความถี่ของแต่ระดับคะแนน					มัธยฐาน	ควอไทล์ที่ 3	ควอไทล์ที่1
	1	2	3	4	5			
TOOL 1	0	0	4	6	3	4	4.00	3.00
2	0	1	2	5	5	4	5.00	4.00
3	0	0	3	6	4	4	5.00	4.00
4	0	0	1	7	5	4	5.00	4.00
5	0	0	5	5	3	4	4.00	3.00
6	0	0	5	5	3	4	4.00	3.00
7	0	0	5	5	3	4	4.00	3.00
SITE 1	0	0	2	6	5	4	5.00	4.00
2	0	2	3	5	3	4	4.00	3.00
3	0	0	0	7	6	4	5.00	4.00
4	0	1	2	5	5	4	5.00	4.00
5	0	1	5	4	3	4	4.00	3.00
6	1	0	4	5	3	4	4.00	3.00
7	0	2	3	5	3	4	4.00	3.00
8	0	2	3	4	2	4	4.00	3.00
9	0	0	4	6	3	4	4.00	3.00
10	0	0	2	6	5	4	5.00	4.00
11	0	0	2	6	5	4	5.00	4.00
SCED 1	1	0	2	2	8	5	5.00	4.00
2	0	1	2	3	7	5	5.00	4.00

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.2 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 2

คำถาม	ความถี่ของแต่ระดับคะแนน					มัธยฐาน	ฐานนิยม	ผลต่างระหว่างมัธยฐานกับฐานนิยม	ควอไทล์ที่ 3 - ควอไทล์ที่ 1	ผลต่างของควอไทล์
	1	2	3	4	5					
RELY 1	0	1	1	8	3	4	4	0.00	4.00 - 4.00	0.00
2	0	0	0	10	3	4	4	0.00	4.00 - 4.00	0.00
3	0	0	1	9	2	4	4	0.00	4.00 - 4.00	0.00
4	0	0	3	6	4	4	4	0.00	5.00 - 4.00	1.00
DATA 1	0	0	0	8	5	4	4	0.00	5.00 - 4.00	1.00
2	0	0	4	7	2	4	4	0.00	4.00 - 3.00	1.00
CPLX 1	0	0	3	6	4	4	4	0.00	5.00 - 4.00	1.00
2	0	0	0	7	6	4	4	0.00	5.00 - 4.00	1.00
3	0	0	0	5	8	5	5	0.00	5.00 - 4.00	1.00
4	0	0	2	7	4	4	4	0.00	5.00 - 4.00	1.00
5	0	2	4	5	2	4	4	0.00	4.00 - 3.00	1.00
6	0	0	4	8	1	4	4	0.00	4.00 - 3.00	1.00
7	0	1	1	10	1	4	4	0.00	4.00 - 4.00	0.00
8	0	0	4	6	3	4	4	0.00	4.00 - 3.00	1.00
9	1	0	4	6	2	4	4	0.00	4.00 - 3.00	1.00
10	0	0	3	7	3	4	4	0.00	4.00 - 4.00	0.00
11	0	0	4	7	2	4	4	0.00	4.00 - 3.00	1.00
12	0	1	5	6	1	4	4	0.00	4.00 - 3.00	1.00
13	0	0	2	7	4	4	4	0.00	5.00 - 4.00	1.00
14	0	0	0	8	5	4	4	0.00	5.00 - 4.00	1.00

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.2 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 2(ต่อ)

คำถาม	ความถี่ของแต่ระดับคะแนน					มัธยฐาน	ฐานนิยม	ผลต่างระหว่างมัธยฐานกับฐานนิยม	ควอไทล์ที่ 3 - ควอไทล์ที่ 1	ผลต่างของควอไทล์
	1	2	3	4	5					
CPLX 15	0	0	1	8	4	4	4	0.00	5.00 - 4.00	1.00
16	0	1	0	9	3	4	4	0.00	4.00 - 4.00	0.00
17	0	1	1	6	5	4	4	0.00	5.00 - 4.00	1.00
18	0	2	2	9	0	4	4	0.00	4.00 - 3.00	1.00
19	0	0	2	8	3	4	4	0.00	4.00 - 4.00	0.00
20	0	0	5	8	0	4	4	0.00	4.00 - 3.00	1.00
21	1	0	5	6	1	4	4	0.00	4.00 - 3.00	1.00
22	0	0	0	2	11	5	5	0.00	5.00 - 5.00	0.00
23	1	0	0	6	6	4	5	-1.00	5.00 - 4.00	1.00
24	1	0	2	8	2	4	4	0.00	4.00 - 4.00	0.00
25	0	1	1	7	4	4	4	0.00	5.00 - 4.00	1.00
26	0	0	1	3	9	5	5	0.00	5.00 - 4.00	1.00
27	2	1	2	6	2	4	4	0.00	4.00 - 3.00	1.00
28	0	1	2	5	5	4	4	0.00	5.00 - 4.00	1.00
29	0	1	3	6	3	4	4	0.00	4.00 - 3.00	1.00
30	0	1	2	7	3	4	4	0.00	4.00 - 4.00	0.00
31	0	0	3	6	4	4	4	0.00	5.00 - 4.00	1.00
32	0	0	3	6	4	4	4	0.00	5.00 - 4.00	1.00
33	0	0	1	6	6	4	4	0.00	5.00 - 4.00	1.00
RUSE 1	0	0	1	6	6	4	5	-1.00	5.00 - 4.00	1.00
2	0	0	2	6	5	4	4	0	5.00 - 4.00	1.00
3	0	0	2	5	6	4	5	-1.00	5.00 - 4.00	1.00
4	1	1	3	4	4	4	4	0	5.00 - 3.00	2.00
5	1	1	1	7	3	4	4	0	4.00 - 4.00	0.00

ตารางที่ ข.2 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 2(ต่อ)

คำถาม	ความถี่ของแต่ระดับคะแนน					มัธยฐาน	ฐานนิยม	ผลต่างระหว่าง มัธยฐานกับฐานนิยม	ควอไทล์ที่ 3 - ควอไทล์ที่ 1	ผลต่างของ ควอไทล์
	1	2	3	4	5					
DOCU 1	0	0	0	2	11	5	5	0.00	5.00 - 5.00	0.00
2	0	0	1	1	11	5	5	0.00	5.00 - 5.00	0.00
3	0	0	3	4	6	4	4	0.00	5.00 - 4.00	1.00
4	0	0	1	9	3	4	4	0.00	4.00 - 4.00	0.00
TIME	0	0	2	9	2	4	4	0.00	4.00 - 4.00	0.00
STOR	0	0	2	6	5	4	4	0.00	5.00 - 4.00	1.00
PVOL 1	0	2	3	7	1	4	4	0.00	4.00 - 3.00	1.00
2	0	0	3	7	3	4	4	0.00	4.00 - 4.00	0.00
3	0	0	5	6	2	4	4	0.00	4.00 - 3.00	1.00
4	0	2	2	8	1	4	4	0.00	4.00 - 3.00	1.00
ACAP 1	0	0	3	7	3	4	4	0.00	4.00 - 4.00	0.00
2	0	0	2	7	4	4	4	0.00	5.00 - 4.00	1.00
PCAP 1	0	0	3	6	4	4	4	0.00	5.00 - 4.00	1.00
2	0	0	2	4	7	5	5	0.00	5.00 - 4.00	1.00
AEXP 1	0	0	1	5	7	5	5	0.00	5.00 - 4.00	1.00
2	0	0	2	3	8	5	5	0.00	5.00 - 4.00	1.00
PEXP 1	0	0	1	6	6	4	4	0.00	5.00 - 4.00	1.00
2	0	0	1	6	6	4	5	-1.00	5.00 - 4.00	1.00
LTEX 1	0	0	3	6	4	4	4	0.00	5.00 - 4.00	1.00
2	0	0	2	6	5	4	4	0.00	5.00 - 4.00	1.00
PCON 1	0	0	2	7	4	4	4	0.00	5.00 - 4.00	1.00
2	1	0	4	7	1	4	4	0.00	4.00 - 3.00	1.00
3	1	0	5	6	1	4	4	0.00	4.00 - 3.00	1.00
4	0	1	5	6	1	4	4	0.00	4.00 - 3.00	1.00
5	0	0	3	4	6	4	5	-1.00	5.00 - 4.00	1.00

ตารางที่ ข.2 แสดงความถี่ของระดับคะแนนจากแบบสอบถามรอบที่ 2(ต่อ)

คำถาม	ความถี่ของแต่ละระดับคะแนน					มัธยฐาน	ฐานนิยม	ผลต่างระหว่างมัธยฐานกับฐานนิยม	ควอไทล์ที่ 3 - ควอไทล์ที่ 1	ผลต่างของควอไทล์
	1	2	3	4	5					
TOOL 1	0	1	3	7	2	4	4	0.00	3.00 - 4.00	1.00
2	0	0	2	7	4	4	4	0.00	4.00 - 5.00	1.00
3	0	0	1	7	5	4	4	0.00	4.00 - 5.00	1.00
4	0	0	1	9	3	4	4	0.00	4.00 - 4.00	0.00
5	0	0	3	8	2	4	4	0.00	4.00 - 4.00	0.00
6	0	0	3	9	1	4	4	0.00	4.00 - 4.00	0.00
7	0	0	4	7	2	4	4	0.00	3.00 - 4.00	1.00
SITE 1	0	0	0	9	4	4	4	0.00	4.00 - 5.00	1.00
2	0	1	3	7	2	4	4	0.00	3.00 - 4.00	1.00
3	0	0	0	7	6	4	4	0.00	4.00 - 5.00	1.00
4	0	1	0	9	3	4	4	0.00	4.00 - 4.00	0.00
5	0	1	3	6	2	4	4	0.00	3.00 - 4.00	1.00
6	1	0	3	6	3	4	4	0.00	3.00 - 4.00	1.00
7	0	0	3	7	3	4	4	0.00	4.00 - 4.00	0.00
8	0	1	3	7	2	4	4	0.00	3.00 - 4.00	1.00
9	0	0	1	9	3	4	4	0.00	4.00 - 4.00	0.00
10	0	0	1	8	4	4	4	0.00	4.00 - 5.00	1.00
11	0	0	0	9	4	4	4	0.00	4.00 - 5.00	1.00
SCED 1	1	0	2	1	9	5	5	0.00	4.00 - 5.00	1.00
2	0	1	2	5	5	4	5	0.00	4.00 - 5.00	1.00

จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

ตัวอย่างซอฟต์แวร์ที่ต้องการพัฒนา

1. ระบบการสั่งซื้อทางไปรษณีย์(MAIL ORDER SYSTEM)

บริษัท BESTBUY ดำเนินกิจการเกี่ยวกับการขายสินค้าทางไปรษณีย์ (Mail Order) โดยจะขายสินค้าให้กับลูกค้าที่เป็นสมาชิกบัตรเครดิต VISA หรือ MASTER เท่านั้น ลูกค้าสามารถเลือกซื้อสินค้าได้จาก Catalog ที่บริษัทจัดส่งไปให้ทุกๆ 3 เดือน ทั้งสิ้น 50,000 ฉบับ โดยบริษัทนำรายชื่อและที่อยู่ของลูกค้ามาจากธนาคาร Catalog จะประกอบไปด้วย ชื่อสินค้า ราคาสินค้า ราคาพิเศษใช้ร่วมกับคะแนนโบนัส ราคาแรกซื้อโดยใช้คะแนนโบนัส Order Form และวิธีการส่งซื้อ ลูกค้าสามารถสั่งซื้อสินค้าต่างๆใน Catalog ได้ตามช่วงเวลาที่กำหนดไว้ใน Catalog เท่านั้น การสั่งซื้อสินค้าของลูกค้าทำได้ 2 วิธีคือ

1.การสั่งซื้อทางไปรษณีย์(Order Form)

