บทที่ 2

เหตุผล และทฤษฎี

คำนำ

จากการศึกษาถึงแบบจำลองการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์เบื้องต้นพบว่า มีปัจจัยหลายๆประการที่มีผลต่อการผลิตผลิตภัณฑ์ซอฟต์แวร์ เช่นจำนวนบรรทัดคำสั่ง ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์ รวมถึงสภาพแวดล้อมที่เกี่ยวข้องในการพัฒนาซอฟต์แวร์ด้วย

ในการวิจัยนี้ผู้วิจัยเลือกใช้แนวความคิดตามแบบจำลองโคโคโม เพราะเป็นแบบจำลองที่นิยมใช้กันอย่างกว้างขวาง มีการรวบรวมข้อมูลของโครงงานที่พัฒนาเสร็จสมบูรณ์ 63 โครงงาน มาวิเคราะห์เป็นสมการในการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ ตามสภาพแวดล้อมในการพัฒนาซอฟต์แวร์ ทั้งนำเอาจำนวนบรรทัดคำสั่ง และตัวแปรที่วิจัยพบว่ามีผลต่อการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ 15 ลักษณะมาประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ด้วย อีกทั้งสามารถประมาณการโปรแกรมประยุกต์ได้ทุกชนิด ขนาดของคำสั่งที่ใช้ในการประมาณการตั้งแต่ 2000 ถึง 512000 บรรทัดคำสั่ง การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แนวความคิดตามแบบจำลองโคโคโมนี้สามารถประมาณการค่าใช้จ่ายโดยใช้เทคนิค บ็อตทอม อัพ คือประมาณการค่าใช้จ่ายในแต่ละองค์ประกอบแล้วรวมเป็นค่าใช้จ่ายทางด้านซอฟต์แวร์ทั้งโครงงาน ซึ่งช่วยผู้จัดการโครงงานสามารถควบคุม และวัดความก้าวหน้า ในการผลิตภัณฑ์ซอฟต์แวร์ได้ละเอียดยิ่งขึ้น

ภาคการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโม

การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโม คือ การประมาณการค่าใช้จ่ายกำลังคน(manpower) และเวลาที่ใช้ในการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ โคโคโม ย่อมาจาก COnstructive COst MOdel(COCOMO) ประกอบด้วย 3 ภาค คือภาคเบสิก(Basic Version) ภาคอินเตอร์มิเดท(Intermediate Version) และภาคดีเทล(Detail Version)

1. ภาคเบสิก

เป็นการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ที่ดี กรณีต้องการประมาณการค่าใช้จ่ายที่ต้องการเร็ว ค่าที่ประมาณการได้เป็นค่าโดยคร่าวๆ ใช้ประโยชน์ในระยะเริ่มต้นในการพัฒนาซอฟต์แวร์ ซึ่งการประมาณการค่าใช้จ่ายภาคเบสิกนี้อาจเกิดความคลาดเคลื่อนสูง เนื่องจากใช้ข้อมูลเพียงจำนวนบรรทัดคำสั่งเท่านั้น เป็นข้อมูลในการประมาณการ โดยไม่ได้คำนึงถึงสภาพแวดล้อมต่างๆที่มีผลต่อการพัฒนาซอฟต์แวร์เลย

2. ภาคอินเตอร์มีเดท

เป็นแบบจำลองซึ่งให้ความถูกต้องในการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์สูงกว่าภาคเบสิก เพราะได้นำตัวแปรที่มีผลต่อการพัฒนาซอฟต์แวร์ซึ่งเรียกว่า ตัวขับค่าใช้จ่าย 15 ลักษณะ เป็นข้อมูลในการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ด้วย

ตัวขับค่าใช้จ่ายนี้แบ่งออกเป็น 4 กลุ่มใหญ่ๆ ดังนี้

2.1 กลุ่มผลิตภัณฑ์ ประกอบด้วย

2.1.1 ความต้องการความเชื่อถือได้ของผลิตภัณฑ์ซอฟต์แวร์
Required Software Reliability (RELY)

2.1.2 ขนาดของฐานข้อมูลที่ใช้ของผลิตภัณฑ์ซอฟต์แวร์
Data Base Size (DATA)

2.1.3 ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์
Complexity (CPLX)

2.2 กลุ่มคอมพิวเตอร์ ประกอบด้วย

2.2.1 การบังคับเวลากระทำการ

Execution Time Constraint (TIME)

2.2.2 การบังคับหน่วยความจำหลัก

Main Storage Constraint (STOR)

2.2.3 การลบเลือนได้ของเครื่องเสมือน

Virtual Machine Volatility (VIRT)

2.2.4 เวลาครบวงงาน

Computer Turnaround Time (TURN)

2.3 กลุ่มบุคลากร

2.3.1 ความสามารถในการวิเคราะห์

Analyst Capability (ACAP)