2.การสั่งซื้อทางโทรศัพท์

ในปัจจุบันการดำเนินการของบริษัทมีสินค้ากว่า 1000 ชนิด และ มีOrderประมาณ 500 order ต่อวัน โดยจะเปิดทำงาน 8.00 น. ถึง 22.00 น.ซึ่งบริษัทมีรายได้ประมาณปีละ 30,000,000 บาทต่อปี ซึ่งมีกระบวนการดำเนินงานดังนี้

1. รับรายการสั่งสินค้าและตรวจสอบเบื้องต้น

เมื่อรับOrder Form หรือโทรศัพท์สั่งสินค้า Order Clerk ต้องตรวจสอบว่าผู้สั่งซื้อมีคุณสมบัติเป็นลูกค้าของบริษัทหรือไม่ โดยเปรียบเทียบกับข้อมูลสมาชิกของบัตรเครดิตที่ได้จากธนาคาร ที่ข้อมูลลูกค้าที่เขียนใน Order Form หรือข้อมูลส่วนตัวอื่นๆที่ Order Clerk จะสอบถามในกรณีที่สั่งซื้อทางโทรศัพท์ ถ้าข้อมูลไม่ถูกต้องจะปฏิเสธการสั่งนั้น

2. ตรวจสอบสินค้าในคลังสินค้าและตรวจสอบสถานะลูกค้า

Order Clerk จะตรวจสอบรายการสินค้าที่ลูกค้าสั่งซื้อกับคลังสินค้า

2.1ในกรณีสินค้าหมด หรือ มีจำนวนน้อยกว่าที่ลูกค้าต้องการ พนักงานจะสอบถามทันทีว่ารอสินค้าได้ไหม ถ้าลูกค้าต้องการรอสินค้า Order Clerk จะสอบถามข้อมูลกับลูกค้าเพื่อกรอก Order Form และจะต้องทำBack Order Form เสนอให้ผู้จัดการพิจารณาสั่งซื้อสินค้าจาก Supplier

2.2.ในกรณีที่มีสินค้า Order Clerk จะทำการลบสินค้าที่มีอยู่ในคลัง และบันทึกผลลัพธ์ลงในสมุดบันทึกรายการสินค้า ถ้าเป็นโทรศัพท์สั่งซื้อ Order Clerk จะสอบถามข้อมูลของลูกค้าเพื่อกรอก Order Form ไปตรวจสอบ Credit line และ/หรือจำนวนโบนัสของลูกค้าที่ธนาคาร หากลูกค้ามีจำนวนวงเงินและ/หรือจำนวนโบนัสไม่พอชำระสินค้าตามวิธีการชำระเงิน ที่ลูกค้าเลือกไว้ใน Order Formแล้ว Order Clerk ก็จะเก็บ Order Form ไว้ และแก้ไขจำนวนสินค้าใหม่

3. ดำเนินการเพื่อจัดส่ง

ถ้าลูกค้าสามารถชำระค่าสินค้าได้ Order Clerk จะทำ Order Form 2 ชุด โดยสำเนา Order Form ชุดที่ 1 ส่งให้ Packing Clerk เพื่อจัดทำPacking Slip บรรจุสินค้าลงหีบห่อ และชื่อที่อยู่ของผู้รับสินค้าติดที่พัสดุนั้น ส่วนสำเนา ชุดที่ 2 Order Clerk จะรวบรวมไว้เพื่อเก็บเงินกับธนาคารทุกๆสิ้นเดือน

2. ขอบเขตของการพัฒนา Mail Order System

1.ระบบ Mail Order ในส่วนของการสั่งซื้อจะครอบคลุมทั้งการสั่งซื้อทางโทรศัพท์ และไปรษณีย์ ซึ่งจะมีส่วนย่อยดังนี้

การสร้าง Order Form

การสร้าง Back Order Form

การติดตามสถานะการสั่งซื้อ

การพิมพ์ Packing Slip

2.ระบบสามารถติดตามความเคลื่อนไหวของสินค้าในคลังสินค้า และ Maintain สินค้าได้

3.ระบบจะ Maintain ฐานข้อมูลของลูกค้า ผ่านสื่ออิเล็กทรอนิกส์ (Diskette) ที่ทางธนาคารจัดส่งให้เมื่อมีการเปลี่ยนแปลงข้อมูล

4.ระบบไม่สามารถอนุมัติการสั่งจ่ายของลูกค้าได้ ยกเว้นธนาคารเจ้าของบัตรอนุมัติมาให้

5.ระบบไม่ครอบคลุมการทำบัญชี รับ-จ่ายของบริษัท

6.ระบบไม่สามารถทำการกำหนดสินค้าในCatalog หรือจัดทำ Catalog

7.การทำการคำนวณคะแนนโบนัสเป็นหน้าที่ธนาคาร ระบบมีหน้าที่เพียงรับข้อมูลจากธนาคารเท่านั้น

8.การจัดส่งสินค้าให้กับลูกค้า ทางบริษัทจะโอนความรับผิดชอบให้กับทางกรมไปรษณีย์ หากเกิดความเสียหายหรือส่งไม่ถึงมือผู้รับ

3. ข้อกำหนดเบื้องต้น

1. Hardware ใช้สถาปัตยกรรม Clint/Server

PC Computer(Client)	3 เครื่อง
Server	1 เครื่อง
Printer	3 เครื่อง

อุปกรณ์เครือข่าย 1 ชุดสำหรับ 3 Workstation

2. Software

Oracle 8 บน Server

Windows 95/98 บนClient

Windows NT บน Server

3. Person

สามารถพิมพ์ดีดได้ทั้ง ไทย และอังกฤษ

สามารถใช้งานWindows 95 /98 ได้

4.TIME

ระบบจะต้องเสร็จภายใน 3 เดือน

4. งบประมาณ

Hardware			
PC Client	3	30,000	90,000
Pentium II 300			
Ram 32 MB			
HD 4G			
PC-Server	1	150,000	150,000
Pentium III 650			
Ram 128 MB			
HD 8G			
Tape Backup			
UPS ขนาด 550VA / 1050VA	1	4,500	4,500
Printer EPSON 21701	1	21,000	21,000
Printer EPSON 870	2	10,000	20,000
Software			
Windows NT	1	20,000	20,000
Windows 95	3	15,000	45,000
Windows 98	1	10,000	10,000
Visual Basic	1	20,000	20,000
Oracle	1	20,000	20,000
Network			
Network Adapter	4	1,200	1,200
ค่าพัฒนาSoftware			550,000
ค่าติดตั้งระบบ			20,000
รวม			971,700

5. ระบบการทำงานต่าง

1.สามารถตรวจสอบคุณสมบัติของลูกค้าได้โดยการสอบจากคอมพิวเตอร์

2.สามารถบันทึกข้อมูลการสั่งซื้อโดยใช้คอมพิวเตอร์

3.ข้อมูลการสั่งซื้อสินค้าจะนำไปเปลี่ยนแปลงจำนวนสินค้าในคลังทันที ทำให้สามารถรู้ยอดของสินค้าในคลัง

4.สามารถเพิ่ม ลบ แก้ไขข้อมูลในคลังสินค้าได้

5.สามารถLoad ข้อมูลของลูกค้าจากธนาคารเข้าสู่ระบบโดยใช้แผ่นดิสก์

6.สามารถออกOrder Form, Back Order Form, Packing Slip และที่อยู่ของลูกค้าต้องการให้ส่งของไปด้วยเครื่องพิมพ์

7.จัดทำรายงานเพื่อช่วยในการบริการ ได้แก่

รายงานแสดงจำนวนการสั่งซื้อต่อเดือน หรือ ต่อปี

รายงานสรุปยอดการสั่งซื้อสินค้าที่มีจำนวนการสั่งซื้อสูง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายงานแสดงสถานภาพการสั่งซื้อของลูกค้า

รายงานแสดงสินค้าที่ต้องสั่งซื้อเพื่อเสนอผู้จัดการ

รายงานแสดงสินค้าที่มีการสั่งซื้อมากที่สุด

8.สามารถช่วยในการติดตามลูกค้า เช่น กรณีที่สินค้าที่ลูกค้าสั่งหมด เมื่อมีสินค้าในคลังควรติดต่อกับไปยังลูกค้า

6. ผู้ใช้งานระบบ

1.ผู้จัดการ

อนุมัติการสั่งซื้อสินค้าและควบคุมการดำเนินการ

2.Order Clerk

รับ Order Form จากลูกค้าไปรษณีย์

ตรวจสอบว่าลูกค้าเป็นสมาชิกบัตรเครดิตหรือไม่

ตรวจสอบยอดการสั่งซื้อสินค้า

ตรวจยอดคงเหลือของสินค้าในคลังสินค้า

ตรวจสอบรายการสินค้าที่ลูกค้าต้องการในคลังสินค้า

กรอก Order Form จากลูกค้าทางโทรศัพท์

ส่งสำเนา Order Form ให้Packing Clerk

ทำ back Order Form ส่งให้ผู้จัดการ

3.Packing Clerk มีหน้าที่

บรรจุสินค้าตาม Order Form

ติดPacking Slip ไปกับสินค้าเพื่อทำการจัดส่ง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบประเมินค่าใช้จ่ายของการพัฒนาซอฟต์แวร์
ตามแนวความคิดของโคโคโม2
ชุดที่ 1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

<u>คำชี้แจง</u>

1.กรุณาทำความเข้าใจกับโครงการที่จะประเมินค่าใช้จ่าย

2.กรุณาตอบคำถามให้ครบทุกข้อและเลือกลักษณะซอฟต์แวร์ที่ท่านต้องการ เพียงข้อเดียวใน แต่ละตัวขับค่าใช้จ่าย(Cost Driver)หรือตัวขับค่าใช้จ่ายย่อย

**** กรุณาทำเครื่องหมายถูก(✓)ใน☐ที่ท่านเลือกและตอบได้เพียง 1 คำตอบ ในแต่ละข้อ

1.ความน่าเชื่อถือของซอฟต์แวร์ที่ต้องการ (Required Software Reliability: RELY)

☐ ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความไม่สะดวกในการทำงานแต่สามารถดำเนินงานต่อไปได้

☐ ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายเพียงเล็กน้อยในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม

☐ ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายปานกลางในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม

☐ ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความสูญเสียทางการเงินอย่างมาก

☐ ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความเสี่ยงต่อชีวิตมนุษย์

2.ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ (Data Base Size : DATA)

☐ ขนาดของฐานข้อมูล(ไบต์) / ขนาดของซอฟต์แวร์(จำนวนบรรทัดของโปรแกรม) น้อยกว่า 10

☐ ขนาดของฐานข้อมูล(ไบต์) / ขนาดของซอฟต์แวร์(จำนวนบรรทัดของโปรแกรม) มากกว่าเท่ากับ 10 แต่น้อยกว่า 100

☐ ขนาดของฐานข้อมูล(ไบต์) / ขนาดของซอฟต์แวร์(จำนวนบรรทัดของโปรแกรม)มากกว่าเท่ากับ 100 แต่น้อยกว่า 1000

☐ ขนาดของฐานข้อมูล(ไบต์) / ขนาดของซอฟต์แวร์(จำนวนบรรทัดของโปรแกรม) มากกว่า 1000

3. ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์(Complexity :CPLX)

3.1 การดำเนินการควบคุม(Control Operation)

☐การดำเนินการ(programming operation)ของซอฟต์แวร์ ส่วนใหญ่จะเป็นการดำเนินการแบบตรงไปข้างหน้า (straight-line code) แต่ก็มีลักษณะการดำเนินการแบบเป็นเงื่อนไขแต่เป็นเงื่อนไขที่ไม่ซับซ้อน(non-nested structured) เช่น DO, CASE, IF-THEN-ELSE นอกจากนี้อาจมีการใช้โมดุลง่ายๆ เช่น การทำโปรซีเดอร์คอร์ (procedure call) เป็นต้น

☐ การดำเนินการของซอฟต์แวร์จะมีโครงสร้างที่ซับซ้อน(nesting of structured programming) และโดยส่วนใหญ่จะเป็นแบบมีเงื่อนไข เช่น มีลูป WHILE ซ้อน IF-THEN-ELSE หรือ IF-THEN-ELSE ซ้อน IF-THEN-ELSE เป็นต้น

☐การดำเนินการของซอฟต์แวร์โดยส่วนใหญ่จะเป็นแบบมีเงื่อนไขที่ซับซ้อน จึงมีการใช้ตารางตัดสินใจ (decision table) <u>หรือ</u> มีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบ call back หรือแบบการส่งข้อความ (message passing) รวมทั้งมีการดำเนินการที่สนับสนุนการประมวลผลแบบกระจายบนเครือข่าย(distribute processing) เช่น ระบบclient/server

☐โครงสร้างของการดำเนินการมีความซับซ้อนมากซึ่งประกอบด้วยการดำเนินการที่เป็นแบบเงื่อนไขซ้อนเงื่อนไข <u>หรือ</u>มีการใช้คิว(queue) และกองซ้อน(stack) เป็นตัวควบคุมลำดับการดำเนินการ <u>หรือ</u>มีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน (homogeneous) คือ มียี่ห้อหรือเวอร์ชันเดียวกัน <u>หรือ</u>มีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด(soft real time)

☐การดำเนินการจะมีลักษณะเป็นแบบการเวียนบังเกิด (recursive) หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความความต่างกัน(heterogeneous) คือ มีชี่ห้อหรือเวอร์ชันต่างกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด (hard real time)

☐การดำเนินการจะมีลักษณะเป็นแบบการเวียนบังเกิด (recursive) หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความความต่างกัน(heterogeneous) คือ มีชี่ห้อหรือเวอร์ชันต่างกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด (hard real time)

☐การดำเนินการของซอฟต์แวร์จะต้องควบคุมการทำงานของฮาร์ดแวร์หรืออุปกรณ์โดยตรง (Microcode level control) หรือมีตารางการใช้ทรัพยากร(resource)ที่สามารถแก้ไขได้โดยอัตโนมัติตามความเหมาะสม หรือมีการประมวลผลแบบกระจาย(distribute processing) ที่ได้ผลลัพธ์แบบทันทีทันใด(hard real time)

3.2 การดำเนินการคำนวณ(Computation Operation)

☐การดำเนินการคำนวณสามารถหาค่าจากสมการพีชคณิตหรือสมการเส้นตรง เช่น ทำการบวก การลบ การคูณ การหาร ตัวอย่างเช่น A=B+C*(D-E)

☐ การดำเนินการคำนวณสามารถหาค่าจากสมการที่มีความยากระดับกลาง(moderate-level expressions) คือ นอกจากทำการบวก การลบ การคูณ การหาร แล้วยังสามารถทำการยกกำลัง การหารากการคำนวณหาค่าตรีโกณมิติ ตัวอย่างเช่น D= SQRT(B^2 – 4*A*C)

☐ การดำเนินการคำนวณมีการใช้รูทีน(routine) ที่เป็นมาตราฐานทางคณิตศาสตร์ และสถิติ เช่นการหาส่วนเบี่ยงเบนมาตราฐาน การหาค่า Z เป็นต้น นอกจากนี้ยังคำนวณหาค่าของเมตริกซ์(matrix)และเวกเตอร์(vector)ได้

☐ การดำเนินการคำนวณสามารถวิเคราะห์เชิงตัวเลข(numerical analysis)ขั้นพื้นฐานได้ เช่น การหาคำตอบของสมการโดยวิธีการของนิวตัน(Newton's Method) หรือโดยวิธีการของ Guassian การแก้สมการเชิงอนุพันธ์เป็นต้น

☐ การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างแน่นอน เช่น การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน (non-singular matrix equation) การหาค่าจากสมการเชิงอนุพันธ์ย่อย(partial differential equation) และการหารูปแบบของความสัมพันธ์จากข้อมูล 2 ชุด

☐ การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขชนิดที่มีโครงสร้างไม่แน่นอน เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน(highly accurate analysis of noisy) การวิเคราะห์ข้อมูลแบบ stochastic

3.3 การดำเนินการที่เกี่ยวกับอุปกรณ์(Device-dependent Operation)

☐ คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์เป็นคำสั่งที่ใช้งานง่าย หรือเป็นภาษาระดับสูง เช่น การใช้คำสั่ง อ่าน (read) เขียน(write) ในภาษาพาสคาล(Pascal) ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล

☐ คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์ สามารถใช้คำสั่งที่เป็นพื้นฐานที่ภาษามีการเตรียมไว้ให้ เช่น คำสั่งเขียนข้อมูลลงแฟ้มข้อมูล(put) หรือ อ่านข้อมูลในแฟ้มข้อมูล(get) ในภาษาซี(C) โดยที่ผู้พัฒนาไม่จำเป็นต้องมีความรู้เกี่ยวกับอุปกรณ์ที่ใช้

☐ ผู้พัฒนาจะต้องมีความรู้อย่างเจาะจงเกี่ยวกับอุปกรณ์ที่เป็นตัวรับข้อมูลและตัวแสดงผล(input/output) โดยการใช้งานอุปกรณ์ต้องทำ 3 ขั้นตอนดังนี้คือ เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์นั้นและตรวจสอบความผิดพลาดในการทำงานของอุปกรณ์

☐ การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะกระทำในระดับกายภาพ(physical) เช่น การแปลงตำแหน่งของหน่วยความจำทางกายภาพ เพื่อช่วยในการค้นหาและการอ่าน และการกำหนดรูปแบบการซ้อนทับ(overlap)ในหน่วยความจำของอุปกรณ์

☐ การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะมีรูทีน(routines) สำหรับจัดการเมื่อเกิดการผิดพลาดในขณะส่งและรับข้อมูล มีการจัดการเกี่ยวกับการส่งข้อมูลในสายการสื่อสาร หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบเข้มงวด(performance-intensive embedded system) เช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน

☐ มีการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed operations) หรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบวิกฤต (performance-critical embedded systems) เช่น มีการบันทึกการทำงานของอุปกรณ์ทุกชนิดตลอดเวลา

3.4 การดำเนินการเกี่ยวกับการจัดการกับข้อมูล(Data Management Operation)

☐ การดำเนินการเกี่ยวกับการจัดการกับข้อมูลมีการใช้แถวลำดับ(array)ในหน่วยความจำหลัก เพื่อเก็บข้อมูลและสามารถใช้สอบถาม(query)และแก้ไข(update)ได้

☐ การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ แฟ้มข้อมูลสามารถใช้สอบถามและแก้ไขได้

☐ การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลได้หลายแฟ้มข้อมูลโดยจะเป็นแฟ้มส่งข้อมูลออกเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ จะไม่มีการดำเนินการกับแฟ้มข้อมูลที่เกิดในระหว่างการทำงาน เช่น log file

☐ การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถทำทริกเกอร์(trigger)โดยใช้ข้อมูลเพียงชุดเดียว สามารถทำการปฏิวัติโครงสร้างของข้อมูลที่ซับซ้อน(complex data restructuring) เช่นการเปลี่ยนคีย์ การเปลี่ยนความสัมพันธ์ระหว่างแฟ้มข้อมูล เป็นต้น

☐ การดำเนินการเกี่ยวกับการจัดการฐานข้อมูลสามารถทำกับข้อมูลที่เก็บไว้ในหลายสถานที่ได้ สามารถทำทริกเกอร์(trigger)กับข้อมูลจำนวนมากๆได้ สามารถค้นหาข้อมูลในฐานข้อมูลได้

☐ การดำเนินการเกี่ยวกับการจัดการกับข้อมูลเป็นแบบพลวัต(dynamic relational) คือสามารถเปลี่ยนแปลงโครงสร้างบางอย่างโดยอัตโนมัติตามสภาพของข้อมูลที่เก็บได้ มีโครงสร้างเชิงวัตถุ(object structures) มีการจัดการข้อมูลด้วยภาษาธรรมชาติ(natural language data management)

3.5 การดำเนินการส่วนต่อประสานกับผู้ใช้(User Interface Management Operation)

☐ มีโปรแกรมช่วยในการสร้าง(generators)รูปแบบการรับข้อมูลเข้า(input form) และการทำรายงาน (report) แบบง่าย เช่น มีwizardช่วย

☐ มีการใช้ตัวสร้างส่วนต่อประสานกับผู้ใช้(user interface)ที่ไม่ยุ่งยาก เช่น ใช้ตัวสร้างGUI (graphic user interface builders)

☐ มีการใช้งานส่วนต่อประสานกับผู้ใช้เป็นแบบวินโดว์ที่ไม่มีความซับซ้อนมาก เช่น มีการติดต่อแบบ windows95 เป็นต้น.

☐ มีการใช้งานส่วนต่อประสานกับผู้ใช้แบบวินโดว์ซึ่งสามารถใช้เสียงเป็นตัวรับและแสดงผล และยังใช้ติดต่อแบบมัลติมีเดียอย่างง่ายได้ด้วย

☐ มีการใช้ภาพ2 มิติ หรือ 3มิติที่มีความซับซ้อนไม่มาก มีการใช้ภาพที่เป็นพลวัต(dynamic graphic) และมีการใช้มัลติมีเดีย(multimedia)ในการติดต่อกับผู้ใช้

☐ มีการใช้มัลติมีเดียที่ซับซ้อน เช่น วีดิโอคอนเฟอร์เร็น (video conference) และมีการใช้ภาพเสมือนจริง(virtual reality) ในการติดต่อกับผู้ใช้

4 ความต้องการที่จะนำผลิตภัณฑ์กลับมาใช้ใหม่ (Required Reusability :RUSE)

☐ ไม่มีการออกแบบสำหรับการนำโปรแกรมกลับมาใช้ใหม่ในอนาคต

☐ มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโครงการ(Project)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของ Microsoft Word6 มาใช้กับ Microsoft Word7 เป็นต้น

☐ มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโปรแกรม(Program)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำบางส่วนของโปรแกรมของMicrosoft Excel มาใช้กับ Microsoft Word เป็นต้น

☐ มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมที่ไม่ได้อยู่ในระบบ(Product Line) เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของระบบซื้อ-ขาย มาใช้กับระบบธนาคาร เป็นต้น

☐ มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมไปใช้ได้ในหลายๆระบบ เช่น นำส่วนของโปรแกรมของระบบบัญชี ระบบซื้อ-ขาย และระบบงานบุคคล มาใช้กับระบบธนาคาร เป็นต้น

5 ความต้องการเอกสารที่ตรงกับวงจรชีวิต(Documentation match to life-cycle needs: DOCU)

☐ ไม่มีการจัดทำเอกสารการพัฒนาซอฟต์แวร์

☐ เอกสารจัดทำไม่ครอบคลุมในหลายๆขั้นของวงจรชีวิต

☐ เอกสารจัดทำได้ครบตามข้อกำหนดขั้นต่ำของวงจรชีวิต

☐ เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่ใช้ภายในองค์กรแต่ไม่ได้เป็นข้อกำหนดไว้ในวงจรชีวิต เพื่อทำให้เกิดความสมบูรณ์มากขึ้น

☐ เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่มีความละเอียดเพื่อให้เกิดความสมบูรณ์มากที่สุด

6.การจำกัดเวลาที่ใช้ในการประมวลผล (Execute Time Constraint: TIME)

☐ เวลาที่ซอฟต์แวร์เข้ากระทำการน้อยกว่าหรือเท่ากับ 50% ของเวลาที่ซีพียูมีให้

☐ เวลาที่ซอฟต์แวร์เข้ากระทำการมากกว่า 50% ของเวลาที่ซีพียูมีให้แต่น้อยกว่าหรือเท่ากับ 70% ของเวลาที่ซีพียูมีให้