2.3.2 ประสบการณ์การประยุกต์

Application Experience (AEXP)

2.3.3 ความสามารถของโปรแกรมเมอร์

Programmer Capability (PCAP)

2.3.4 ประสบการณ์เครื่องเสมือน

Virtual Machine Experience (VEXP)

2.3.4 ประสบการณ์โปรแกรมภาษา

Programming Language Experience (LEXP)

2.4 กลุ่มโครงงาน

2.4.1 การปฏิบัติการเขียนโปรแกรมสมัยใหม่

Modern Programming Practices (MODP)

2.4.2 การใช้เครื่องมือซอฟต์แวร์

USE of Software Tools (TOOL)

2.4.3 การบังคับการจัดกำหนดการ

Schedule Constraint (SCED)

การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ภาคอินเตอร์มีเดียทนี้ ประมาณการค่าใช้จ่ายความพยายามในการพัฒนาซอฟต์แวร์และเวลาที่ใช้ในการพัฒนาซอฟต์แวร์ โดยประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ทั้งระบบ และประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยแบ่งออกเป็นระยะตามวัฏจักรในการพัฒนาซอฟต์แวร์อันประกอบด้วย การวางแผนและศึกษาความต้องการ การออกแบบ การเขียนคำสั่ง และการรวมและทดสอบ ซึ่งช่วยให้ผู้จัดการโครงงานสามารถวัดความก้าวหน้าในการพัฒนาซอฟต์แวร์ได้เป็นระยะๆ นอกจากนี้ยังประมาณการจำนวนคนที่ใช้ในแต่ละระยะตามกิจกรรม (activity)ต่างๆในการพัฒนาซอฟต์แวร์ และประมาณการผลผลิต(productivity)ในการพัฒนาซอฟต์แวร์ของโครงงานอีกด้วย

3. ภาคดีเทล

เป็นการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ ซึ่งขยายขีดความสามารถของภาคอินเตอร์มีเดีท เนื่องจากภาคอินเตอร์มีเดียทประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ทั้งระบบโดยรวม ส่วนภาคดีเทลนั้นสามารถประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ได้ในระดับโปรแกรมย่อย รวมทั้งประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ระดับมอดุลในแต่ละโปรแกรมย่อยได้อีกด้วย ซึ่งการประมาณการค่าใช้จ่ายใช้หลักการเหมือนภาคอินเตอร์มีเดีท โดยแบ่งเป็นระยะตามวัฏจักรในการพัฒนาซอฟต์แวร์ โดยข้อมูลที่ใช้ในการประมาณการค่าใช้จ่าย นอกจากจำนวนบรรทัดคำสั่งยังใช้อัตรา(rating)ของสภาพแวดล้อมที่มีผลกระทบต่อการพัฒนาซอฟต์แวร์ 15 ลักษณะแบ่งเป็นระยะตามวัฏจักรในการพัฒนาซอฟต์แวร์อันประกอบด้วย การวางแผนและศึกษาความต้องการ การออกแบบ การเขียนคำสั่ง และการรวมและทดสอบ เป็นข้อมูลในการประมาณการค่าใช้จ่ายซอฟต์แวร์ในระดับโปรแกรมย่อยและระดับมอดุล การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ภาคนี้เหมาะสำหรับใช้ประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์สำหรับโครงงานที่มีขนาดใหญ่มาก ประกอบด้วยหลายๆโปรแกรมย่อย ในแต่ละโปรแกรมย่อยอาจประกอบด้วยหลายๆมอดุล ทำให้ผู้จัดการโครงงานควบคุมการผลิตภัณฑ์ซอฟต์แวร์และวัดความก้าวหน้าในการพัฒนาซอฟต์แวร์ได้ละเอียดยิ่งขึ้น

ภาวะในการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโม

การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโมนี้ ในแต่ละภาคยังแบ่งการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ออกเป็น 3 ภาวะ ซึ่งขึ้นอยู่กับสภาพแวดล้อมในการพัฒนาซอฟต์แวร์ ตามรายละเอียดดังนี้

1. ภาวะออแกนนิค (Organic mode)

เป็นการประมาณการค่าใช้จ่ายซอฟต์แวร์ที่ใช้สำหรับทีมงานเล็ก ๆ มีความคุ้นเคยกันดี สภาพแวดล้อมในการพัฒนาเดิม ๆ ทีมงานส่วนมากมีประสบการณ์ในทำงานสูงมาก ซึ่งอาจทำงานเกี่ยวข้องกับระบบนั้นหรือทำงานอยู่ในหน่วยงานนั้น มีความเข้าใจระบบงานตามความต้องการของหน่วยงาน โดยทีมงานสามารถพัฒนาซอฟต์แวร์ในระยะเริ่มต้นประสบผลสำเร็จด้วยดี ขนาดที่ใช้ประมาณการตั้งแต่ 2000 ถึง 50000 บรรทัดคำสั่ง เช่น แบบจำลองทางธุรกิจ แบบจำลองทางวิทยาศาสตร์ ระบบสินค้าคงคลัง ระบบควบคุมผลผลิต เป็นต้น