☐ เวลาที่ซอฟต์แวร์เข้ากระทำการมากกว่า 70% ของเวลาที่ซีพียูมีให้แต่น้อยกว่าหรือเท่ากับ 85% ของเวลาที่ซีพียูมีให้

☐ เวลาที่ซอฟต์แวร์เข้ากระทำการมากกว่า 85% ของเวลาที่ซีพียูมีให้แต่น้อยกว่าหรือเท่ากับ 95% ของเวลาที่ซีพียูมีให้

7.การจำกัดหน่วยความจำหลัก (Main Storage Constraint: STOR)

☐ ใช้งานขนาดของหน่วยความจำหลักน้อยกว่าหรือเท่ากับ 50% ของขนาดของหน่วยความจำหลักมีให้

☐ ใช้งานขนาดของหน่วยความจำหลักมากกว่า 50% แต่น้อยกว่าหรือเท่ากับ 70% ของขนาดของหน่วยความจำหลักมีให้

☐ ใช้งานขนาดของหน่วยความจำหลักมากกว่า 70% แต่น้อยกว่าหรือเท่ากับ 85% ของขนาดของหน่วยความจำหลักมีให้

☐ ใช้งานขนาดของหน่วยความจำหลักมากกว่า 85% แต่น้อยกว่าหรือเท่ากับ 95% ของขนาดของหน่วยความจำหลักมีให้

8.การเปลี่ยนแปลงได้ง่ายของแพลทฟอร์ม (Platform Volatility: PVOL)

☐ จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 12 เดือนหรือทำการปรับปรุงบางส่วนทุกๆ 1 เดือน

☐ จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 6 เดือนหรือทำการปรับปรุงบางส่วนทุกๆ 2 สัปดาห์

☐ จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 เดือนหรือทำการปรับปรุงบางส่วนทุกๆ 1 สัปดาห์

☐ จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 สัปดาห์หรือทำการปรับปรุงบางส่วนทุกๆ 2 วัน

9.ความสามารถในการวิเคราะห์ (Analyst Capability: ACAP)

☐ เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25

☐ เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45

☐ เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65

☐ เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85

☐ เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

10.ความสามารถของโปรแกรมเมอร์ (Programmer Capability: PCAP)

☐ เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25

☐ เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45

☐ เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65

☐ เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85

☐ เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

11.ประสบการณ์การพัฒนาแอปพลิเคชั่น (Application Experience: AEXP)

☐ น้อยกว่าหรือเท่ากับ 2 เดือน

☐ มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน

☐ มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี

☐ มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ3 ปี

☐ มากกว่า 3 ปี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

12.ประสบการณ์การใช้แพลทฟอร์ม(Platform Experience: PEXP)

☐ น้อยกว่าหรือเท่ากับ 2 เดือน

☐ มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน

☐ มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี

☐ มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ3 ปี

☐ มากกว่า 3 ปี

13.ประสบการณ์การใช้ภาษาโปรแกรมและเครื่องมือ(Language and Tool Experience: LTEX)

☐ น้อยกว่าหรือเท่ากับ 2 เดือน

☐ มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน

☐ มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี

☐ มากกว่า 1 ปีแต่น้อยกว่าหรือเท่ากับ3 ปี

☐ มากกว่า 3 ปี

14.ความต่อเนื่องของบุคลากร(Personnel Continuity: PCON)

☐ บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 48%ต่อปี

☐ บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 24%ต่อปี แต่น้อยกว่า48%ต่อปี

☐ บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 12%ต่อปี แต่น้อยกว่า24%ต่อปี

☐ บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 6%ต่อปี แต่น้อยกว่า12%ต่อปี

☐ บุคลากรมีการเปลี่ยนแปลงน้อยกว่า 6%ต่อปี

15.การใช้เครื่องมือซอฟต์แวร์ (Use of software tool: TOOL)

☐ เครื่องมือสามารถสามารถเขียน(Code) บรรณาธิกร(Edit) และตรวจสอบความผิดพลาดในการเขียนโปรแกรมได้(Debug)

☐ เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำส่วนเสริมหน้า(Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงแรกๆ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา(Required Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase) หรือ ช่วยในขั้นตอนการออกแบบ(Design Phase) หรือแบบเครื่องมือเคสที่ทำส่วนเสริมหลัง (Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต (SDLC)ในช่วงหลัง เช่น ช่วยในขั้นตอนของการทำให้เกิดผล(Implementation Phase) ช่วยในขั้นตอนของการรวม(Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา(Maintenance Phase)

☐ เครื่องมือที่ใช้พัฒนาซอฟต์แวร์สามารถใช้ได้ในทุกขั้นตอนของวงจรชีวิต(SDLC) ในขั้นพื้นฐานได้ นอกจากนี้ยังสามารถทำการรวม(integrated)บางขั้นตอนของวงจรชีวิตได้

☐ เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นของวงจรชีวิตและยังสามารถทำการรวม (Integrated)ทุกขั้นของวงจรชีวิตได้

☐ เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นตามรูปแบบของวงจรชีวิต(SDLC) นอกจากนี้ยังสามารถทำการรวมกัน(Integrated)โดยระเบียบวิธีการ(Method) กระบวนการ(Process) หรือการนำกลับมาใช้ (Reuse)

16.การพัฒนาในหลายสถานที่ (Multisite Development: SITE)

16.1สถานที่ที่พัฒนา

☐ สถานที่ที่พัฒนาซอฟต์แวร์มีทั้งในประเทศและต่างประเทศ

☐ สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด และมีบริษัทที่พัฒนาหลายบริษัท

☐ สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด หรือ มีบริษัทที่พัฒนาหลายบริษัท หรือ ตั้งอยู่ในจังหวัดที่ไม่สำคัญเพียงจังหวัดเดียว เช่น หนองบัวลำพูล ปัตตานี เป็นต้น

☐ สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในกรุงเทพฯ หรือจังหวัดสำคัญๆ(กรุงเทพฯ เชียงใหม่ นครราชสีมา ภูเก็ต สงขลา ..)

☐ สถานที่ที่พัฒนาซอฟต์แวร์มีหลายตึก หรือ เป็นหมู่ตึก

☐ สถานที่ที่พัฒนาซอฟต์แวร์จะอยู่ในบริเวณเดียวกัน เช่น อยู่ในตึกเดียวกัน อยู่ในชั้นเดียวกัน เป็นต้น

16.2. การสื่อสารระหว่างพัฒนา

☐ เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์รวม หรือ จดหมาย

☐ เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์ส่วนตัวหรือแฟกส่วนตัว

☐ เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณแคบ เช่น ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น

☐ เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น

☐ เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์

☐ เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ มัลติมีเดียที่สามารถโต้ตอบได้

17.กำหนดเวลาของการพัฒนาที่ต้องการ (Required Development Schedule: SCED)

☐ เวลาในการพัฒนาน้อยกว่าหรือเท่ากับ 75% ของเวลาปกติ

☐ เวลาในการพัฒนามากกว่า 75%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ 85%

☐ เวลาในการพัฒนามากกว่า 85%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ100%

☐ เวลาในการพัฒนามากกว่า 100%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ130%

☐ เวลาในการพัฒนามากกว่า 130%ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ160%

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แบบประเมินค่าใช้จ่ายของการพัฒนาซอฟต์แวร์
ตามแนวความคิดของโคโคโม2
ชุดที่ 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

<u>คำชี้แจง</u>

1.กรุณาทำความเข้าใจกับโครงการที่จะประเมินค่าใช้จ่าย

2.กรุณาตอบคำถามให้ครบทุกข้อและใกล้เคียงความเป็นจริงมากที่สุด

ในกรณีตอบว่าอื่นๆ หมายถึง ผู้ประเมินไม่ทราบ ไม่ต้องการตอบ หรือไม่แน่ใจ

**** กรุณาทำเครื่องหมายถูก(✓)ใน ☐ ที่ท่านเลือกหรือเติมข้อมูลลงในช่องว่าง

1.ความน่าเชื่อถือของซอฟต์แวร์ที่ต้องการ (Required Software Reliability: RELY)	ใช่	ไม่ใช่	อื่นๆ
1) ซอฟต์แวร์ที่พัฒนาจะถูกนำไปใช้ในด้านที่อาจเกิดความเสี่ยงกับชีวิตมนุษย์ หรือมีผลกระทบต่อชีวิตมนุษย์ ใช่หรือไม่	☐	☐	☐
2) ซอฟต์แวร์ที่จะพัฒนาเป็นซอฟต์แวร์ที่เกี่ยวข้องกับรายรับ-รายจ่ายขององค์กรเป็นอย่างมาก ซึ่งเมื่อซอฟต์แวร์ไม่สามารถทำงานได้จะกระทบต่อรายรับ-รายจ่ายขององค์กร ใช่หรือไม่	☐	☐	☐
3) ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติ ท่านจำเป็นต้องดำเนินการให้มีการแก้ไขทันที ใช่หรือไม่	☐	☐	☐
4) ท่านสามารถดำเนินงานในลักษณะนั้นต่อได้ ถึงแม้ว่าซอฟต์แวร์จะไม่สามารถทำงานได้ก็ตาม ใช่หรือไม่ เช่นการใช้บุคลากรทำงานแทน เป็นต้น	☐	☐	☐

2.ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ (Data Base Size : DATA)

1) ขนาดของหน่วยความจำสำรองที่ใช้เก็บข้อมูล(กิโลไบต์:KB)

☐ ในกรณีที่ท่านทราบ

ขนาดของหน่วยความจำสำรองที่ใช้เก็บข้อมูลประมาณ.....KB

☐ ในกรณีที่ท่านไม่ทราบ

ให้ท่านประมาณจำนวนแฟ้มข้อมูล(Files) จำนวนรายการ(Transactions) จำนวนเขตข้อมูล(Fields) และจำนวนอักษร

	Files	จำนวนRecord	ความยาวของRecord
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
ฯลฯ			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2) ขนาดของซอฟต์แวร์ที่จะพัฒนา(จำนวนบรรทัดของโปรแกรม: SLOC)

☐ ในกรณีที่ท่านทราบ

ขนาดของซอฟต์แวร์ที่จะพัฒนามีจำนวนบรรทัดของโปรแกรมประมาณ.....บรรทัด

☐ ในกรณีที่ท่านไม่ทราบ(ประมาณโดยวิธีของFunction Point)

(1) ภาษาที่ใช้พัฒนาซอฟต์แวร์จัดอยู่ในประเภทใด

( ) Ada	( ) ANSI/QUICK/Turbo Basic	( ) AI
( ) Assembly(Macro)	( ) Assembly	( ) Basic – Compiled
( ) Basic – Interpreted	( ) ANSI Cobol 85	( ) C
( ) C++	( ) Fortan77	( ) Forth
( ) Jovial	( ) Lisp	( ) Modula2
( ) Pascal .	( ) Prolog	( ) Report Generator
( ) Spreadsheet	( ) Shell APL	

(2) ประเภทของแฟ้มของข้อมูลที่จะใช้กับซอฟต์แวร์

(2.1) จำนวนชุดข้อมูลที่รับมาจากภายนอก (External Input: EI).....ชุด

(2.2) จำนวนชุดข้อมูลที่ส่งออกไปภายนอก (External Output: EO) .....ชุด

(2.3) จำนวนชุดข้อมูลภายในเชิงตรรกะ (Internal Logical File: ILF) .....ชุด

(2.4) จำนวนชุดข้อมูลที่ต่อประสานกับภายนอก (External Interface File: EIF) .....ชุด

(2.5) จำนวนชุดข้อมูลที่เป็นคำถามจากภายนอก (External Inquiry: EQ) .....ชุด

3.ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์(Complexity :CPLX)	ใช่	ไม่ใช่	อื่นๆ
1) ซอฟต์แวร์ที่พัฒนาต้องเขียนโปรแกรมเป็นภาษาเครื่องเพื่อควบคุมการทำงานหรือใช้งานอุปกรณ์โดยตรง ใช่หรือไม่	☐	☐	☐
2) ซอฟต์แวร์ที่จะพัฒนานี้สามารถประมวลผลแบบกระจาย(distribute processing) เช่น การประมวลผลแบบ client/sever ใช่หรือไม่	☐	☐	☐
3) ซอฟต์แวร์ที่พัฒนามีการประมวลผลแบบทันทีทันใด(real time) ใช่หรือไม่	☐	☐	☐
4) มีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน(homogeneous) คือ มีชี่ห้อและเวอร์ชันเดียวกัน หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความความต่างกัน (heterogeneous) คือ มีชี่ห้อและเวอร์ชันต่างกัน	☐	☐	☐
5) ซอฟต์แวร์ที่จะพัฒนามีการเขียนโปรแกรมลักษณะเป็นแบบการเวียนบังเกิด(recursion) ใช่หรือไม่	☐	☐	☐
6) ตัวดำเนินการมีการใช้ตารางตัดสินใจ(decision table) เพื่อช่วยควบคุมการดำเนินการ ใช่หรือไม่	☐	☐	☐
7) ซอฟต์แวร์ที่จะพัฒนามีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบcall back หรือแบบการส่งข้อความ(message passing) ใช่หรือไม่	☐	☐	☐

	ใช่	ไม่ใช่	อื่นๆ
8) ซอฟต์แวร์ที่จะพัฒนามีส่วนของโปรแกรมเป็นเงื่อนไขซ้อนเงื่อนไข เช่น IFซ้อน IF IF ซ้อนลูป While หรือ ลูป For ซ้อน IF เป็นต้น ใช่หรือไม่	☐	☐	☐
9) ซอฟต์แวร์ที่จะพัฒนามีโครงสร้างที่ซับซ้อนจึงต้องมีการใช้ Queue หรือStack เพื่อจัดลำดับการทำงานของโปรแกรมแบบที่เป็นเงื่อนไข ใช่หรือไม่	☐	☐	☐
10) ซอฟต์แวร์ที่จะพัฒนาเกี่ยวข้องกับการวิเคราะห์โครงสร้างข้อมูลเชิงตัวเลขชนิดที่มีโครงสร้างที่ไม่แน่นอน ใช่หรือไม่ เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน การวิเคราะห์สภาพอากาศ เป็นต้น	☐	☐	☐
11) ซอฟต์แวร์ที่จะพัฒนาจะต้องวิเคราะห์โครงสร้างเชิงตัวเลขได้จากชุดข้อมูล ใช่หรือไม่ เช่น การวิเคราะห์จุดสูงสุดจุดต่ำสุดของข้อมูล การหาสมการแทนการกระจายของชุดข้อมูล การหาค่าจากสมการเชิงอนุพันธ์ย่อย การหาคำตอบจากสมการเมตริกที่ไม่เอกฐาน เป็นต้น	☐	☐	☐
12) ซอฟต์แวร์ที่จะพัฒนาสามารถวิเคราะห์เชิงตัวเลขขั้นพื้นฐานได้ ใช่หรือไม่ เช่น สามารถหาอนุพันธ์ การหาคำตอบของสมการโดยวิธีการของนิวตัน(Newton's Method) การหาพื้นที่ใต้กราฟจากการอินติเกรท(Integrate) เป็นต้น	☐	☐	☐
13) ซอฟต์แวร์ที่จะพัฒนาต้องมีฟังก์ชันที่เป็นมาตราฐานทางคณิตศาสตร์และสถิติ ใช่หรือไม่ เช่นการหาส่วนเบี่ยงเบนมาตราฐาน การหาค่าZ เป็นต้น	☐	☐	☐
14) ซอฟต์แวร์ที่จะพัฒนามีการหาคำตอบจากสมการพีชคณิตหรือสมการเส้นตรง เช่น การทำบวก ลบ คูณ หาร ใช่หรือไม่	☐	☐	☐
15) ซอฟต์แวร์สามารถใช้งานอุปกรณ์โดยการเขียนโปรแกรมไปติดต่อกับการควบคุมการทำงานของฮาร์ดแวร์โดยตรง(micro-programmed) ใช่หรือไม่	☐	☐	☐
16) ซอฟต์แวร์ที่จะพัฒนามีระบบตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์ แบบเข้มงวด เช่นมีการบันทึกการทำงานของอุปกรณ์ทุกตัว หรือ แบบวิกฤตเช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน ใช่หรือไม่	☐	☐	☐
17) ซอฟต์แวร์ที่จะพัฒนามีฟังก์ชันรองรับการขัดจังหวะ(Interrupt routines) เมื่อเกิดเหตุผิดพลาดขณะที่ซอฟต์แวร์กำลังใช้งานอุปกรณ์ ใช่หรือไม่	☐	☐	☐
18) ซอฟต์แวร์ที่จะพัฒนาจะต้องสามารถทำการแปลงตำแหน่งเชิงกายภาพ(address) ซึ่งเป็นที่เก็บข้อมูลจริงๆในหน่วยความจำสำรองหรือหน่วยความจำหลัก ใช่หรือไม่	☐	☐	☐
19) ซอฟต์แวร์ที่จะพัฒนาจะต้องสามารถจัดการกับอุปกรณ์ได้ เช่น เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานของอุปกรณ์ ตรวจสอบความผิดพลาดของอุปกรณ์ เป็นต้น ใช่หรือไม่	☐	☐	☐
20) ใช้ภาษาระดับสูงในการใช้งานอุปกรณ์ เช่น การใช้คำสั่ง อ่าน(read) เขียน(write) ในภาษาPascal ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล ใช่หรือไม่	☐	☐	☐
21) ซอฟต์แวร์ที่จะพัฒนาสามารถใช้งานข้อมูลโดยภาษาธรรมชาติ(Natural Language)ได้ ใช่หรือไม่	☐	☐	☐
22) ซอฟต์แวร์ที่จะพัฒนามีการใช้งานข้อมูลที่เก็บแบบฐานข้อมูล(database) คือ มีโครงสร้างของฐานข้อมูล ประกอบด้วยแฟ้มข้อมูลมากกว่า 1 แฟ้มและมีการกำหนดความสัมพันธ์ของแฟ้มข้อมูลในฐานข้อมูล ใช่หรือไม่	☐	☐	☐

	ใช่	ไม่ใช่	อื่นๆ
23) ซอฟต์แวร์ที่จะพัฒนามีการจัดการกับฐานข้อมูลที่เป็นแบบกระจาย(distributed database) ใช่หรือไม่ เช่น การจัดการกับสินค้าคงคลังของห้างสรรพสินค้าทุกๆสาขา	☐	☐	☐
24) ซอฟต์แวร์ที่จะพัฒนาสามารถทำทริกเกอร์(trigger) ได้ เช่น การแก้ไข บรรณาธิกรการเพิ่ม การลบ ข้อมูลในแฟ้มข้อมูลได้โดยการเขียนเป็นชุดคำสั่งที่มีเงื่อนไขไว้ เมื่อเงื่อนไขถูกตรวจสอบว่าเป็นจริง ชุดคำสั่งนั้นก็จะทำงานทันที ใช่หรือไม่	☐	☐	☐
25) ซอฟต์แวร์ที่จะพัฒนาสามารถแก้ไขหรือเปลี่ยนแปลงโครงสร้าง(restructuring)ของฐานข้อมูลได้ ใช่หรือไม่ เช่น เพิ่มจำนวนฟิลด์(field)ข้อมูล การเปลี่ยนแปลงความสัมพันธ์ของข้อมูล เป็นต้น	☐	☐	☐
26) ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลได้ ใช่หรือไม่	☐	☐	☐
27) ซอฟต์แวร์ที่จะพัฒนาสามารถเก็บข้อมูลลงในแฟ้มข้อมูลเพียงแฟ้มเดียว ใช่หรือไม่	☐	☐	☐
28) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้ภาพเสมือนจริง(virtual reality) หรือ วีดิโอคอนเฟอร์เรน(video conference) ใช่หรือไม่	☐	☐	☐
29) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้โดยใช้รูปภาพ2มิติ หรือ 3มิติ ที่เคลื่อนไหวได้ ใช่หรือไม่	☐	☐	☐
30) ซอฟต์แวร์ที่จะพัฒนาสามารถใช้เสียงในการรับหรือแสดงผลได้ ใช่หรือไม่	☐	☐	☐
31) ซอฟต์แวร์ที่จะพัฒนาสามารถติดต่อกับผู้ใช้ โดยมีลักษณะการใช้ใกล้เคียงกับการใช้วินโดว์ ใช่หรือไม่ เช่น มีไอคอน(Icon) มีทูบาร์(toolbars) เป็นต้น	☐	☐	☐
32) ส่วนต่อประสานกับผู้ใช้แบบรูปภาพแบบง่ายและมีเครื่องมือช่วยสร้าง ใช่หรือไม่	☐	☐	☐
33) ท่านมีเครื่องมือที่ใช้สร้าง(generators)ส่วนที่ติดต่อประสานกับผู้ใช้นบบไม่มีความซับซ้อน ใช่หรือไม่ เช่น ใช้wizardสร้าง เป็นต้น	☐	☐	☐

4. ความต้องการที่จะนำผลิตภัณฑ์กลับมาใช้ใหม่ (Required Reusability :RUSE)	ใช่	ไม่ใช่	อื่นๆ
1) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมกลับมาใช้ใหม่ในอนาคต ใช่หรือไม่	☐	☐	☐
2) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบให้มีลักษณะเป็นฟังก์ชันหรือเป็นโมดูลที่เล็กที่สุดที่มีฟังก์ชั่นเดียว(Generic Design) เพื่อให้ซอฟต์แวร์ใดๆก็ได้สามารถนำบางส่วนกลับไปใช้ใหม่ได้ ใช่หรือไม่	☐	☐	☐
3) ซอฟต์แวร์ที่จะพัฒนานี้มีการออกแบบสำหรับการนำบางส่วนของโปรแกรมไปใช้กับระบบที่มีความใกล้เคียงกับระบบที่จะพัฒนาเท่านั้น ใช่หรือไม่	☐	☐	☐
4) ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโปรแกรม(Program)ที่มีความใกล้เคียงหรือเหมือนกับโปรแกรมที่จะพัฒนา ใช่หรือไม่ เช่น นำบางส่วนของโปรแกรมของMicrosoft Excel มาใช้กับ Microsoft Word เป็นต้น	☐	☐	☐
5) ซอฟต์แวร์จะถูกออกแบบให้สามารถนำบางส่วนของโปรแกรมไปใช้กับโครงการ(Project)ใหม่ที่มีความใกล้เคียงหรือเหมือนกับโครงการที่จะพัฒนา ใช่หรือไม่	☐	☐	☐

5. ความต้องการเอกสารที่ตรงกับวงจรชีวิต(Documentation match to life-cycle needs: DOCU)	ใช่	ไม่ใช่	อื่นๆ
1) ท่านต้องการให้มีการจัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์ ใช่หรือไม่	☐	☐	☐
2) ท่านต้องการให้มีการจัดทำเอกสารครบตามที่วงจรชีวิตได้ระบุไว้ ใช่หรือไม่	☐	☐	☐
3) ท่านต้องการให้มีการจัดทำเอกสารที่นอกเหนือจากวงจรชีวิตระบุเพื่อใช้ในองค์กรด้วย ใช่หรือไม่ เช่นเอกสารที่ระบุถึงความสำคัญของบุคคลในองค์กรที่มีต่อซอฟต์แวร์ที่จะพัฒนา หรือข้อตกลงที่ใช้เฉพาะในองค์กร เป็นต้น	☐	☐	☐
4) ท่านต้องการให้เอกสารในส่วนที่เพิ่มเติมจากข้อกำหนดของวงจรชีวิตมีความละเอียดในทุกขั้นตอน ใช่หรือไม่	☐	☐	☐

6. การจำกัดเวลาที่ใช้ในการประมวลผล (Execute Time Constraint: TIME)

เวลาที่ท่านคาดว่าซอฟต์แวร์จะใช้ในการกระทำการ(Execution Time)

☐ในกรณีที่ท่านทราบ

(1) เวลาที่ระบบให้ซอฟต์แวร์เข้ากระทำการ.....ชั่วโมงต่อวัน

(2) เวลาที่ซอฟต์แวร์เข้าไปใช้งานในซีพียูโดยประมาณ.....ชั่วโมงต่อวัน

☐ในกรณีที่ท่านไม่ทราบ

(1) ซอฟต์แวร์ที่พัฒนาเป็นซอฟต์แวร์ประเภทใด

( ) การจัดการฐานข้อมูล(Database Management) เช่น MS Access, CA-Clipper, FoxPro, Oracle, Paradox Alpha Four เป็นต้น

( ) การสื่อสาร(Communication) เช่น Comm Work, Crosstalk, SmartCom, Terminal Plus เป็นต้น

( ) การประมวลคำ (Word Processing) เช่น AmiPro3.01, MS Word, WordPerfect, WordStar เป็นต้น

( ) Spreadsheet เช่น CA-SuperCale, Lotus Improv, MS Excel, Quattro Pro เป็นต้น

( ) รูปภาพ(Graphic) เช่น CorelDRAW, FreeHand, Illustrator, DesignWorks เป็นต้น

( ) ซอฟต์แวร์ระบบ(System Software) เช่น OS/2, Win95/98/NT, UNIX, Linux เป็นต้น

( ) Integrated Software เช่น MS Work, Lotus Work, ClarisWork, Framework XE, Legato เป็นต้น

( ) แอปพลิเคชั่น(Applications) เช่น ระบบบัญชี ระบบบุคลากร เป็นต้น

( ) ระบบอื่น ระบุ(.....)