2. ภาวะเอ็มเบ็ดเด็ด (Embedded mode)

เป็นการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ที่ทีมงานในการพัฒนาซอฟต์แวร์มีประสบการณ์ในระบบงานน้อย สภาพแวดล้อมในการพัฒนาซอฟต์แวร์ไม่คุ้นเคย มีข้อจำกัดหรือข้อบังคับในการพัฒนาซอฟต์แวร์มากทั้งทางด้านฮาร์ดแวร์และทางด้านซอฟต์แวร์ วิธีการพิจารณาการดำเนินการยุ่งยากซับซ้อนมาก ขนาดที่ใช้ประมาณการได้ไม่เกิน 512,000 บรรทัดคำสั่ง ตัวอย่างเช่นระบบควบคุมการจราจร ระบบฝากถอนเงินอัตโนมัติ เป็นต้น

3. ภาวะเซไมดิเท็ด (Semidetached mode)

เป็นการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ ที่ใช้ประมาณการในสภาพแวดล้อมที่อยู่ระหว่างภาวะออแกนนิคและภาวะเอ็มเบ็ดเด็ด คือทีมงานในการพัฒนาซอฟต์แวร์มีการผสมกันระหว่างผู้มีประสบการณ์กับผู้ไม่มีประสบการณ์ทำงานร่วมกัน หรือสมาชิกในทีมงานบางคนเคยทำงานเกี่ยวข้องกับระบบพอสมควร ขนาดที่ใช้ประมาณการตั้งแต่ 2,000 ถึง 300,000 บรรทัดคำสั่ง ตัวอย่างเช่น ระบบการจัดการใหม่ คำสั่งควบคุม ระบบจัดการฐานข้อมูลใหม่ เป็นต้น

การประมาณการความพยายามและจัดกำหนดการผลิตภัณฑ์ซอฟต์แวร์ภาคเบสิก

ตารางที่ 2-1 แสดงถึงสมการในการประมาณการความพยายามและจัดกำหนดการผลิตภัณฑ์ซอฟต์แวร์ภาคเบสิกสำหรับภาวะออแกนนิค ภาวะเซไมดิเท็ด และภาวะเอ็มเบ็ดเด็ด

ตารางที่ 2-1 แสดงสมการในการประมาณการความพยายามและจัดกำหนดการภาคเบสิก[1]

ภาวะ	ความพยายาม	จัดกำหนดการ
ออแกนนิค	$MM = 2.4\ (KDSI)^{1.05}$	$TDEV = 2.5\ (MM)^{0.38}$
เซไมดิเท็ด	$MM = 3.0\ (KDSI)^{1.12}$	$TDEV = 2.5\ (MM)^{0.35}$
เอ็มเบ็ดเด็ด	$MM = 3.6\ (KDSI)^{1.20}$	$TDEV = 2.5\ (MM)^{0.32}$

โดยที่ MM คือ ความพยายามในการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ (คน-เดือน)
โดย 1 เดือนทำงาน 152 ชั่วโมง
กรณีเป็น คน-ชั่วโมง คูณด้วย 152,
คน-วัน คูณด้วย 19,
คน-ปี หารด้วย 19

KDSI คือ จำนวน 1000 บรรทัดคำสั่ง

DSI คือ จำนวนบรรทัดของคำสั่งที่สามารถแปลเป็นภาษาเครื่องได้ ไม่รวมหมายเหตุ และโปรแกรมอัตถประโยชน์
กรณีในหนึ่งบรรทัดมีหลายคำสั่งจะถือเป็น 1 DSI

TDEV คือ เวลาที่ใช้ในการพัฒนา(เดือน) ซึ่งเริ่มตั้งแต่ระยะในการออกแบบผลิตภัณฑ์ ถึง ระยะในการรวมและทดสอบ

[1]Barry Boehm, Software Engineering Economic, (Prentice-Hall, 1981), p.75.

จากสมการในการประมาณการค่าใช้จ่ายความพยายามและจัดกำหนดการ ตามตาราง ที่ 2-1 นั้นเป็นการประมาณการค่าใช้จ่ายความพยายามและจัดกำหนดการทางด้านซอฟต์แวร์โดยรวมทั้งโครงงาน ซึ่งในการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโมนี้สามารถประมาณการค่าใช้จ่ายความพยายามและจัดกำหนดการโดยแบ่งเป็นระยะตามวัฏจักรในการพัฒนาซอฟต์แวร์อันประกอบด้วย ระยะการวางแผนและศึกษาความต้องการ ระยะการออกแบบผลิตภัณฑ์ ระยะการเขียนคำสั่ง และระยะการรวมและทดสอบ ซึ่งประมาณการเป็นเปอร์เซนต์ขึ้นกับภาวะในการพัฒนา และ ขนาดของผลิตภัณฑ์ซอฟต์แวร์ ตามตารางที่ 2-2

ตารางที่ 2-2 แสดงเปอร์เซนต์ในการกระจายระยะของการประมาณการความพยายามและจัดกำหนดการในการพัฒนาซอฟต์แวร์ทุกภาวะ[1]

Effort distribution		Size				
Mode	Phase	Small 2 KDSI	Inter-mediate 8 KDSI	Medium 32 KDSI	Large 128 KDSI	Very Large 512 KDSI
Organic	Plans and requirements (%)	6	6	6	6	
	Product design	16	16	16	16	
	Programming	68	65	62	59	
	Detailed design	26	25	24	23	
	Code and unit test	42	40	38	36	
	Integration and test	16	19	22	25	
Semidetached	Plans and requirements (%)	7	7	7	7	7
	Product design	17	17	17	17	17
	Programming	64	61	58	55	52
	Detailed design	27	26	25	24	23
	Code and unit test	37	35	33	31	29
	Integration and test	19	22	25	28	31
Embedded	Plans and requirements (%)	8	8	8	8	8
	Product design	18	18	18	18	18
	Programming	60	57	54	51	48
	Detailed design	28	27	26	25	24
	Code and unit test	32	30	28	26	24
	Integration and test	22	25	28	31	34
Schedule distribution		2 KDSI	8 KDSI	32 KDSI	128 KDSI	512 KDSI
Organic	Plans and requirements (%)	10	11	12	13	
	Product design	19	19	19	19	
	Programming	63	59	55	51	
	Integration and test	18	22	26	30	
Semidetached	Plans and requirements (%)	16	18	20	22	24
	Product design	24	25	26	27	28
	Programming	56	52	48	44	40
	Integration and test	20	23	26	29	32
Embedded	Plans and requirements (%)	24	28	32	36	40
	Product design	30	32	34	36	38
	Programming	48	44	40	36	32
	Integration and test	22	24	26	28	30

[1] Barry Boehm, Software Engineering Economic, (Prentice-Hall, 1981), p.90

ในการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยแบ่งการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์เป็นระยะตามวัฏจักรในการพัฒนาซอฟต์แวร์ กรณีที่ขนาดของโครงงานไม่สอดคล้องกับขนาดของผลิตภัณฑ์ตามตารางที่ 2-2 ต้องใช้วิธี ลิเนียร์ อินเตอร์โพเลชั่น (Linear interpolation) นั่นคือถ้ากำหนดให้ x0 และ x1 เป็นขนาดของผลิตภัณฑ์ซอฟต์แวร์ y0 และ y1 เป็นเปอร์เซนต์ของการประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ของขนาด x0 และ x1 ตามลำดับ เราสามารถหาค่า y ที่สอดคล้องกับจุด x ระหว่าง x0 และ x1 (x0 <= x <= x1) ตามสมการที่ 2.1

$$y = y0 + \left(\frac{x-x0}{x1-x0}\right)(y1-y0) \qquad \ldots\ldots 2.1$$

นอกจากนี้การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโมนี้ยังสามารถประมาณการจำนวนคนที่ใช้ในแต่ละระยะตามวัฏจักรในการพัฒนาซอฟต์แวร์ ซึ่งในแต่ละระยะยังแบ่งออกเป็น 8 กิจกรรม ประกอบด้วย

1. การวิเคราะห์ความต้องการ
2. การออกแบบผลิตภัณฑ์
3. การเขียนคำสั่ง
4. การทดสอบแผนงาน
5. การทวนสอบและตรวจสอบความสมเหตุสมผล
6. หน้าที่ของโครงงานในสำนักงาน
7. การจัดการรูปร่าง และประกันคุณภาพ
8. การจัดทำคู่มือ

การประมาณการจำนวนคนเฉลี่ยที่ใช้ในระยะต่างๆตามวัฏจักรในการพัฒนา ในแต่ละกิจกรรมต่าง ๆ นั้นขึ้นอยู่กับขนาด และ ภาวะในการพัฒนาซอฟต์แวร์ โดยคิดเป็นเปอร์เซนต์จากจำนวนคนทั้งหมดที่ใช้ในแต่ละระยะของการพัฒนาซอฟต์แวร์ ตามตารางที่ 2-3 2-4 และ 2-5 ในกรณีที่ขนาดของซอฟต์แวร์ไม่สอดคล้องกับขนาดตามสมการ สามารถจะประมาณการได้โดยวิธีการลิเนียร์ อินเตอร์โพเลชั่น ตามสมการที่ 2.1