(2) ระบบของท่านสามารถทำงานแบบ

( ) ผู้ใช้สามารถใช้งานได้พร้อมกันหลายคน

( ) ผู้ใช้สามารถใช้งานได้ทีละคน

7.การจำกัดหน่วยความจำหลัก (Main Storage Constraint: STOR)

ขนาดของหน่วยความจำหลักที่ท่านคาดว่าซอฟต์แวร์จะใช้

☐ ในกรณีที่ท่านทราบ

(1) ขนาดของหน่วยความจำที่ระบบมีให้กับซอฟต์แวร์.....KB

(2) ขนาดของหน่วยความจำที่คาดว่าซอฟต์แวร์ต้องการใช้.....KB

☐ ในกรณีที่ท่านไม่ทราบ

(1) ขนาดของหน่วยความจำหลักที่ซอฟต์แวร์จองไว้ใช้งาน.....KB

(2) ขนาดของหน่วยความจำที่คาดว่าจะใช้เก็บข้อมูลที่ใช้งานกับซอฟต์แวร์.....KB

(3) ขนาดของหน่วยความจำที่ซอฟต์แวร์ใช้ร่วมกับซอฟต์แวร์อื่นๆ.....KB

(4) ขนาดของหน่วยความจำของเครื่องคอมพิวเตอร์ที่แนะนำ.....KB

8.การเปลี่ยนแปลงได้ง่ายของแพลทฟอร์ม (Platform Volatility: PVOL)	ใช่	ไม่ใช่	อื่นๆ
1) ท่านมีนโยบายที่จะทำการเปลี่ยนแพลทฟอร์มของซอฟต์แวร์ที่จะพัฒนาเป็นประจำในช่วงระยะเวลาใดเวลาหนึ่ง ใช่หรือไม่ เช่น ใช้เวลาในการเปลี่ยนเวอร์ชั่นของแพลทฟอร์มทุกๆ 6 เดือน หรือ ใช้เวลาในการเปลี่ยนชนิดของแพลทฟอร์มทุกๆ 12 เดือน เป็นต้น	☐	☐	☐
2) โดยเฉลี่ยจะใช้งานแพลทฟอร์มเป็นเวลาประมาณ.....เดือนก่อนที่จะทำการเปลี่ยนแปลงในแต่ละครั้ง			
3) ท่านจะต้องทำการปรับปรุงแพลทฟอร์มเพียงบางส่วนเป็นประจำในช่วงระยะเวลาใดเวลาหนึ่ง เพื่อให้ซอฟต์แวร์สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น หรือสามารถรองรับการใช้งานได้ ใช่หรือไม่ เช่น การขยายความสามารถในการใช้งานของผู้ให้บริการทางอินเตอร์เน็ต(Internet Service Provider) โดยการจัดการกับพื้นที่ในหน่วยความจำสำรองทุกสัปดาห์ เป็นต้น	☐	☐	☐
4) โดยเฉลี่ยใช้งานแพลทฟอร์มเป็นเวลาประมาณ.....สัปดาห์ก่อนที่จะทำการเปลี่ยนแปลงในแต่ละครั้ง			

9.คำถามความสามารถในการวิเคราะห์ (Analyst Capability: ACAP)

1) จำนวนนักวิเคราะห์ระบบในทีมที่พัฒนาซอฟต์แวร์.....คน

2) ท่านคิดว่านักวิเคราะห์ระบบแต่ละท่านในทีมมีความสามารถอยู่ในระดับใดบ้าง

(มีความสามารถมากที่สุด = 5, มีความสามารถมาก = 4, มีความสามารถปานกลาง = 3, มีความสามารถน้อย = 2, มีความสามารถน้อยที่สุด = 1)

นักวิเคราะห์ระบบคนที่ 1 มีความสามารถอยู่ในระดับ.....

นักวิเคราะห์ระบบคนที่ 2 มีความสามารถอยู่ในระดับ.....

นักวิเคราะห์ระบบคนที่ 3 มีความสามารถอยู่ในระดับ.....

นักวิเคราะห์ระบบคนที่ 4 มีความสามารถอยู่ในระดับ.....

นักวิเคราะห์ระบบคนที่ 5 มีความสามารถอยู่ในระดับ.....ฯลฯ

10. ความสามารถของโปรแกรมเมอร์ (Programmer Capability: PCAP)

1) จำนวนโปรแกรมเมอร์ในทีมที่พัฒนาซอฟต์แวร์.....คน

2) ท่านคิดว่าโปรแกรมเมอร์แต่ละท่านในทีมมีความสามารถอยู่ในระดับใดบ้าง

(มีความสามารถมากที่สุด = 5, มีความสามารถมาก = 4, มีความสามารถปานกลาง = 3,มีความสามารถน้อย = 2,มีความสามารถน้อยที่สุด =1)

โปรแกรมเมอร์คนที่ 1 มีความสามารถอยู่ในระดับ.....

โปรแกรมเมอร์คนที่ 2 มีความสามารถอยู่ในระดับ.....

โปรแกรมเมอร์คนที่ 3 มีความสามารถอยู่ในระดับ.....

โปรแกรมเมอร์คนที่ 4 มีความสามารถอยู่ในระดับ.....

โปรแกรมเมอร์คนที่ 5 มีความสามารถอยู่ในระดับ.....ฯลฯ

11. ประสบการณ์การพัฒนาแอปพลิเคชั่น (Application Experience: AEXP)

1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน

2) ประสบการณ์ในการพัฒนาซอฟต์แวร์ชนิดเดียวกับซอฟต์แวร์ที่จะพัฒนาของแต่ละท่าน

บุคลากรคนที่ 1 มีประสบการณ์.....เดือน

บุคลากรคนที่ 2 มีประสบการณ์.....เดือน

บุคลากรคนที่ 3 มีประสบการณ์.....เดือน

บุคลากรคนที่ 4 มีประสบการณ์.....เดือน

บุคลากรคนที่ 5 มีประสบการณ์.....เดือน

บุคลากรคนที่ 6 มีประสบการณ์.....เดือน

บุคลากรคนที่ 7 มีประสบการณ์.....เดือน

บุคลากรคนที่ 8 มีประสบการณ์.....เดือน

บุคลากรคนที่ 9 มีประสบการณ์.....เดือน

ฯลฯ

12.ประสบการณ์การใช้แพลทฟอร์ม(Platform Experience: PEXP)

1) จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน

2) ประสบการณ์ในการใช้แพลทฟอร์มชนิดเดียวกับแพลทฟอร์มที่จะใช้พัฒนาซอฟต์แวร์ของแต่ละท่าน

บุคลากรคนที่ 1 มีประสบการณ์.....เดือน

บุคลากรคนที่ 2 มีประสบการณ์.....เดือน

บุคลากรคนที่ 3 มีประสบการณ์.....เดือน

บุคลากรคนที่ 4 มีประสบการณ์.....เดือน

บุคลากรคนที่ 5 มีประสบการณ์.....เดือน

ฯลฯ

13.ประสบการณ์การใช้ภาษาโปรแกรมและเครื่องมือ(Language and Tool Experience: LTEX)

1)จำนวนของบุคลากรในทีมพัฒนา(ประกอบด้วย ผู้จัดการโครงการและผู้ช่วย ผู้บริหารโครงการและผู้ช่วย นักวิเคราะห์ระบบ นักออกแบบระบบ วิศวกรระบบ โปรแกรมเมอร์).....คน

2) ประสบการณ์ในการใช้ภาษาโปรแกรม(Language Programming)ชนิดเดียวกับที่จะใช้พัฒนาซอฟต์แวร์ของโปรแกรมเมอร์แต่ละท่าน

บุคลากรคนที่ 1 มีประสบการณ์.....เดือน
บุคลากรคนที่ 2 มีประสบการณ์.....เดือน
บุคลากรคนที่ 3 มีประสบการณ์.....เดือน
บุคลากรคนที่ 4 มีประสบการณ์.....เดือน
บุคลากรคนที่ 5 มีประสบการณ์.....เดือน
บุคลากรคนที่ 6 มีประสบการณ์.....เดือน
บุคลากรคนที่ 7 มีประสบการณ์.....เดือน
บุคลากรคนที่ 8 มีประสบการณ์.....เดือน
บุคลากรคนที่ 9 มีประสบการณ์.....เดือน
ฯลฯ

3) ประสบการณ์ในการใช้เครื่องมือ(Tool)ชนิดเดียวกับที่จะใช้พัฒนาซอฟต์แวร์ของบุคลากรแต่ละท่าน

บุคลากรคนที่ 1 มีประสบการณ์.....เดือน
บุคลากรคนที่ 2 มีประสบการณ์.....เดือน
บุคลากรคนที่ 3 มีประสบการณ์.....เดือน
บุคลากรคนที่ 4 มีประสบการณ์.....เดือน
บุคลากรคนที่ 5 มีประสบการณ์.....เดือน
บุคลากรคนที่ 6 มีประสบการณ์.....เดือน
บุคลากรคนที่ 7 มีประสบการณ์.....เดือน
บุคลากรคนที่ 8 มีประสบการณ์.....เดือน
บุคลากรคนที่ 9 มีประสบการณ์.....เดือน
ฯลฯ

14. ความต่อเนื่องของบุคลากร(Personnel Continuity: PCON)

1) จำนวนบุคลากรที่พัฒนาเมื่อเริ่มการพัฒนาซอฟต์แวร์.....คน
2) ท่านมีนโยบายที่จะรับบุคลากรเข้าทีมพัฒนาเพิ่มอีก.....คน
3) ท่านมีนโยบายให้บุคลากรลาออกได้.....คน
4) บุคลากรที่รับเข้ามาใหม่จะทำงานแทนบุคลากรคนเก่าที่ลาออกไป...คน
5) ซอฟต์แวร์มีกำหนดเวลาของการพัฒนาประมาณ.....ปี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