ตารางที่ 2-3 แสดงเปอร์เซนต์ของการกระจายกิจกรรมโครงงานของภาวะออแกนนิค[1]

Phase	Plans and Requirements	Product Design	Programming	Integration and Test	Development	Maintenance
Product Size	S I M L	S I M L	S I M L	S I M L	S I M L	S I M L
Overall Phase Percentage	6	16	68 65 62 59	16 19 22 25		
Activity percentage						
Requirements analysis	46	15	5	3	6	7
Product design	20	40	10	6	14	13
Programming	3	14	58	34	48 47 46 45	45 44 43 42
Test planning	3	5	4	2	4	3
Verification and validation	6	6	6	34	10 11 12 13	10 11 12 13
Project office	15	11	6	7	7	7
CM/QA	2	2	6	7	5	5
Manuals	5	7	5	7	6	10

[1]Barry Boehm, Software Engineering Economic, (Prentice-Hall, 1981),p.101

ตารางที่ 2-4 แสดงเปอร์เซนต์ของการกระจายกิจกรรมโครงงานของภาวะเซไมดิเท็ด

Phase	Plans and Requirements					Product Design					Programming					Integration and Test					Development					Maintenance				
Product Size	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL
Overall Phase Percentage	7	7	7	7	7	17	17	17	17	17	64	61	58	55	52	19	22	25	28	31										
Activity percentage																														
Requirements analysis	48	47	46	45	44	12.5	12.5	12.5	12.5	12.5	4	4	4	4	4	2.5	2.5	2.5	2.5	2.5	5	5	5	5	5	6.5	6.5	6.5	6	6
Product design	16	16.5	17	17.5	18	41	41	41	41	41	8	8	8	8	8	5	5	5	5	5	13	13	13	13	13	12	12	12	12	12
Programming	2.5	3.5	4.5	5.5	6.5	12	12.5	13	13.5	14	56.5	56.5	56.5	56.5	56.5	33	35	37	39	41	45	45	44.5	44.5	44.5	41.5	41.5	41	41	41
Test planning	2.5	3	3.5	4	4.5	4.5	5	5.5	6	6.5	4	4.5	5	5.5	6	2.5	2.5	3	3	3.5	4	4	4.5	5	5.5	3	3	3.5	4	4.5
Verification and validation	6	6.5	7	7.5	8	6	6.5	7	7.5	8	7	7.5	8	8.5	9	32	31	29.5	28.5	27	11	12	13	13.5	14	11	12	13	13.5	14
Project office	15.5	14.5	13.5	12.5	11.5	13	12	11	10	9	7.5	7	6.5	6	5.5	8.5	8	7.5	7	6.5	8.5	8	7.5	7	6.5	8.5	8	7.5	7	6.5
CM/QA	3.5	3	3	3	2.5	3	2.5	2.5	2.5	2	7	6.5	6.5	6.5	6	8.5	8	8	8	7.5	6.5	6	6	6	5.5	6.5	6	6	6	5.5
Manuals	6	6	5.5	5	5	8	8	7.5	7	7	6	6	5.5	5	5	8	8	7.5	7	7	7	7	6.6	6	6	11	11	10.5	10.5	10.5

[1]Barry Boehm, Software Engineering Economic, (Prentice-Hall, 1981),p. 99

ตารางที่ 2-5 แสดงเปอร์เซนต์ของการกระจายกิจกรรมโครงงานของภาวะเอ็มเบ็ดเด็ด[1]

Phase	Plans and Requirements					Product Design					Programming					Integration and Test					Development					Maintenance				
Product Size	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL
Overall Phase Percentage	8	8	8	8	8	18	18	18	18	18	60	57	54	51	48	22	25	28	31	34										
Activity percentage																														
Requirements analysis	50	48	46	44	42	10	10	10	10	10	3	3	3	3	3	2	2	2	2	2	4	4	4	4	4	6	6	6	5	5
Product design	12	13	14	15	16	42	42	42	42	42	6	6	6	6	6	4	4	4	4	4	12	12	12	12	12	11	11	11	11	11
Programming	2	4	6	8	10	10	11	12	13	14	55	55	55	55	55	32	36	40	44	48	42	43	43	44	45	38	39	39	40	41
Test planning	2	3	4	5	6	4	5	6	7	8	4	5	6	7	8	3	3	4	4	5	4	4	5	6	7	3	3	4	5	6
Verification and validation	6	7	8	9	10	6	7	8	9	10	8	9	10	11	12	30	28	25	23	20	12	13	14	14	14	12	13	14	14	14
Project office	16	14	12	10	8	15	13	11	9	7	9	8	7	6	5	10	9	8	7	6	10	9	8	7	6	10	9	8	7	6
CM/QA	5	4	4	4	3	4	3	3	3	2	8	7	7	7	6	10	9	9	9	8	8	7	7	7	6	8	7	7	7	6
Manuals	7	7	6	5	5	9	9	8	7	7	7	7	6	5	5	9	9	8	7	7	8	8	7	6	6	12	12	11	11	11

[1]Barry Boehm, Software Engineering Economic, (Prentice-Hall, 1981),p. 100

การประมาณการความพยายามและจัดกำหนดการผลิตภัณฑ์ซอฟต์แวร์ภาคอินเตอร์มีเดีท

การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโมภาคอินเตอร์มีเดีทนั้นมีวิธีประมาณการความพยายามและจัดกำหนดการผลิตภัณฑ์ซอฟต์แวร์ โดยให้ผลลัพธ์ในการประมาณการค่าใช้จ่ายต่าง ๆ เหมือนกับภาคเบสิก แตกต่างกันที่ข้อมูลที่ใช้ในการประมาณการโดยเพิ่มข้อมูลสภาพแวดล้อมต่าง ๆ ที่มีผลกระทบต่อการพัฒนาซอฟต์แวร์ 15 ลักษณะซึ่งเรียกว่าตัวขับค่าใช้จ่าย เป็นตัวคูณในสมการการประมาณการความพยายาม ดังตารางที่ 2-6

ตารางที่ 2-6 แสดงสมการในการประมาณการความพยายามและจัดกำหนดการโดยใช้แบบจำลองโคโคโมภาคอินเตอร์มีเดีท[1]

ภาวะ	ความพยายาม	จัดกำหนดการ
ออแกนนิค	$MM = 3.2\ (KDSI)^{1.05}\ m(x)$	$TDEV = 2.5\ (MM)^{0.38}$
เซไมดิเท็ด	$MM = 3.0\ (KDSI)^{1.12}\ m(x)$	$TDEV = 2.5\ (MM)^{0.35}$
เอ็มเบ็ดเด็ด	$MM = 2.8\ (KDSI)^{1.20}\ m(x)$	$TDEV = 2.5\ (MM)^{0.32}$

โดยที่ m(x) คือ ตัวคูณของตัวขับค่าใช้จ่าย 15 ลักษณะ

ค่าของตัวคูณของตัวขับค่าใช้จ่ายต่าง ๆ เหล่านี้ขึ้นอยู่กับสภาพแวดล้อมในการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ โดยแบ่งเป็นอัตราตามสภาพแวดล้อมที่แตกต่างกันออกไป รวม 6 อัตราประกอบด้วย ต่ำมาก ต่ำ ปกติ สูง สูงมาก สูงพิเศษ ซึ่งในแต่ละอัตราจะมีค่าซึ่งเป็นตัวคูณที่ใช้ในการประมาณการความพยายามภาคอินเตอร์มีเดีท ดังรายละเอียด ตารางที่ 2-7 สำหรับรายละเอียดของอัตราตัวขับค่าใช้จ่ายซอฟต์แวร์ แสดงดังตารางที่ 2-8 และ 2-9 ตามลำดับ

[1]Barry Boehm, Software Engineering Economic, (Prentice-Hall, 1981), p.117

ตารางที่ 2-7 แสดงตัวคูณความพยายามในการพัฒนาซอฟต์แวร์[1]

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

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

[1]Barry Boehm, Software Engineering Economic, (Prentice-Hall,1981),p.118

ตารางที่ 2-8 แสดงอัตราตัวขับค่าใช้จ่ายซอฟต์แวร์[1]

Cost Driver	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
RELY	Effect: slight inconvenience	Low, easily recoverable losses	Moderate, recoverable losses	High financial loss	Risk to human life	
DATA		$\frac{\text{DB bytes}}{\text{Prog. DSI}} < 10$	$10 \leq \frac{D}{P} < 100$	$100 \leq \frac{D}{P} < 1000$	$\frac{D}{P} \geq 1000$	
CPLX	See Table					
Computer attributes						
TIME			≤ 50% use of available execution time	70%	85%	95%
STOR			≤ 50% use of available storage	70%	85%	95%
VIRT		Major change every 12 months Minor: 1 month	Major: 6 months Minor: 2 weeks	Major: 2 months Minor: 1 week	Major: 2 weeks Minor: 2 days	
TURN		Interactive	Average turnaround <4 hours	4–12 hours	>12 hours	
Personnel attributes						
ACAP	15th percentile*	35th percentile	55th percentile	75th percentile	90th percentile	
AEXP	≤4 months experience	1 year	3 years	6 years	12 years	
PCAP	15th percentile*	35th percentile	55th percentile	75th percentile	90th percentile	
VEXP	≤1 month experience	4 months	1 year	3 years		
LEXP	≤1 month experience	4 months	1 year	3 years		
Project attributes						
MODP	No use	Beginning use	Some use	General use	Routine use	
TOOL	Basic microprocessor tools	Basic mini tools	Basic midi/maxi tools	Strong maxi programming, test tools	Add requirements, design, management, documentation tools	
SCED	75% of nominal	85%	100%	130%	160%	