15. การใช้เครื่องมือซอฟต์แวร์ (Use of software tool: TOOL)	ใช่	ไม่ใช่	อื่นๆ
1) เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นภาษาโปรแกรม เช่น ภาษา C,Visual Basic, Pascal, Java เป็นต้น ใช่หรือไม่	☐	☐	☐
2) เครื่องมือที่ใช้พัฒนาซอฟต์แวร์มีลักษณะเป็นเครื่องมือเคส (Computer Aided Software Engineering :CASE) ใช่หรือไม่	☐	☐	☐
3) เครื่องมือที่ใช้นอกจากจะเป็นแบบเครื่องมือเคสที่ช่วยในการพัฒนาซอฟต์แวร์ได้ทุกขั้นตอนของวงจรชีวิต(SDLC)แล้วยังสามารถนำไปใช้รวม(Integrate)กับซอฟต์แวร์อื่นๆได้ ใช่หรือไม่	☐	☐	☐
4) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ช่วยในหลายขั้นตอนของวงจรชีวิต(SDLC) เช่น ช่วยในการเก็บรวบรวมข้อมูล(Required Phase)และนำข้อมูลที่ได้มาวิเคราะห์และออกแบบการพัฒนาซอฟต์แวร์(Design Phase) หรือ ช่วยออกแบบซอฟต์แวร์(Design Phase) แล้วนำไปสร้างโปรแกรม(Code Generator) เป็นต้น ใช่หรือไม่	☐	☐	☐
5) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำส่วนเสริมหน้า (Frontend Computer Aided Software Engineering : Frontend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงแรกๆ ใช่หรือไม่ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆเกี่ยวกับซอฟต์แวร์ที่จะพัฒนา(Required Phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning Phase) หรือ ช่วยในขั้นตอนการออกแบบ(Design Phase)	☐	☐	☐
6) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำส่วนเสริมหลัง(Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิต(SDLC)ในช่วงหลัง ใช่หรือไม่ เช่น ช่วยในขั้นตอนของการทำให้เกิดผล(Implementation Phase) ช่วยในขั้นตอนของการรวม (Integration Phase) หรือช่วยในขั้นตอนของการบำรุงรักษา(Maintenance Phase)	☐	☐	☐
7) เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสแบบการรวม(Integrated CASE) ซึ่งสามารถทำการรวม(Integrated) แบบกระบวนการ(Process) วิธีการ(Method) หรือการใช้ใหม่(Reuse) ใช่หรือไม่	☐	☐	☐

16. การพัฒนาในหลายสถานที่ (Multisite Development: SITE)	ใช่	ไม่ใช่	อื่นๆ
1) ซอฟต์แวร์จะถูกพัฒนาภายในประเทศเท่านั้น ใช่หรือไม่	☐	☐	☐
2) ซอฟต์แวร์จะถูกพัฒนาภายในจังหวัดเดียวกัน ใช่หรือไม่	☐	☐	☐
3) ซอฟต์แวร์จะถูกพัฒนาภายในบริษัทเดียวหรือหลายบริษัท	☐	☐	☐
4) ซอฟต์แวร์จะถูกพัฒนาในกรุงเทพฯหรือจังหวัดสำคัญๆ ใช่หรือไม่	☐	☐	☐
5) ซอฟต์แวร์จะถูกพัฒนาภายในอาคารเดียวกันหรือหลายอาคารที่อยู่ในบริเวณใกล้เคียงกัน ใช่หรือไม่	☐	☐	☐
6) ซอฟต์แวร์จะถูกพัฒนาภายในบริเวณเดียวกัน ห้องเดียวกัน หรือชั้นเดียวกัน ใช่หรือไม่	☐	☐	☐
7) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ คือ มัลติมีเดียที่สามารถได้ตอบได้ ใช่หรือไม่	☐	☐	☐

	ใช่	ไม่ใช่	อื่นๆ
8) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดิทัศน์ ใช่หรือไม่	☐	☐	☐
9) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN) เป็นต้น ใช่หรือไม่	☐	☐	☐
10) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้สัญญาณอิเล็กทรอนิกส์ช่องสัญญาณแคบ ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ(private network) เครือข่ายท้องถิ่น(LAN) เป็นต้น ใช่หรือไม่	☐	☐	☐
11) เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้โทรศัพท์ส่วนตัว หรือแฟกส่วนตัว ใช่หรือไม่	☐	☐	☐

17. กำหนดเวลาของการพัฒนาที่ต้องการ (Required Development Schedule: SCED)

1) หลังจากที่ท่านได้เก็บรวบรวมความต้องการ(Requirement) ต่างๆจากผู้ใช้แล้ว ท่านคาดว่าจะใช้เวลาในการพัฒนาซอฟต์แวร์ประมาณ .....เดือน

2)เวลาที่ท่านบวกเผื่อไว้(กรณีใดๆ)ประมาณ.....เดือน

########## จบ ##########

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

โครงสร้างฐานข้อมูล

1. ชื่อตาราง : AcapDetail

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับของแฟ้มข้อมูล
2	ProjectName	อักขระ	100	ชื่อโครงการ
3	Code	ตัวเลข	2	รหัสโครงการ
4	Date	วันที่	8	วันที่ที่ประเมินโครงการ
5	Score	ตัวเลข	1	ระดับความสามารถของนักวิเคราะห์ระบบ

เขตข้อมูลหลัก(Primary Key) : NO

2 ชื่อตาราง : AcapTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	TotalSA	ตัวเลข	2	จำนวนนักวิเคราะห์ระบบ

เขตข้อมูลหลัก : ProjectName, Date, Code

3. ชื่อตาราง : AexpDetail

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับของแฟ้มข้อมูล
2	ProjectName	อักขระ	100	ชื่อโครงการ
3	Code	ตัวเลข	2	รหัสโครงการ
4	Date	วันที่	8	วันที่ที่ประเมินโครงการ
5	EXP	ตัวเลข	4	ประสบการณ์การพัฒนาซอฟต์แวร์เป็นระยะเวลา(เดือน)

เขตข้อมูลหลัก : ProjectName, Date, Code

4. ชื่อตาราง : AexpTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	TotalPerson	ตัวเลข	2	จำนวนบุคลากรทั้งหมด

เขตข้อมูลหลัก : ProjectName, Date, Code

5. ชื่อตาราง : CplxTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Q	อักขระ	6	คำถาม
5	Ans	อักขระ	6	คำตอบ: ใช่,ไม่ใช่

เขตข้อมูลหลัก : ProjectName, Date, Code

6. ชื่อตาราง : DataFile

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับของแฟ้มข้อมูล
2	ProjectName	อักขระ	100	ชื่อโครงการ
3	Code	ตัวเลข	2	รหัสโครงการ
4	Date	วันที่	8	วันที่ที่ประเมินโครงการ
5	Rec	ตัวเลข	10	จำนวนรายการ
6	RecLenght	ตัวเลข	10	ขนาดของรายการ(ไบต์)

เขตข้อมูลหลัก : ProjectName, Date, Code

7. ชื่อตาราง : DataTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	StorSize	ตัวเลข	10	ขนาดของหน่วยความจำสำรอง(กิโลไบต์)

เขตข้อมูลหลัก : ProjectName, Date, Code

9. ชื่อตาราง : DesignModelTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Date	วันที่	8	วันที่ที่ประเมินโครงการ
3	PERS	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPERS
4	PREX	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPREX
5	FCIL	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทFCIL
6	SCED	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทSCED
7	PDIF	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPDIF
8	RCPX	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทRCPX
9	RUSE	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทRUSE
10	PREC	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPREX
11	FLEX	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทFCIL
12	RESL	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทSCED
13	TEAM	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPDIF
14	PMAT	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทRCPX
15	Author	อักขระ	50	ชื่อผู้ทำการประเมิน
16	PersonMonth	ตัวเลข	10	ค่าใช้จ่ายในการพัฒนา(คน)
17	TimeDev	ตัวเลข	10	ระยะเวลาที่ใช้พัฒนา(เดือน)
18	TotalCost	ตัวเลข	10	ค่าใช้จ่ายมีหน่วยเป็นบาท
19	Size	ตัวเลข	10	ขนาดของซอฟต์แวร์มีหน่วยเป็นบรรทัด

เขตข้อมูลหลัก : ProjectName, Date, Code

10. ชื่อตาราง : DocuTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Q	อักขระ	6	คำถาม
5	Ans	อักขระ	6	คำตอบ : ใช่,ไม่ใช่

เขตข้อมูลหลัก : ProjectName, Date, Code

11. ชื่อตาราง : LangTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับ
2	Language	อักขระ	20	ภาษา : Ada AI Shell APL Basic Assembly Macro Assembly Quick-TurboBasic Compiled Basic Interpreted Basic C C++ ANSI Cobol 85 Forth Fortran77 Forth Generation High-Level Jovial Lisp Modula2 Object-Oriented Pascal Procedural Language Program Generator Prolog Query Language Report Generator Spreadsheet
3	SLOC/UFP	ตัวเลข	3	จำนวนบรรทัดของโปรแกรมต่อหนึ่งฟังก์ชันพอยท์ของแต่ละภาษาโปรแกรม

เขตข้อมูลหลัก : ProjectName, Date, Code

12. ชื่อตาราง : LineOfCode

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	ตัวเลข	2	ชื่อโครงการ
2	Date	อักขระ	50	วันที่ทำการประเมิน
3	Sloc	ตัวเลข	3	ขนาดของซอฟต์แวร์(บรรทัด)
4	SBrak	ตัวเลข	5	เปอร์เซ็นต์ของการเปลี่ยนแปลง
5	IntSloc	ตัวเลข	5	ขนาดของซอฟต์แวร์(บรรทัด)เริ่มต้น
6	DM	ตัวเลข	5	เปอร์เซ็นต์การเปลี่ยนแปลงของการออกแบบ
7	CM	ตัวเลข	5	เปอร์เซ็นต์การเปลี่ยนแปลงของเขียนโปรแกรม
8	IM	ตัวเลข	5	เปอร์เซ็นต์การเปลี่ยนแปลงของการทดสอบ
9	SU	ตัวเลข	5	ระดับความเข้าใจโปรแกรมของผู้พัฒนา
10	AA	ตัวเลข	5	ระดับสิ่งที่นำมาใช้ได้
11	UNFM	ตัวเลข	5	ความคุ้นเคยของโปรแกรมเมอร์ที่มีต่อโปรแกรม
12	AT	ตัวเลข	5	เปอร์เซ็นต์ของการแปลงอัตโนมัติ
13	ATPROD	ตัวเลข	5	ตัวคูณของการเปลี่ยนแปลงโดยอัตโนมัติ
14	AAF	ตัวเลข	5	เปอร์เซ็นต์โดยรวมของการปรับปรุง
15	ASloc	ตัวเลข	5	ขนาดของซอฟต์แวร์(บรรทัด)
16	ABrak	ตัวเลข	5	เปอร์เซ็นต์ของการเปลี่ยนแปลง
17	Lang	ตัวเลข	5	ภาษาโปรแกรม
18	EIL	ตัวเลข	5	จำนวนชุดข้อมูลที่รับจากภายนอกระดับต่ำ
19	EIN	ตัวเลข	5	จำนวนชุดข้อมูลที่รับจากภายนอกระดับปานกลาง
20	EIH	ตัวเลข	5	จำนวนชุดข้อมูลที่รับจากภายนอกระดับสูง
21	EI	ตัวเลข	5	จำนวนชุดข้อมูลที่รับจากภายนอกทั้งหมด
22	EOL	ตัวเลข	5	จำนวนชุดข้อมูลที่ส่งออกไปภายนอกระดับต่ำ
23	EON	ตัวเลข	5	จำนวนชุดข้อมูลที่ส่งออกไปภายนอกระดับปานกลาง
24	EOH	ตัวเลข	5	จำนวนชุดข้อมูลที่ส่งออกไปภายนอกระดับสูง
25	EO	ตัวเลข	5	จำนวนชุดข้อมูลที่ส่งออกไปภายนอกทั้งหมด
26	ILFL	ตัวเลข	5	จำนวนชุดข้อมูลภายในเชิงตรรกะระดับต่ำ
27	ILFN	ตัวเลข	5	จำนวนชุดข้อมูลภายในเชิงตรรกะระดับปานกลาง
28	ILFH	ตัวเลข	5	จำนวนชุดข้อมูลภายในเชิงตรรกะระดับสูง
29	ILF	ตัวเลข	5	จำนวนชุดข้อมูลภายในเชิงตรรกะทั้งหมด
30	EIFL	ตัวเลข	5	จำนวนชุดข้อมูลที่ต่อประสานกับภายนอกระดับต่ำ
31	EIFN	ตัวเลข	5	จำนวนชุดข้อมูลที่ต่อประสานกับภายนอกระดับปานกลาง
32	EIFH	ตัวเลข	5	จำนวนชุดข้อมูลที่ต่อประสานกับภายนอกระดับสูง