* Team rating criteria: analysis (programming) ability, efficiency, ability to communicate and cooperate

[1] Barry Boehm, Software Engineering Economic, (Prentice-Hall,1981),p.119

ตารางที่ 2-9 แสดงอัตราความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์[1]

Rating	Control Operations	Computational Operations	Device-dependent Operations	Data Management Operations
Very low	Straightline code with a few non-nested SP* operators: DOs, CASEs, IFTHENELSEs. Simple predicates	Evaluation of simple expressions: e.g., A = B + C * (D - E)	Simple read, write statements with simple formats	Simple arrays in main memory
Low	Straightforward nesting of SP operators. Mostly simple predicates	Evaluation of moderate-level expressions, e.g., D = SQRT (B**2-4.*A*C)	No cognizance needed of particular processor or I/O device characteristics. I/O done at GET/PUT level. No cognizance of overlap	Single file subsetting with no data structure changes, no edits, no intermediate files
Nominal	Mostly simple nesting. Some inter-module control. Decision tables	Use of standard math and statistical routines. Basic matrix/vector operations	I/O processing includes device selection, status checking and error processing	Multi-file input and single file output. Simple structural changes, simple edits
High	Highly nested SP operators with many compound predicates. Queue and stack control. Considerable intermodule control.	Basic numerical analysis: multivariate interpolation, ordinary differential equations. Basic truncation, roundoff concerns	Operations at physical I/O level (physical storage address translations; seeks, reads, etc). Optimized I/O overlap	Special purpose subroutines activated by data stream contents. Complex data restructuring at record level
Very high	Reentrant and recursive coding. Fixed-priority interrupt handling	Difficult but structured N.A.: near-singular matrix equations, partial differential equations	Routines for interrupt diagnosis, servicing, masking. Communication line handling	A generalized, parameter-driven file structuring routine. File building, command processing, search optimization
Extra high	Multiple resource scheduling with dynamically changing priorities. Microcode-level control	Difficult and unstructured N.A.: highly accurate analysis of noisy, stochastic data	Device timing-dependent coding, micro-programmed operations	Highly coupled, dyhamic relational structures. Natural language data management

* SP = structured programming

[1] Barry Boehm, Software Engineering Economic, (Prentice-Hall,1981),p.122

การประมาณการค่าใช้จ่ายด้านซอฟต์แวร์โดยใช้แบบจำลองโคโคโมภาคอินเตอร์มีเดทนี้ ยังสามารถประมาณการค่าใช้จ่ายด้านซอฟต์แวร์ ในกรณีที่มีการดัดแปรซอฟต์แวร์เก่ามาใช้งานได้อีกด้วย โดยสามารถหาจำนวนบรรทัดคำสั่งจากสมการที่ 2.3

$$EDSI = (ASDI)([0.4(DM) + 0.3(CM) + 0.3(IM)]/100) \quad \ldots\ldots 2.3$$

โดยที่ EDSI - จำนวนของบรรทัดคำสั่งที่เพิ่มในผลิตภัณฑ์ซอฟต์แวร์

ASDI - จำนวนของบรรทัดคำสั่งที่แก้ไขจากซอฟต์แวร์ที่กำลังทำงาน

DM - เปอร์เซนต์การแก้ไขการออกแบบ

CM - เปอร์เซนต์การแก้ไขคำสั่ง

IM - เปอร์เซนต์การแก้ไขการรวม

ซึ่งทำให้การประมาณการค่าใช้จ่ายทางด้านซอฟต์แวร์ของผลิตภัณฑ์ซอฟต์แวร์มีความถูกต้องยิ่งขึ้น

<u>การประมาณการความพยายามและจัดกำหนดการผลิตภัณฑ์ซอฟต์แวร์ภาคดีเทล</u>

สำหรับการประมาณการค่าใช้จ่ายซอฟต์แวร์โดยใช้แบบจำลองโคโคโมภาคดีเทลนี้เหมาะสำหรับโครงงานขนาดใหญ่มากๆ โครงงานประกอบด้วยหลายๆโปรแกรมย่อย ในแต่ละโปรแกรมย่อยประกอบด้วยหลายๆมอดุล การประมาณการภาคนี้ขยายขีดความสามารถในการประมาณการภาคอินเตอร์มีเดท โดยสามารถประมาณการค่าใช้จ่ายได้ในระดับโปรแกรมย่อยและระดับมอดุล สมการการประมาณการความพยายามและจัดกำหนดการเหมือนภาคอินเตอร์มีเดท แตกต่างกันที่ค่าของตัวคูณตัวขับค่าใช้จ่ายแบ่งเป็นระยะตามวัฏจักรในการพัฒนาซอฟต์แวร์อันประกอบด้วย ระยะการวางแผนและศึกษาความต้องการ ระยะการออกแบบ ระยะการเขียนคำสั่ง และระยะในการรวมและทดสอบ ซึ่งมีค่าตัวคูณแตกต่างกัน ตามระดับของการประมาณการค่าใช้จ่ายซอฟต์แวร์ดังรายละเอียดตามตารางที่ 2-10 และ ตารางที่ 2-11

ตารางที่ 2-10 แสดงตัวคูณความพยายามในระดับมอดุล[1]

Attribute	Rating	RPD	DD	CUT	IT
CPLX	Very low	.70	.70	.70	.70
	Low	.85	.85	.85	.85
	Nominal	1.00	1.00	1.00	1.00
	High	1.15	1.15	1.15	1.15
	Very high	1.30	1.30	1.30	1.30
	Extra high	1.65	1.65	1.65	1.65
PCAP	Very low	1.00	1.50	1.50	1.50
	Low	1.00	1.20	1.20	1.20
	Nominal	1.00	1.00	1.00	1.00
	High	1.00	.83	.83	.83
	Very high	1.00	.65	.65	.65
VEXP	Very low	1.10	1.10	1.30	1.30
	Low	1.05	1.05	1.15	1.15
	Nominal	1.00	1.00	1.00	1.00
	High	.90	.90	.90	.90
LEXP	Very low	1.02	1.10	1.20	1.20
	Low	1.00	1.05	1.10	1.10
	Nominal	1.00	1.00	1.00	1.00
	High	1.00	.98	.92	.92

ตารางที่ 2-11 แสดงตัวคูณความพยายามในระดับโปรแกรมย่อย[1]

Attribute	Rating	RPD	DD	CUT	IT
Product					
RELY	Very low	.80	.80	.80	.60
	Low	.90	.90	.90	.80
	Nominal	1.00	1.00	1.00	1.00
	High	1.10	1.10	1.10	1.30
	Very high	1.30	1.30	1.30	1.70
DATA	Low	.95	.95	.95	.90
	Nominal	1.00	1.00	1.00	1.00
	High	1.10	1.05	1.05	1.15
	Very high	1.20	1.10	1.10	1.30
Computer					
TIME	Nominal	1.00	1.00	1.00	1.00
	High	1.10	1.10	1.10	1.15
	Very high	1.30	1.25	1.25	1.40
	Extra high	1.65	1.55	1.55	1.95

[1]Barry Boehm, Software Engineering Economic, (Prentice-Hall,1981),p.354

ตารางที่ 2-11 (ต่อ)

Attribute	Rating	RPD	DD	CUT	IT
STOR	Nominal	1.00	1.00	1.00	1.00
	High	1.05	1.05	1.05	1.10
	Very high	1.20	1.15	1.15	1.35
	Extra high	1.55	1.45	1.45	1.85
VIRT	Low	.95	.90	.85	.80
	Nominal	1.00	1.00	1.00	1.00
	High	1.10	1.12	1.15	1.20
	Very high	1.20	1.25	1.30	1.40
TURN	Low	.98	.95	.70	.90
	Nominal	1.00	1.00	1.00	1.00
	High	1.00	1.00	1.10	1.15
	Very high	1.02	1.05	1.20	1.30
Personnel					
ACAP	Very low	1.80	1.35	1.35	1.50
	Low	1.35	1.15	1.15	1.20
	Nominal	1.00	1.00	1.00	1.00
	High	.75	.90	.90	.85
	Very high	.55	.75	.75	.70
AEXP	Very low	1.40	1.30	1.25	1.25
	Low	1.20	1.15	1.10	1.10
	Nominal	1.00	1.00	1.00	1.00
	High	.87	.90	.92	.92
	Very high	.75	.80	.85	.85
Project					
MODP	Very low	1.05	1.10	1.25	1.50
	Low	1.00	1.05	1.10	1.20
	Nominal	1.00	1.00	1.00	1.00
	High	1.00	.95	.90	.83
	Very high	1.00	.90	.80	.65
TOOL	Very low	1.02	1.05	1.35	1.45
	Low	1.00	1.02	1.15	1.20
	Nominal	1.00	1.00	1.00	1.00
	High	.98	.95	.90	.85
	Very high	.95	.90	.80	.70
SCED	Very low	1.10	1.25	1.25	1.25
	Low	1.00	1.15	1.15	1.10
	Nominal	1.00	1.00	1.00	1.00
	High	1.10	1.10	1.00	1.00
	Very high	1.15	1.15	1.05	1.05