12. ชื่อตาราง : LineOfCode(ต่อ)

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
33	EIF	ตัวเลข	5	จำนวนชุดข้อมูลที่ต่อประสานกับภายนอกทั้งหมด
34	EQL	ตัวเลข	5	จำนวนชุดข้อมูลที่เป็นคำถามจากภายนอกระดับต่ำ
35	EQN	ตัวเลข	5	จำนวนชุดข้อมูลที่เป็นคำถามจากภายนอกระดับปานกลาง
36	EQH	ตัวเลข	5	จำนวนชุดข้อมูลที่เป็นคำถามจากภายนอกระดับสูง
37	EQ	ตัวเลข	5	จำนวนชุดข้อมูลที่เป็นคำถามจากภายนอกทั้งหมด
38	TotalFP	ตัวเลข	5	จำนวนฟังก์ชันพอยท์
39	FSloc	ตัวเลข	5	ขนาดของซอฟต์แวร์(บรรทัด)
40	FBrak	ตัวเลข	5	เปอร์เซ็นต์ของการเปลี่ยนแปลง

เขตข้อมูลหลัก : NO

13. ชื่อตาราง : LtexDetail

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับของแฟ้มข้อมูล
2	ProjectName	อักขระ	100	ชื่อโครงการ
3	Code	ตัวเลข	2	รหัสโครงการ
4	Date	วันที่	8	วันที่ที่ประเมินโครงการ
5	EXPL	ตัวเลข	4	ประสบการณ์การการใช้ภาษา(เดือน)
5	EXPT	ตัวเลข	4	ประสบการณ์การการใช้เครื่องมือ(เดือน)

เขตข้อมูลหลัก : ProjectName, Date, Code

14. ชื่อตาราง : LtexTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	TotalPerson	ตัวเลข	2	จำนวนบุคลากรทั้งหมด

เขตข้อมูลหลัก : ProjectName, Date, Code

15. ชื่อตาราง : MainTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	RELY	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทRELY
5	DATA	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทDATA
6	CPLX	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทCPLX
7	RUSE	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทRUSE
8	DOCU	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทDOCU
9	TIME	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทTIME
10	STOR	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทSTOR
11	PVOL	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPVOL
12	ACAP	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทACAP
13	PCAP	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPCAP
14	AEXP	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทAEXP
15	PEXP	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPEXP
16	LTEX	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทLTEX
17	PCON	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทPCON
18	TOOL	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทTOOL
19	SITE	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทSITE
20	SCED	อักขระ	10	ตัวขับค่าใช้จ่ายประเภทSCED
21	PREC	อักขระ	10	ปัจจัยประเภท PREC
22	FLEX	อักขระ	10	ปัจจัยประเภท FLEX
23	RESL	อักขระ	10	ปัจจัยประเภท RESL
24	TEAM	อักขระ	10	ปัจจัยประเภท TEAM
25	PMAT	อักขระ	10	ปัจจัยประเภท PMAT
26	Author	อักขระ	50	ชื่อผู้ทำการประเมิน
27	PersonMonth	ตัวเลข	10	ค่าใช้จ่ายในการพัฒนา(คน)
28	TimeDev	ตัวเลข	10	ระยะเวลาที่ใช้พัฒนา(เดือน)
29	TotalCost	ตัวเลข	10	ค่าใช้จ่ายมีหน่วยเป็นบาท
30	Size	ตัวเลข	10	ขนาดของซอฟต์แวร์มีหน่วยเป็นบรรทัด

เขตข้อมูลหลัก : ProjectName, Date, Code

16. ชื่อตาราง : MultiplierTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับ
2	CostDriver	อักขระ	4	ชื่อย่อตัวขับค่าใช้จ่าย
3	Level	อักขระ	10	ระดับ : ต่ำที่สุด, ต่ำ, ปานกลาง, สูง สูงมาก,สูงที่สุด
4	Multiplier	ตัวเลข	5	ค่าความพยายาม

เขตข้อมูลหลัก : NO

17. ชื่อตาราง : ObjectPointTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Date	วันที่	8	วันที่ที่ประเมินโครงการ
3	Screen1	อักขระ	10	จำนวนหน้าจอที่ใช้ตารางน้อยกว่า 4 ตาราง
4	Screen2	อักขระ	10	จำนวนหน้าจอที่ใช้ตารางน้อยกว่า 8 ตาราง
5	Screen3	อักขระ	10	จำนวนหน้าจอที่ใช้ตารางมากกว่า 8 ตาราง
6	Report1	อักขระ	10	จำนวนหน้าจอที่ใช้ตารางน้อยกว่า 4 ตาราง
7	Report2	อักขระ	10	จำนวนหน้าจอที่ใช้ตารางน้อยกว่า 8 ตาราง
8	Report3	อักขระ	10	จำนวนหน้าจอที่ใช้ตารางมากกว่า 8 ตาราง
9	Reuse	ตัวเลข	2	เปอร์เซ็นต์ของการนำโปรแกรมเก่ามาใช้
10	PROD	ตัวเลข	4	อัตราการผลิต
11	PersonMonth	ตัวเลข	10	ค่าใช้จ่ายในการพัฒนา(คน)

เขตข้อมูลหลัก : ProjectName, Date

18. ชื่อตาราง : PcapDetail

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับของแฟ้มข้อมูล
2	ProjectName	อักขระ	100	ชื่อโครงการ
3	Code	ตัวเลข	2	รหัสโครงการ
4	Date	วันที่	8	วันที่ที่ประเมินโครงการ
5	Score	ตัวเลข	1	ระดับความสามารถของโปรแกรมเมอร์

เขตข้อมูลหลัก : ProjectName, Date, Code

19. ชื่อตาราง : PcapTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	TotalProg	ตัวเลข	2	จำนวนโปรแกรมเมอร์

เขตข้อมูลหลัก : ProjectName, Date, Code

20. ชื่อตาราง : PconTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	TotalPerson	ตัวเลข	2	จำนวนบุคลากรทั้งหมด
5	IN	ตัวเลข	2	จำนวนบุคลากรที่รับเข้ามาใหม่
6	OUT	ตัวเลข	2	จำนวนบุคลากรที่ลาออก
7	Replace	ตัวเลข	2	จำนวนบุคลากรที่รับเข้ามาแทนที่บุคลากรเดิมที่ลาออก
8	DevTime	ตัวเลข	2	ระยะเวลาที่ใช้พัฒนาซอฟต์แวร์(เดือน)

เขตข้อมูลหลัก : ProjectName, Date, Code

21. ชื่อตาราง : PexpDetail

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	NO	ตัวเลข	2	ลำดับของแฟ้มข้อมูล
2	ProjectName	อักขระ	100	ชื่อโครงการ
3	Code	ตัวเลข	2	รหัสโครงการ
4	Date	วันที่	8	วันที่ที่ประเมินโครงการ
5	EXP	ตัวเลข	4	ประสบการณ์การพัฒนาแพลทฟอร์มเป็นระยะเวลา(เดือน)

เขตข้อมูลหลัก : ProjectName, Date, Code

22. ชื่อตาราง : PexpTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	TotalPerson	ตัวเลข	2	จำนวนบุคลากรทั้งหมด

เขตข้อมูลหลัก : ProjectName, Date, Code

23. ชื่อตาราง : PvolTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Change	ตัวเลข	4	ระยะเวลาที่จะเปลี่ยนแพลทฟอร์ม
5	Improve	ตัวเลข	4	ระยะเวลาที่จะปรับปรุงแพลทฟอร์ม

เขตข้อมูลหลัก : ProjectName, Date, Code

24.ชื่อตารางข้อมูล : RelyTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Q	อักขระ	6	คำถาม
5	Ans	อักขระ	6	คำตอบ : ใช่,ไม่ใช่

เขตข้อมูลหลัก : ProjectName, Date, Code

25. ชื่อตาราง : RuseTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Q	อักขระ	6	คำถาม
5	Ans	อักขระ	6	คำตอบ : ใช่,ไม่ใช่

เขตข้อมูลหลัก : ProjectName, Date, Code

26. ชื่อตาราง : ScedTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Q1	ตัวเลข	6	ระยะเวลาที่คาดว่าจะใช้ในการพัฒนาซอฟต์แวร์(เดือน)
5	Q2	ตัวเลข	2	ระยะเวลาที่เผื่อ(เดือน)
6	Q3	ตัวเลข	2	ระยะเวลาที่สามารถเร่งให้เร็วขึ้น(เดือน)

เขตข้อมูลหลัก : ProjectName, Date, Code

27. ชื่อตาราง : SiteTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Q	อักขระ	6	คำถาม
5	Ans	อักขระ	6	คำตอบ : ใช่,ไม่ใช่

เขตข้อมูลหลัก : ProjectName, Date, Code

28. ชื่อตาราง : SoftwareType

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	SFType	อักขระ	100	ชนิดของซอฟต์แวร์
2	Use	วันที่	8	จำนวนผู้ที่ใช้งาน
3	Level	อักขระ	10	ระดับ

เขตข้อมูลหลัก : SFType,Use

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

29 ชื่อตาราง : StorTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	MemSize	ตัวเลข	4	ขนาดของหน่วยความจำหลักที่ระบบให้ซอฟต์แวร์
5	ExecSize	ตัวเลข	4	ขนาดของหน่วยความจำหลักที่ซอฟต์แวร์ใช้
6	MemReserve	ตัวเลข	4	ขนาดของหน่วยความจำหลักที่ซอฟต์แวร์จองไว้ใช้
7	DataSize	ตัวเลข	4	ขนาดของหน่วยความจำหลักที่ซอฟต์แวร์ใช้เก็บข้อมูล
8	Recommend	ตัวเลข	4	ขนาดของหน่วยความจำหลักที่แนะนำให้ใช้กับซอฟต์แวร์

เขตข้อมูลหลัก : ProjectName, Date, Code

30. ชื่อตาราง : TimeTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	AvTime	ตัวเลข	4	ระยะเวลาที่ระบบให้ซอฟต์แวร์
5	ExecTime	ตัวเลข	4	ระยะเวลาที่ซอฟต์แวร์เข้ากระทำการ
6	SoftwareType	อักขระ	20	ประเภทของซอฟต์แวร์
7	Use	อักขระ	50	ลักษณะการใช้งาน

เขตข้อมูลหลัก : ProjectName, Date, Code

31. ชื่อตาราง :ToolTable

ลำดับ	เขตข้อมูล	ประเภท	ความยาว	คำอธิบาย
1	ProjectName	อักขระ	100	ชื่อโครงการ
2	Code	ตัวเลข	2	รหัสโครงการ
3	Date	วันที่	8	วันที่ที่ประเมินโครงการ
4	Q	อักขระ	6	คำถาม
5	Ans	อักขระ	6	คำตอบ : ใช่,ไม่ใช่

เขตข้อมูลหลัก : ProjectName, Date, Code

ประวัติผู้เขียน

นางสาวศรินทร์ วัชรบุศราคำ เกิดวันที่ 19 ธันวาคม พศ. 2515 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตรบัณฑิต สาขาสถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2537 หลังจากจบการศึกษาได้เข้าทำงานที่ธนาคารทหารไทย ในตำแหน่งโปรแกรมเมอร์ ก่อนลาออกมาเข้าศึกษาต่อในสาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พศ. 2539

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย