การตัดสินใจด้านทรัพยากรในโครงการซอฟต์แวร์โดยใช้แบบจำลองเชิงความเสี่ยง

นายวิบูลย์ เจียมทับทักษิณ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548
ISBN 974-53-2842-1
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

RESOURCE DECISIONS IN SOFTWARE PROJECT USING RISK-BASED MODELS

Mr. Wiboon Jiamthubthugsin

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2005

ISBN 974-53-2842-1

หัวข้อวิทยานิพนธ์	การตัดสินใจด้านทรัพยากรในโครงการซอฟต์แวร์โดยใช้แบบจำลองเชิงความเสี่ยง
โดย	นายวิบูลย์ เจียมทับทักษิณ
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ดาริชา สุธีวงศ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร.ดาริชา สุธีวงศ์)

.....กรรมการ
(อาจารย์ ดร.โปรดปราน บุณยพุกกณะ)

.....กรรมการ
(อาจารย์ นครทิพย์ พร้อมพูล)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นายวิบูลย์ เจียมทับทักษิณ : การตัดสินใจด้านทรัพยากรในโครงการซอฟต์แวร์โดยใช้แบบจำลองเชิงความเสี่ยง. (RESOURCE DECISIONS IN SOFTWARE PROJECT USING RISK-BASED MODELS) อ. ที่ปรึกษา : อ.ดร.ดาริชา สุธีวงศ์, 58 หน้า. ISBN 974-53-2842-1.

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อหาแบบจำลองในการตัดสินใจด้านการจัดสรรทรัพยากรในโครงการซอฟต์แวร์โดยการประยุกต์ใช้แบบจำลองประเมินความเสี่ยงสำหรับการจัดสรรทรัพยากรในโครงการเดียวและแบบจำลองต้นทุน COCOMO II สำหรับการจัดสรรทรัพยากรในกลุ่มโครงการ โดยวิธีการเดิมสำหรับการจัดสรรทรัพยากรในโครงการเดียวนั้นอาจใช้โครงข่ายแบบเบย์ซึ่งสนับสนุนการจัดสรรทรัพยากรเชิงภาวะถ่วงดุล แต่ในกรณีที่ต้องการหารูปแบบการจัดสรรทรัพยากรที่เหมาะที่สุด ผู้บริหารจะต้องเปรียบเทียบคำตอบในแต่ละเหตุการณ์เอง ซึ่งถ้ามีจำนวนเหตุการณ์มากจะทำให้การหาคำตอบมีความซับซ้อน วิทยานิพนธ์นี้จึงเสนอการประยุกต์ใช้แบบจำลองประเมินความเสี่ยงร่วมกับกำหนดการแบบไม่เชิงเส้นเพื่อหารูปแบบการจัดสรรทรัพยากรที่เหมาะที่สุดได้โดยง่าย นอกจากนี้วิธีการเดิมสำหรับจัดสรรทรัพยากรในกลุ่มโครงการนั้นจำกัดทางเลือกที่จะปรับปรุงความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ โดยสามารถปรับเปลี่ยนปัจจัยเพียงสองปัจจัยเท่านั้น กล่าวคือ จำนวนพนักงานและเส้นตาย ดังนั้นวิทยานิพนธ์นี้จึงเสนอการใช้แบบจำลองต้นทุน COCOMO II อันประกอบด้วยปัจจัยจำนวนมากมาร่วมในการตัดสินใจจัดสรรทรัพยากรในกลุ่มโครงการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....วิบูลย์ เจียมทับทักษิณ.....
สาขาวิชา...วิศวกรรมซอฟต์แวร์..... ลายมือชื่ออาจารย์ที่ปรึกษา.....ดาริชา สุธีวงศ์.....
ปีการศึกษา .....2548.....

4770455121 : MAJOR SOFTWARE ENGINEERING

KEY WORD: RESOURCE DECISIONS / SOFTWARE PROJECT MANAGEMENT / RISK MANAGEMENT / SOFTWARE DEVELOPMENT / RISK ASSESSMENT MODEL / PORTFOLIO-BASED PROJECT MANAGEMENT

WIBOON JIAMTHUBTHUGSIN : RESOURCE DECISIONS IN SOFTWARE PROJECT USING RISK-BASED MODELS. THESIS ADVISOR : DARICHA SUTIVONG, Ph.D., 58 pp. ISBN 974-53-2842-1.

This thesis aims to find models that assist resource decisions in software development projects by incorporating a risk assessment model for an individual project management and the COCOMO II cost model for portfolio management. For an individual project, the existing method based on Bayesian networks can be used for trade-off analysis. However, in order to find an optimal resource allocation, a manager must consider all possible scenarios, the number of which can be prohibitively large. This thesis proposes utilizing a risk assessment model along with nonlinear programming to find an optimal resource allocation. For a portfolio of software projects, the current method considers only a few resources, namely the number of employees and deadlines, in order to improve probability of portfolio success. Therefore, this thesis proposes integrating the current model with COCOMO II, which consists of many resource factors, in order to increase alternatives for resource decisions.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department.....Computer Engineering.....Student's signature Wiboon Jiamthubthugsin

Field of study....Software Engineering.....Advisor's signature Daricha Sutivong

Academic year .....2005.....

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา อ.ดร.ดาริชา สุธีวงศ์ ซึ่งได้ให้คำแนะนำตลอดจนแนวทางที่เป็นประโยชน์ต่องานวิจัย รวมทั้งคอยติดตามอย่างใกล้ชิดอันทำให้งานวิจัยนี้ดำเนินไปได้ด้วยดี

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่าน รศ.ดร.บุญเสริม กิจศิริกุล อ.ดร. โปรดปราน บุณยพุกกณะ และ อ.นครทิพย์ พร้อมพูล ที่กรุณาให้คำแนะนำอันเป็นประโยชน์และช่วยตรวจสอบแก้ไขวิทยานิพนธ์ฉบับนี้

ขอบคุณสมาชิกในห้องปฏิบัติการ 20-06 รวมทั้งห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ และเพื่อนๆ ทุกคนที่คอยช่วยเหลือ ให้คำปรึกษา คำแนะนำ และกำลังใจสำหรับการวิจัยครั้งนี้

สุดท้ายนี้ ขอขอบคุณ คุณพ่อ คุณแม่ ญาติพี่น้องทุกคน ที่คอยเป็นห่วงดูแล ให้กำลังใจ และสนับสนุนในทุก ๆ ด้านเสมอมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญภาพ ..... ญ
1. บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 3
1.3 ขอบเขตของการวิจัย 3
1.4 การดำเนินการวิจัย 4
1.5 ประโยชน์ที่คาดว่าจะได้รับ 4
1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 4
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6
2.1 การจัดการความเสี่ยง 6
2.2 COCOMO II 7
2.3 กระบวนการเชิงวิวัฒน์ 15
2.4 งานวิจัยที่เกี่ยวข้อง 16
3. แนวคิดการวิจัย 21
3.1 แนวคิดการจัดสรรทรัพยากรสำหรับโครงการเดียว 21
3.2 แนวคิดการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ 23
4. การทดลองและวิเคราะห์ผล 29
4.1 การทดลองการจัดสรรทรัพยากรสำหรับโครงการเดียว 29
4.2 การทดลองการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ 36
5. สรุปผลการวิจัยและข้อเสนอแนะ 53
5.1 สรุปผลการวิจัย 53
5.2 ข้อเสนอแนะ 53
รายการอ้างอิง 55

หน้า

ประวัติผู้เขียนวิทยานิพนธ์.....58

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 2.1 การระบุระดับของตัวประกอบมาตราส่วน 11
ตารางที่ 2.2 การระบุระดับของตัวขับต้นทุน 12
ตารางที่ 2.3 การระบุระดับของ CPLX 14
ตารางที่ 2.4 ความไม่แน่นอนของขนาดของซอฟต์แวร์ตามเฟสการพัฒนา.....20
ตารางที่ 3.1 ตัวเลขแทนค่าระดับของปัจจัยใน COCOMO II 24
ตารางที่ 3.2 การทดลองประโยชน์ของการบริหารแบบกลุ่มโครงการ 26
ตารางที่ 3.3 ค่ามากและน้อยที่สุดสำหรับแต่ละปัจจัย.....28
ตารางที่ 4.1 ค่ากลางของช่วงอัตราการเกิดและตายและการเปลี่ยนแปลงของความต้องการ 30
ตารางที่ 4.2 ค่าคงที่ของตัวแปรที่ใช้ในการทดลอง.....31
ตารางที่ 4.3 ค่าระดับโดยปริยายของตัวประกอบมาตราส่วนและตัวขับต้นทุน.....36
ตารางที่ 4.4 ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ.....39
ตารางที่ 4.5 ผลการทดลองของการปรับเปลี่ยน TOOL.....40
ตารางที่ 4.6 ประโยชน์จากการบริหารแบบกลุ่มโครงการโดยใช้เงื่อนไขความเพียรพยายาม.....41
ตารางที่ 4.7 ประโยชน์จากการบริหารแบบกลุ่มโครงการโดยใช้เงื่อนไขเวลา.....41
ตารางที่ 4.8 ค่าความเพียรพยายามและเดือนสิ้นกำหนดที่ทำให้ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเท่ากับ 50%.....42
ตารางที่ 4.9 ความเพียรพยายามและเวลาลำดับที่ 625 และ 626 ของกลุ่มโครงการ 43
ตารางที่ 4.10 ลำดับการเรียงของตัวขับต้นทุนตามค่าช่วงห่างสำหรับเงื่อนไขความเพียรพยายามและเวลา.....49
ตารางที่ 4.11 ลำดับการเรียงของตัวขับต้นทุนตามความน่าจะเป็นสำหรับเงื่อนไขความเพียรพยายามและเวลา.....51

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 2.1 กระบวนการแบบเวียน....16
รูปที่ 2.2 ตัวอย่างการแจกแจงความน่าจะเป็นแบบไวบุลล์....18
รูปที่ 3.1 ตัวอย่างการแจกแจงความน่าจะเป็นของ TOOL....25
รูปที่ 4.1 ผลิตภาพของพนักงานที่ต้องการ ณ การเปลี่ยนแปลงความต้องการที่ระดับต่าง ๆ....31
รูปที่ 4.2 เวลาที่ใช้ในการพัฒนา ณ ผลิตภาพของพนักงานที่ระดับต่าง ๆ....32
รูปที่ 4.3 อัตราการเปลี่ยนแปลงของความน่าจะเป็นที่โครงการจะเสร็จสิ้นและเวลาที่ใช้ในการพัฒนา....33
รูปที่ 4.4 ค่าตอบแทนคาดหมายสุทธิ ณ เวลาและผลิตภาพของพนักงาน....34
รูปที่ 4.5 ค่าตอบแทนคาดหมายสุทธิและเวลา....34
รูปที่ 4.6 ค่าตอบแทนคาดหมายสุทธิและผลิตภาพของพนักงาน....35
รูปที่ 4.7 ขนาดของซอฟต์แวร์....37
รูปที่ 4.8 ค่าลอการิทึมของขนาดของซอฟต์แวร์....38
รูปที่ 4.9 ตัวอย่างการแจกแจงความน่าจะเป็นของความเพียรพยายาม....38
รูปที่ 4.10 ตัวอย่างการแจกแจงความน่าจะเป็นของเวลา....39
รูปที่ 4.11 แผนภูมิทอร์นาโดของปัจจัยด้านทรัพยากรสำหรับเงื่อนไขความเพียรพยายาม....44
รูปที่ 4.12 แผนภูมิทอร์นาโดของปัจจัยด้านทรัพยากรสำหรับเงื่อนไขเวลา....45
รูปที่ 4.13 แผนภูมิแท่งของความน่าจะเป็นที่น้อยที่สุดของปัจจัยด้านทรัพยากร....46
รูปที่ 4.14 แผนภูมิแท่งของความน่าจะเป็นที่มากที่สุดของปัจจัยด้านทรัพยากร....47

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กระบวนการแรก ๆ ในการพัฒนาซอฟต์แวร์คือ การวางแผน ซึ่งเป็นการกำหนดว่าควรทำอะไรและเมื่อไรในการพัฒนาซอฟต์แวร์ และในช่วงการวางแผนนี้จะประกอบด้วยขั้นตอนหลายขั้นตอน หนึ่งในขั้นตอนเหล่านี้คือ การประมาณต้นทุนหรือการจัดสรรทรัพยากร ซึ่งขั้นตอนนี้จะประมาณและกำหนดต้นทุนที่จะใช้สำหรับการพัฒนาซอฟต์แวร์ วิธีที่ใช้ในการประมาณทรัพยากรโดยทั่วไปมีด้วยกัน 3 แบบ คือ แบบจำลองต้นทุน (Algorithmic Cost Model) การเรียนรู้ด้วยเครื่อง (Machine Learning) และการตัดสินโดยผู้เชี่ยวชาญ (Expert Judgment) **(1, 2)**

อย่างไรก็ตามเนื่องจากขั้นตอนนี้กระทำในช่วงแรกจึงทำให้ข้อมูลไม่เพียงพออันทำให้ค่าที่ประมาณได้นั้นขาดความแม่นยำ ซึ่งจะก่อให้เกิดความเสี่ยงต่อโครงการจากความคลาดเคลื่อนของการประมาณค่าทรัพยากร เช่น ถ้าทรัพยากรที่ได้จากการประมาณนั้นน้อยเกินไปอาจทำให้ทรัพยากรที่ต้องการใช้ไม่เพียงพอทำให้โครงการไม่สำเร็จหรือเสร็จล่าช้าได้ หรือถ้าทรัพยากรที่ได้จากการประมาณนั้นมีจำนวนมากเกินไปจะทำให้สิ้นเปลืองงบประมาณ ดังนั้นในการประมาณทรัพยากรในโครงการพัฒนาซอฟต์แวร์จึงจำเป็นต้องมีการควบคุมความเสี่ยงเหล่านี้ด้วย

โดยทั่วไปการประเมินความเสี่ยงจะอยู่ในรูปแบบของความน่าจะเป็นซึ่งช่วยให้สามารถรับรู้ความไม่แน่นอนของความเสี่ยงเหล่านี้ในเชิงปริมาณได้ เช่น งานวิจัยของ Fenton และคณะ **(3)** ซึ่งแสดงความไม่แน่นอนสำหรับแต่ละส่วนของการพัฒนาซอฟต์แวร์ในรูปของความน่าจะเป็นมีเงื่อนไข (Conditional Probability) เช่น ความน่าจะเป็นของเวลาที่ใช้ในการพัฒนาเมื่อรู้ความเพียรพยายาม (Effort) ความน่าจะเป็นของคุณภาพของซอฟต์แวร์เมื่อรู้คุณภาพของกระบวนการและบุคลากร เป็นต้น พร้อมทั้งสามารถช่วยในการตัดสินใจเกี่ยวกับทรัพยากรเชิงภาวะถ่วงดุล (Trade-off) ซึ่งวิธีมาตรวัดดั้งเดิมไม่สามารถสนับสนุนได้ ภาวะถ่วงดุลนี้สามารถช่วยสนับสนุนการตัดสินใจของผู้บริหารได้ดี กล่าวคือผู้บริหารสามารถรับรู้ได้ว่าทรัพยากรใดมีผลต่อทรัพยากรใดและสามารถทำการปรับหรือลดทรัพยากรที่ต้องการโดยการปรับหรือลดทรัพยากรที่เกี่ยวข้องได้ ตัวอย่างเช่น ในกรณีที่ต้องการคุณภาพของความต้องการ (Requirements Quality) ที่สมบูรณ์จะต้องใช้คนและกระบวนการที่มีคุณภาพสูงมาก แต่ถ้าผู้บริหารตัดสินใจที่จะลดให้คุณภาพของความ

ต้องการเหลือแค่ปานกลางก็จะทำให้สามารถใช้คุณภาพของคนและกระบวนการเหลือแค่ระดับสูงเท่านั้น

อย่างไรก็ตามวิธีการนี้ยังประสบปัญหาในบางประการคือ ในกรณีที่ผู้บริหารต้องการหาการจัดสรรทรัพยากรที่ดีที่สุด ผู้บริหารจะต้องทำการเปรียบเทียบคำตอบที่เป็นไปได้ของแต่ละเหตุการณ์ (Scenario) เอง ซึ่งในกรณีที่จำนวนเหตุการณ์มีมากย่อมทำให้การหาคำตอบที่ต้องการมีความซับซ้อนมากยิ่งขึ้นผู้บริหารจึงตัดสินใจได้ไม่สะดวก และนอกจากนี้ค่าของคำตอบที่ได้มีลักษณะค่อนข้างตายตัว เช่น ค่าที่เป็นไปได้ของคุณภาพคนและกระบวนการคือ น้อยมาก น้อย ปานกลาง สูง และสูงมาก จากลักษณะที่กล่าวมานี้ทำให้ผู้ใช้ไม่สามารถเลือกใช้ชุดของค่าที่ต้องการได้ ตัวอย่างเช่น ผู้ใช้อาจจะต้องการมาตราของคำตอบที่ละเอียดหรือหยาบกว่านี้ เช่น น้อยพิเศษ น้อยมาก น้อย ปานกลาง สูง สูงมาก และสูงพิเศษ นอกจากนี้ค่าเหล่านี้อยู่ในลักษณะไม่ต่อเนื่อง (Discrete) ทำให้ความละเอียดของค่าคำตอบที่ได้ขึ้นอยู่กับการแบ่งช่วงของค่า ถึงแม้จะสามารถแก้ไขด้วยการแบ่งค่าให้มีจำนวนช่วงมาก ๆ แต่จำนวนช่วงที่มากนี้ย่อมส่งผลให้การคำนวณมีความซับซ้อนมากขึ้นด้วย

ดังนั้นงานวิจัยนี้จึงได้เสนอการใช้แบบจำลองประเมินความเสี่ยง สำหรับการจัดสรรทรัพยากรในการพัฒนาซอฟต์แวร์ซึ่งสามารถช่วยแก้ปัญหาที่กล่าวมาคือ สามารถหาคำตอบที่เหมาะที่สุด (Optimal Solution) โดยกำหนดเงื่อนไขทรัพยากรที่ต้องการได้โดยง่าย และเนื่องจากคำตอบอยู่ในลักษณะค่าต่อเนื่อง จึงทำให้ค่ามีความละเอียดและสามารถกำหนดมาตราที่ผู้ใช้ต้องการได้เอง จากที่ได้กล่าวตอนต้นว่าวิธีการประมาณทรัพยากรมี 3 วิธีนั้น การใช้แบบจำลองประเมินความเสี่ยงถือเป็นประเภทหนึ่งของแบบจำลองต้นทุน ดังนั้นจึงมีข้อดี คือ ไม่ขึ้นอยู่กับคุณภาพของผู้เชี่ยวชาญเหมือนวิธีการตัดสินโดยผู้เชี่ยวชาญ **(4)** และไม่จำเป็นต้องใช้ข้อมูลจำนวนมากเหมือนดังวิธีการเรียนรู้ด้วยเครื่อง จึงสามารถประยุกต์ใช้ได้กับโครงการที่ขาดข้อมูลในอดีต ถึงแม้คำตอบที่ได้จากวิธีแบบจำลองอาจไม่แม่นยำเท่าวิธีทั้งสอง ที่สร้างมาโดยเฉพาะสำหรับบริษัทหรือสถานการณ์หนึ่ง ๆ แต่อย่างน้อยก็สามารถช่วยเป็นแนวทางในการตัดสินใจเมื่อประสบข้อจำกัดดังที่ได้กล่าวมา

อนึ่งในการจัดสรรทรัพยากรนั้นผู้บริหารอาจสามารถบริหารทรัพยากรของหลาย ๆ โครงการพร้อมกันได้ ดังนั้นจึงมีแนวคิดของการจัดสรรทรัพยากรแบบกลุ่มโครงการ (Portfolio) เช่น งานวิจัยที่มีการพิจารณาความเสี่ยงสำหรับกลุ่มซอฟต์แวร์ของ Fewster และคณะ **(5)** แต่จากการศึกษาพบว่าวิธีนี้ยังมีข้อจำกัดบางประการคือ ปัจจัยที่ใช้พิจารณาในการตัดสินใจยังมีจำนวนน้อย คือ มีเพียงแค่จำนวนพนักงานและเส้นตาย (Deadline) เท่านั้น ซึ่งในการตัดสินใจที่มีประสิทธิภาพควรให้ความสำคัญทั้ง 4 ด้าน คือ บุคลากร (People) ผลิตภัณฑ์ (Product) กระบวนการ

(Process) และโครงการ (Project) **(6)** ดังนั้นงานวิจัยนี้จึงได้เสนอการประยุกต์ใช้แบบจำลองต้นทุน (COCOMO II) ในการจัดสรรทรัพยากรของกลุ่มซอฟต์แวร์ ซึ่งการประยุกต์ใช้แบบจำลองต้นทุนนี้นอกจากจะคงความสามารถในการจัดการความเสี่ยงแล้ว การใช้แบบจำลองต้นทุนยังสามารถเพิ่มทางเลือกในการตัดสินใจจากปัจจัยด้านทรัพยากรที่มีจำนวนมากขึ้น เช่น การใช้เครื่องมือ ความสามารถในการสื่อสาร ความสามารถของบุคลากร เป็นต้น

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์คือ นำแบบจำลองประเมินความเสี่ยงมาประยุกต์ใช้ในการจัดสรรทรัพยากรสำหรับโครงการเดียวทำให้สามารถวิเคราะห์ทรัพยากรเชิงภาวะถ่วงดุลและหาคำตอบที่เหมาะที่สุดได้ นอกจากนี้ยังได้ใช้แบบจำลองต้นทุนในการเพิ่มปัจจัยด้านทรัพยากรสำหรับการตัดสินใจในการจัดสรรทรัพยากรแบบกลุ่มโครงการ

1.3 ขอบเขตของการวิจัย

1. ในการจัดสรรทรัพยากรในโครงการเดียวใช้แบบจำลองประเมินความเสี่ยงแบบค่าต่อเนื่อง

2. ในการจัดสรรทรัพยากรในโครงการเดียวใช้สมการอย่างง่ายสำหรับหารูปแบบการจัดสรรทรัพยากรที่เหมาะที่สุด กล่าวคือพิจารณาเฉพาะปัจจัยที่เกี่ยวข้องกับแบบจำลองประเมินความเสี่ยง คือ การเปลี่ยนแปลงของความต้องการ ผลิตภาพของพนักงาน ขนาดของซอฟต์แวร์ และเวลาที่ใช้ในการพัฒนา

3. ใช้ COCOMO II ตรวจสอบความสมเหตุสมผลของแบบจำลองประเมินความเสี่ยงและพิจารณาเฉพาะเหตุการณ์กลาง

4. ตัววัดประโยชน์ของการบริหารแบบกลุ่มโครงการคืออัตราส่วนของความน่าจะเป็นของโครงการที่บริหารแบบกลุ่มโครงการต่อความน่าจะเป็นของโครงการที่บริหารแบบเป็นอิสระต่อกัน

5. การวิเคราะห์ความไวของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จต่อปัจจัยด้านทรัพยากร จะสมมุติว่าผู้บริหารเลือกใช้ปัจจัยทรัพยากรนั้นกับทุกโครงการในกลุ่มโครงการ เช่น ถ้าผู้บริหารเลือกปรับเปลี่ยนเครื่องมือก็จะปรับเปลี่ยนเครื่องมือของทุกโครงการในกลุ่มโครงการ

1.4 การดำเนินการวิจัย

1. ศึกษาทฤษฎีความเสี่ยงในโครงการซอฟต์แวร์ และแบบจำลองประเมินความเสี่ยงในโครงการซอฟต์แวร์

2. ศึกษาการบริหารโครงการซอฟต์แวร์แบบกลุ่มโครงการ และแบบจำลองต้นทุน COCOMO II

3. ออกแบบวิธีการทดลอง

4. ทำการทดลอง

5. วิเคราะห์ผลการทดลอง

6. สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถใช้แบบจำลองประเมินความเสี่ยงเป็นอีกทางเลือกหนึ่งในการจัดสรรทรัพยากรเชิงภาวะถ่วงดุลในการพัฒนาซอฟต์แวร์สำหรับโครงการเดียว พร้อมทั้งสามารถหาคำตอบที่เหมาะที่สุดได้โดยง่ายสำหรับสถานการณ์ต่างๆ กล่าวคือในเหตุการณ์ที่ทรัพยากรในโครงการมีอยู่จำกัด ผู้บริหารควรจะจัดสรรทรัพยากรอย่างไรแก่โครงการเพื่อให้ได้ผลตอบแทนสูงสุด นอกจากนี้สามารถใช้แบบจำลองต้นทุนเพิ่มทางเลือกในการตัดสินใจนอกจากปัจจัยเส้นตายและจำนวนพนักงานพร้อมทั้งตัดสินใจเลือกปัจจัยที่เหมาะสมได้ กล่าวคือในเหตุการณ์ที่ผู้บริหารต้องการเพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ ผู้บริหารอาจเลือกที่จะเปลี่ยนปัจจัยต่าง ๆ ในโครงการ เช่น การใช้เครื่องมือ ความสามารถในการสื่อสาร ความสามารถของบุคลากร เป็นต้น นอกเหนือจากการเพิ่มจำนวนพนักงานหรือขยายเวลาการพัฒนา พร้อมทั้งสามารถเลือกปัจจัยที่มีความเหมาะสมได้โดยดูว่าปัจจัยใดส่งผลมากหรือน้อยต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนเนื้อหาด้านการจัดสรรทรัพยากรสำหรับโครงการเดียวของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “Resource Decisions in Software Development Using Risk Assessment Model” โดย Wiboon Jiamthubthugsin และ Daricha Sutivong **(7)** ในงานประชุมวิชาการ “The 39th Hawaii International Conference on System Sciences” ณ ฮาวาย ประเทศสหรัฐอเมริกา ในระหว่างวันที่ 4-7 มกราคม 2549

และส่วนเนื้อหาด้านการจัดสรรทรัพยากรสำหรับกลุ่มโครงการของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “Portfolio Management of Software Development Projects Using COCOMO II” โดย Wiboon Jiamthubthugsin และ Daricha Sutivong **(8)** ในงานประชุมวิชาการ “The 28th International Conference on Software Engineering” ณ เซี่ยงไฮ้ ประเทศจีน ในระหว่างวันที่ 20-28 พฤษภาคม 2549

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้นำหลักการการจัดการความเสี่ยง แบบจำลองต้นทุน COCOMO II และกระบวนการเชิงวิวัฒน์มาประยุกต์ใช้ในงานวิจัย โดยมีรายละเอียดดังต่อไปนี้

2.1 การจัดการความเสี่ยง (Risk Management)

โครงการซอฟต์แวร์มีลักษณะเหมือนกับโครงการทั่วไป คือ มีความเสี่ยงที่โครงการจะประสบความล้มเหลวและก่อให้เกิดผลเสียในด้านต่าง ๆ ทั้งด้านงบประมาณ เวลา และทรัพยากร ตัวอย่างของเหตุการณ์ความเสี่ยงที่มีความสำคัญสูง คือ การขาดข้อผูกมัดโครงการจากผู้บริหารระดับสูง การขัดข้องในการได้รับข้อผูกมัดจากลูกค้า การเข้าใจความต้องการผิด เป็นต้น **(9)**

ดังนั้นจึงจำเป็นต้องมีการจัดการความเสี่ยงที่จะช่วยลดและหลีกเลี่ยงความเสี่ยงของโครงการซอฟต์แวร์อย่างมีประสิทธิภาพ หลักการพื้นฐานทั่วไปที่ใช้ในการจัดการความเสี่ยงคือหลักการโอกาสความเสี่ยง (Risk Exposure) หรือที่เรียกว่าผลกระทบความเสี่ยง หรือปัจจัยความเสี่ยง (Risk Factor) **(10)** โดยที่

$$RE = P(UO) \cdot L(UO)$$

เมื่อ RE คือ โอกาสความเสี่ยง (Risk Exposure)

$P(UO)$ คือความน่าจะเป็นของเหตุการณ์ความเสี่ยงที่จะก่อให้เกิดความสูญเสีย (Probability of Unsatisfactory Outcome)

$L(UO)$ คือความสูญเสียที่จะเกิดขึ้นเมื่อเกิดเหตุการณ์นั้น (Loss from Unsatisfactory Outcome)

การจัดการความเสี่ยงประกอบด้วยขั้นตอนหลัก 2 ขั้นตอน คือ การประเมินความเสี่ยง (Risk Assessment) และการควบคุมความเสี่ยง (Risk Control) การประเมินความเสี่ยงจะประกอบด้วยการระบุความเสี่ยง (Risk Identification) การวิเคราะห์ความเสี่ยง (Risk Analysis) และการลำดับความสำคัญของความเสี่ยง (Risk Prioritization) และสำหรับการควบคุมความเสี่ยงจะ

ประกอบด้วย การวางแผนจัดการความเสี่ยง (Risk–Management Planning) การขจัดความเสี่ยง (Risk Resolution) และการเฝ้าสังเกตความเสี่ยง (Risk Monitoring)

โดยขั้นตอนในการประเมินความเสี่ยง ขั้นแรกจะทำการระบุความเสี่ยงโดยสร้างรายการของความเสี่ยงที่คาดว่ามีผลต่อความสำเร็จของโครงการ หลังจากที่ทำการระบุความเสี่ยงแล้วจะทำการวิเคราะห์ความเสี่ยงซึ่งจะประเมินความน่าจะเป็นของความสูญเสียและขนาดของความสูญเสียของแต่ละความเสี่ยงที่ได้ระบุแล้ว หลังจากนั้นจะทำการจัดลำดับความสำคัญของความเสี่ยงซึ่งโดยทั่วไปมักจะเรียงตามค่าโอกาสความเสี่ยง

สำหรับขั้นตอนการควบคุมความเสี่ยง ขั้นแรกคือ การวางแผนจัดการความเสี่ยงซึ่งเป็นการเตรียมพร้อมสำหรับกำจัดความเสี่ยงนั้น หลังจากนั้นจะขจัดความเสี่ยงเพื่อกำจัดความเสี่ยงนั้น ๆ และสุดท้ายจะทำการเฝ้าสังเกตความเสี่ยงเพื่อติดตามความก้าวหน้าของการขจัดความเสี่ยงต่าง ๆ

2.2 COCOMO II (Constructive Cost Model II)

COCOMO II **(11-12)** เป็นแบบจำลองต้นทุนสำหรับใช้ในการวางแผนโครงการซอฟต์แวร์ ซึ่งแบ่งออกเป็น 3 ประเภท คือ Early Prototyping, Early Design และ Post-Architecture โดยในงานวิจัยนี้จะศึกษาละเอียดในประเภท Post-Architecture ซึ่งประกอบด้วยปัจจัยที่น่าสนใจจำนวนหลายตัว คือ ตัวประกอบมาตราส่วน (Scale Factor) 5 ตัวและตัวขับต้นทุน (Cost Driver) 17 ตัว โดยตัวประกอบมาตราส่วนเป็นปัจจัยที่มีผลต่อการประหยัดเนื่องจากขยายขนาดผลิต (Economies of Scale) โดยการประหยัดเนื่องจากขยายขนาดผลิตนี้จะแทนด้วยตัวแปร E ดังสมการที่ 2.1 ในกรณีที่ E<1.0 จะหมายความว่าโครงการมีลักษณะที่จะประหยัดเนื่องจากขยายขนาดผลิต ดังนั้นถ้าขนาดของซอฟต์แวร์ถูกเพิ่มสองเท่าความเพียรพยายามที่ใช้เพิ่มจะน้อยกว่าสองเท่า ส่วนตัวขับต้นทุนนั้นแสดงถึงคุณลักษณะของการพัฒนาซอฟต์แวร์ที่ส่งผลต่อความเพียรพยายามที่ใช้ในการทำโครงการ ดังนั้นตัวขับต้นทุนนี้จะใช้เป็นตัวปรับความเพียรพยายามที่ประมาณได้ หลังจากกำหนดขนาดของซอฟต์แวร์และระดับของตัวประกอบมาตราส่วนและตัวขับต้นทุนแล้วสามารถหาความเพียรพยายามและเวลาที่ใช้พัฒนาซอฟต์แวร์ได้ดังสมการที่ 2.1 และ 2.2

$$Effort = A \times Size^{E} \times \prod_{i=1}^{17} EM_i \tag{2.1}$$

$$Time = C \times (Effort)^{F} \tag{2.2}$$

โดยที่ E และ F ในสมการที่ 2.1 และ 2.2 คำนวณจากสมการที่ 2.3 และ 2.4

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j \tag{2.3}$$

$$F = D + 0.2 \times (E - B) \tag{2.4}$$

Size หมายถึงขนาดของซอฟต์แวร์ (KLOC) ส่วน A, B, EM (Effort Multipliers) และ SF (Scale Factor) ได้จากการปรับค่าด้วยข้อมูลความเพียรพยายามของโครงการ ส่วน C และ D ได้จากการปรับค่าด้วยข้อมูลเวลาของโครงการ และรายละเอียดของตัวประกอบมาตราส่วนมีดังนี้

1. การมีแบบอย่างมาก่อน (Precedentedness: PREC)

ถ้าลักษณะซอฟต์แวร์ที่พัฒนาคล้ายคลึงกับของเดิม การมีแบบอย่างมาก่อนจะมีค่าสูง

2. ความยืดหยุ่นในการพัฒนา (Development Flexibility: FLEX)

ความยืดหยุ่นในการพัฒนาจะมีค่ามากถ้าซอฟต์แวร์ไม่จำเป็นต้องตรงกับความต้องการและข้อกำหนดตัวต่อประสานภายนอก

3. สถาปัตยกรรมและการขจัดความเสี่ยง (Architecture/Risk Resolution: RESL)

สถาปัตยกรรมและการขจัดความเสี่ยงจะมีค่ามากในกรณีดังต่อไปนี้ การจัดการความเสี่ยงได้ระบุความเสี่ยงที่วิกฤตทั้งหมดแล้ว กำหนดการและงบประมาณเข้ากันได้กับการจัดการความเสี่ยง ให้เวลากับการสร้างสถาปัตยกรรมมาก มีสถาปัตยกรรมที่ต้องการอยู่แล้ว เครื่องมือที่สนับสนุนการขจัดความเสี่ยง การพัฒนา และการทวนสอบข้อกำหนดด้านสถาปัตยกรรมมีมาก ระดับความไม่แน่นอนของปัจจัยที่มีผลต่อสถาปัตยกรรมมีน้อย และจำนวนของความเสี่ยงที่วิกฤตมีน้อย

4. ความเชื่อมแน่นของทีม (Team Cohesion: TEAM)

ความเชื่อมแน่นของทีมเกี่ยวข้องกับความยากในการประสานเวลาของผู้ที่เกี่ยวข้องกับการพัฒนาซึ่งความยากในการประสานเวลานี้ก่อให้เกิดความปั่นป่วนและความไม่เป็นระเบียบในโครงการ

5. วุฒิภาวะของกระบวนการ (Process Maturity: PMAT)

วุฒิภาวะของกระบวนการกำหนดจากระดับ CMM (Capability Maturity Model)

การมีแบบอย่างมาก่อนและความยืดหยุ่นในการพัฒนามีผลโดยตรงต่อโครงการและเป็นสิ่งที่ควบคุมไม่ได้ แต่สถาปัตยกรรมและการขจัดความเสี่ยง ความเชื่อมแน่นของทีม และวุฒิภาวะของกระบวนการนั้นสามารถควบคุมได้

สำหรับตัวขับต้นทุน 17 ตัวนั้นสามารถแบ่งออกเป็น 4 หมวด คือปัจจัยด้านผลิตภัณฑ์ (Product Factors) ปัจจัยด้านแพลตฟอร์ม (Platform Factors) ปัจจัยด้านบุคลากร (Personnel Factors) และปัจจัยด้านโครงการ (Project Factors) โดยมีรายละเอียดดังนี้

1. ปัจจัยด้านผลิตภัณฑ์

1.1 ความเชื่อถือได้ของซอฟต์แวร์ที่ต้องการ (Required Software Reliability: RELY)

ถ้าความขัดข้องของซอฟต์แวร์ส่งผลแค่ก่อให้เกิดความไม่สะดวกแล้ว ความเชื่อถือได้ของซอฟต์แวร์ที่ต้องการจะมีค่าต่ำมาก แต่ถ้าความขัดข้องนี้ส่งผลต่อความเสี่ยงในชีวิตมนุษย์แล้ว ความเชื่อถือได้ของซอฟต์แวร์ที่ต้องการจะมีค่าสูงมาก

1.2 ขนาดของฐานข้อมูล (Database Size: DATA)

ขนาดของฐานข้อมูลมีความเกี่ยวข้องกับความเพียรพยายามที่ต้องใช้เพื่อรวมและบำรุงรักษาข้อมูลที่จำเป็นสำหรับการทดสอบโปรแกรม

1.3 ความซับซ้อนของผลิตภัณฑ์ (Product Complexity: CPLX)

ความซับซ้อนแบ่งออกเป็น 5 ส่วน คือ การดำเนินการด้านควบคุม (Control Operations) การดำเนินการด้านคำนวณ (Computational Operations) การดำเนินการที่ขึ้นกับอุปกรณ์ (Device-Dependent Operations) การดำเนินการด้านการจัดการข้อมูล (Data Management Operations) และการดำเนินการด้านการจัดการส่วนต่อประสานกับผู้ใช้ (User Interface Management Operations)

1.4 การพัฒนาสำหรับความสามารถใช้ซ้ำ (Developed for Reusability: RUSE)

การพัฒนาสำหรับความสามารถใช้ซ้ำมีความเกี่ยวข้องกับความเพียรพยายามเพิ่มเติมที่ต้องใช้สำหรับสร้างส่วนโปรแกรมเพื่อใช้ซ้ำในปัจจุบันหรืออนาคต

1.5 การเข้ากันได้ของการสร้างเอกสารกับความจำเป็นของวัฏจักรชีวิต (Documentation Match to Life-Cycle Needs: DOCU)

การเข้ากันได้ของการสร้างเอกสารกับความจำเป็นของวัฏจักรชีวิตประเมินจากความเหมาะสมของการสร้างเอกสารในโครงการนั้นต่อความจำเป็นด้านเอกสารของวัฏจักรชีวิต

2. ปัจจัยด้านแพลตฟอร์ม

2.1 เงื่อนไขบังคับด้านเวลากระทำการ (Execution Time Constraint: TIME)

เงื่อนไขบังคับด้านเวลากระทำการอธิบายด้วยเปอร์เซ็นต์ของเวลากระทำการที่มีอยู่ซึ่งใช้โดยระบบหรือระบบย่อย

2.2 เงื่อนไขบังคับด้านหน่วยเก็บหลัก (Main Storage Constraint: STOR)

เงื่อนไขบังคับด้านหน่วยเก็บหลักอธิบายด้วยเปอร์เซ็นต์ของหน่วยเก็บที่มีอยู่ซึ่งใช้โดยระบบหรือระบบย่อย

2.3 การเปลี่ยนแปลงของแพลตฟอร์ม (Platform Volatility: PVOL)

แพลตฟอร์มในที่นี้เน้นความซับซ้อนของฮาร์ดแวร์และซอฟต์แวร์ที่ผลิตภัณฑ์ซอฟต์แวร์ใช้ในการปฏิบัติงาน เช่น ถ้าพัฒนาเบราว์เซอร์เชิงข้อความสำหรับเครือข่าย (Network Text Browser) แพลตฟอร์มจะหมายถึงเครือข่าย ฮาร์ดแวร์ ระบบปฏิบัติการ และที่เก็บสารสนเทศแบบกระจาย (Distributed Information Repositories) ถ้าแพลตฟอร์มนี้มีการเปลี่ยนครั้งใหญ่ทุกสองสัปดาห์แล้ว การเปลี่ยนแปลงของแพลตฟอร์มจะมีค่าสูงมาก

3. <u>ปัจจัยด้านบุคลากร</u>

3.1 ความสามารถของนักวิเคราะห์ (Analyst Capability: ACAP)

ความสามารถของนักวิเคราะห์พิจารณาจากความสามารถ ประสิทธิภาพ ความละเอียดถี่ถ้วน และความสามารถในการสื่อสารและร่วมมือ

3.2 ความสามารถของนักเขียนโปรแกรม (Programmer Capability: PCAP)

ความสามารถของนักเขียนโปรแกรมเน้นความสามารถของนักเขียนโปรแกรมในเชิงทีมมากกว่ารายบุคคล ปัจจัยหลักที่พิจารณาคือ ความสามารถ ประสิทธิภาพ ความละเอียดถี่ถ้วน และความสามารถในการสื่อสารและร่วมมือ

3.3 ประสบการณ์ในโปรแกรมประยุกต์ (Applications Experience: AEXP)

ประสบการณ์ในโปรแกรมประยุกต์หมายถึงระดับประสบการณ์ของผู้พัฒนาในโปรแกรมประยุกต์ประเภทเดียวกับที่จะพัฒนา

3.4 ประสบการณ์ด้านแพลตฟอร์ม (Platform Experience: PEXP)

ประสบการณ์ด้านแพลตฟอร์มหมายถึงความเข้าใจในการใช้แพลตฟอร์มที่มีประสิทธิภาพ รวมทั้งความสามารถในส่วนต่อประสานกับผู้ใช้แบบกราฟิก ฐานข้อมูล เครือข่าย และมิดเดิลแวร์แบบกระจาย (Distributed Middleware)

3.5 ประสบการณ์ด้านภาษาและเครื่องมือ (Language and Tool Experience: LTEX)

ประสบการณ์ด้านภาษาและเครื่องมือหมายถึงระดับประสบการณ์ด้านภาษาโปรแกรมและเครื่องมือที่ใช้ของผู้พัฒนา

3.6 ความต่อเนื่องของบุคลากร (Personnel Continuity: PCON)

ความต่อเนื่องของบุคลากรวัดจากการหมุนเวียน (Turnover) รายปีของบุคลากร

4. <u>ปัจจัยด้านโครงการ</u>

4.1 การใช้เครื่องมือของซอฟต์แวร์ (Use of Software Tools: TOOL)

การใช้เครื่องมือของซอฟต์แวร์จะมีค่าต่ำมากถ้าเครื่องมือมีลักษณะลงรหัส (Code) หรือแก้ไขอย่างง่าย แต่จะมีค่าสูงมากถ้าเครื่องมือสามารถจัดการวัฏจักรการพัฒนาได้อย่างเบ็ดเสร็จ

4.2 การพัฒนาแบบหลายที่ตั้ง (Multisite Development: SITE)

การพัฒนาแบบหลายที่ตั้งพิจารณาจากสองปัจจัย คือ การอยู่ร่วมกันของที่ตั้งซึ่งมีตั้งแต่อยู่ร่วมกันอย่างเต็มที่จนถึงกระจายระหว่างประเทศ และการสนับสนุนด้านการสื่อสารซึ่งมีตั้งแต่การใช้อีเมลและโทรศัพท์อย่างผิวเผินไปจนถึงการใช้สื่อประสมเชิงโต้ตอบ (Interactive Multimedia) อย่างเต็มที่

4.3 กำหนดการของการพัฒนาที่ต้องการ (Required Development Schedule: SCED)

กำหนดการของการพัฒนาที่ต้องการหมายถึงเงื่อนไขบังคับของกำหนดการของผู้พัฒนาค่านี้จึงกำหนดจากเปอร์เซ็นต์การขยายหรือเร่งกำหนดการปกติ

ที่กล่าวมาข้างต้นเป็นรายละเอียดของตัวประกอบมาตราส่วนและตัวขับต้นทุน ซึ่งวิธีการระบุระดับของแต่ละตัวประกอบมาตราส่วนและตัวขับต้นทุนแสดงดังตารางที่ 2.1 2.2 และ 2.3

ตารางที่ 2.1 การระบุระดับของตัวประกอบมาตราส่วน

ตัวประกอบมาตราส่วน	ต่ำมาก	ต่ำ	ปานกลาง	สูง	สูงมาก	สูงพิเศษ
PREC	ไม่เคยทั้งหมด	ไม่เคยมาก	ค่อนข้างไม่เคย	คุ้นเคยโดยทั่วไป	คุ้นเคยมาก	คุ้นเคยโดยทั้งหมด
FLEX	เข้มงวด	ผ่อนปรนบางโอกาส	ผ่อนปรนบ้าง	สอดคล้องโดยทั่วไป	สอดคล้องบ้าง	เป้าหมายทั่วไป
RESL	เล็กน้อย (20%)	มีบ้าง (40%)	บ่อย (60%)	ทั่วไป (75%)	ส่วนมาก (90%)	ครบถ้วน (100%)
TEAM	โต้ตอบยากมาก	โต้ตอบยากบ้าง	โต้ตอบร่วมกันโดยพื้นฐาน	โต้ตอบร่วมกันโดยส่วนมาก	โต้ตอบร่วมกันสูง	โต้ตอบร่วมกันอย่างราบรื่น
PMAT	CMM ระดับ 1 ขั้นต่ำ	CMM ระดับ 1 ขั้นสูง	CMM ระดับ 2	CMM ระดับ 3	CMM ระดับ 4	CMM ระดับ 5

ตารางที่ 2.2 การระบุระดับของตัวขับต้นทุน

ตัวขับต้นทุน	ต่ำมาก	ต่ำ	ปานกลาง	สูง	สูงมาก	สูงพิเศษ
RELY	ไม่สะดวกเล็กน้อย	ต่ำและกู้ความสูญเสียได้ง่าย	ปานกลางและกู้ความสูญเสียได้ง่าย	สูญเสียทางการเงินสูง	เสี่ยงต่อชีวิตมนุษย์	
DATA		ขนาดฐานข้อมูลต่อจำนวนบรรทัดของโปรแกรมน้อยกว่า 10	ขนาดฐานข้อมูลต่อจำนวนบรรทัดของโปรแกรมตั้งแต่ 10 ถึง 100	ขนาดฐานข้อมูลต่อจำนวนบรรทัดของโปรแกรมตั้งแต่ 100 ถึง 1000	ขนาดฐานข้อมูลต่อจำนวนบรรทัดของโปรแกรมมากกว่า 1000	
CPLX	ดูตารางที่ 2.3					
RUSE		ไม่มี	ข้ามโครงการ	ข้ามโปรแกรม	ข้ามสายผลิตภัณฑ์	ข้ามหลายสายผลิตภัณฑ์
DOCU	ไม่ครอบคลุมความต้องการของวัฏจักรชีวิตมาก	ไม่ครอบคลุมความต้องการของวัฏจักรชีวิตบ้าง	ตรงกับที่วัฏจักรชีวิตต้องการ	เกินกว่าที่วัฏจักรชีวิตต้องการ	เกินกว่าที่วัฏจักรชีวิตต้องการมาก	
TIME			น้อยกว่า 50% ของเวลากระทำการที่มี	70%	85%	95%
STOR			น้อยกว่า 50% ของหน่วยเก็บหลักที่มี	70%	85%	95%
PVOL		เปลี่ยนครั้งใหญ่ทุก 12 เดือนหรือครั้งย่อยทุก 1 เดือน	ครั้งใหญ่ทุก 6 เดือนหรือครั้งย่อยทุก 2 สัปดาห์	ครั้งใหญ่ทุก 2 เดือนหรือครั้งย่อยทุก 1 สัปดาห์	ครั้งใหญ่ทุก 2 สัปดาห์หรือครั้งย่อยทุก 2 วัน	
ACAP	เปอร์เซ็นไทล์ที่ 15	เปอร์เซ็นไทล์ที่ 35	เปอร์เซ็นไทล์ที่ 55	เปอร์เซ็นไทล์ที่ 75	เปอร์เซ็นไทล์ที่ 90	

ตารางที่ 2.2 (ต่อ) การระบุระดับของตัวขับต้นทุน

ตัวขับต้นทุน	ต่ำมาก	ต่ำ	ปานกลาง	สูง	สูงมาก	สูงพิเศษ
PCAP	เปอร์เซ็นไทล์ที่ 15	เปอร์เซ็นไทล์ที่ 35	เปอร์เซ็นไทล์ที่ 55	เปอร์เซ็นไทล์ที่ 75	เปอร์เซ็นไทล์ที่ 90	
PCON	48%/ปี	24%/ปี	12%/ปี	6%/ปี	3%/ปี	
AEXP	≤ 2เดือน	6 เดือน	1 ปี	3 ปี	6 ปี	
PEXP	≤ 2เดือน	6 เดือน	1 ปี	3 ปี	6 ปี	
LTEX	≤ 2เดือน	6 เดือน	1 ปี	3 ปี	6 ปี	
TOOL	แก้ไข ลงรหัส และแก้จุดบกพร่อง	วิศวกรรมใช้คอมพิวเตอร์ช่วยอย่างง่าย (CASE) และรวมกันเล็กน้อย	เครื่องมือวัฏจักรชีวิตอย่างง่ายและรวมกันปานกลาง	เครื่องมือวัฏจักรชีวิตสมบูรณ์และโดดเด่นและรวมกันปานกลาง	เครื่องมือวัฏจักรชีวิตสมบูรณ์และเข้าควบคุมสถานการณ์และรวมกับกระบวนการวิธี การใช้ซ้ำเป็นอย่างดี	
SITE: การอยู่ร่วม	ระหว่างประเทศ	หลายเมืองและหลายบริษัท	หลายเมืองหรือหลายบริษัท	เมืองเดียวกัน	อาคารเดียวกัน	อยู่ร่วมกันอย่างเต็มที่
SITE: การสื่อสาร	อีเมลและโทรศัพท์บ้าง	แฟกซ์และโทรศัพท์รายบุคคล	อีเมลแบบแถบความถี่แคบ (Narrowband)	การสื่อสารอิเล็กทรอนิกส์แบบแถบความถี่กว้าง (Wideband)	การสื่อสารอิเล็กทรอนิกส์แบบแถบความถี่กว้างและประชุมทางวีดิทัศน์บางโอกาส	สื่อประสมเชิงโต้ตอบอย่างเต็มที่
SCED	75% จากเวลาปกติ	85%	100%	130%	160%	

ตารางที่ 2.3 การระบุระดับของ CPLX

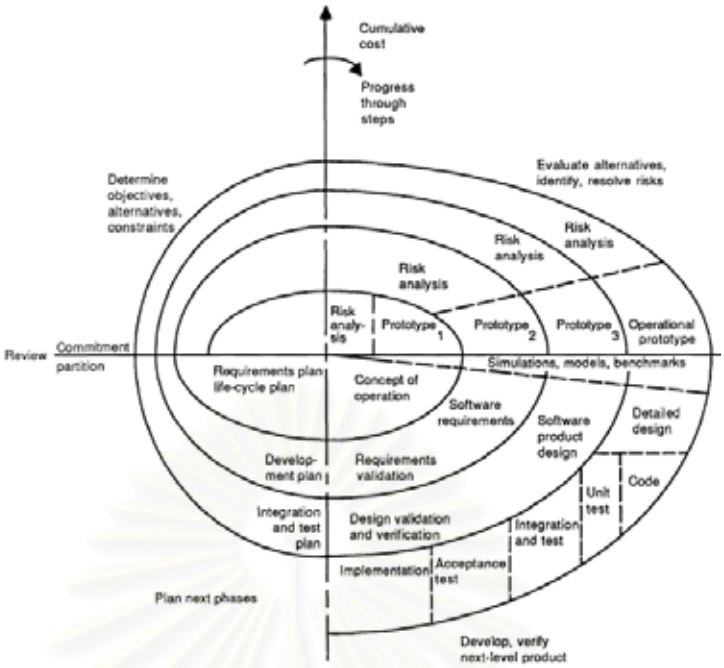
ระดับ	การดำเนินการด้านควบคุม	การดำเนินการด้านคำนวณ	การดำเนินการที่ขึ้นกับอุปกรณ์	การดำเนินการด้านการจัดการข้อมูล	การดำเนินการด้านการจัดการส่วนต่อประสานกับผู้ใช้
ต่ำมาก	รหัสคำสั่งมีตัวดำเนินการโครงสร้างไม่เป็นแบบกลุ่ม คือ Dos, CASEs และ IF-THEN-ELSEs จำนวนน้อย ประกอบมอดูลกันด้วยการเรียกกระบวนงาน (Procedure)	ประเมินค่าด้วยนิพจน์อย่างง่าย เช่น A=B+C*(D-E)	ข้อความสั่ง (Statement) อ่านและเขียนอย่างง่ายด้วยรูปแบบอย่างง่าย	แถวลำดับ (Array) อย่างง่ายในหน่วยความจำหลัก สอบถามและปรับฐานข้อมูลอย่างง่าย	รูปแบบการนำเข้าและสร้างรายการอย่างง่าย
ต่ำ	ทำกลุ่มของตัวดำเนินการในโปรแกรมอย่างง่าย โดยมากเป็นเพรดิเคต (Predicate) อย่างง่าย	ประเมินค่านิพจน์แบบปานกลาง เช่น D=SQRT(B**2-4.*A*C)	ไม่ต้องรู้ลักษณะตัวประมวลผลหรืออุปกรณ์ไอ/โอ และไอ/โอทำหน้าที่ระดับ GET/PUT	เป็นแฟ้มเดียว ความซับซ้อนของการสอบถามและปรับฐานข้อมูลปานกลาง	ใช้ตัวสร้างส่วนต่อประสานกราฟิกกับผู้ใช้อย่างง่าย
ปานกลาง	ส่วนใหญ่ทำกลุ่มอย่างง่าย มีตัวควบคุมระหว่างมอดูลบ้าง การส่งสาร (Message) อย่างง่าย	ใช้รูทีน (Routine) คณิตศาสตร์หรือสถิติมาตรฐาน การดำเนินการเวกเตอร์และเมทริกซ์อย่างง่าย	การประมวลผลไอ/โอมีการเลือกอุปกรณ์ ตรวจสอบสถานะและประมวลผลความผิดพลาด	นำเข้าหลายแฟ้มและนำออกแฟ้มเดียว การเปลี่ยนโครงการอย่างง่าย สอบถามและปรับฐานข้อมูลอย่างซับซ้อน	ใช้กลุ่มชิ้นส่วน (Widget) อย่างง่าย

ตารางที่ 2.3 (ต่อ) การระบุระดับของ CPLX

ระดับ	การดำเนินการด้านควบคุม	การดำเนินการด้านคำนวณ	การดำเนินการที่ขึ้นกับอุปกรณ์	การดำเนินการด้านการจัดการข้อมูล	การดำเนินการด้านการจัดการส่วนต่อประสานกับผู้ใช้
สูง	ตัวดำเนินการแบบกลุ่มพร้อมเพรดิเคตเชิงประกอบสูง การประมวลผลแบบกระจายเชิงเอกพันธ์ (Homogeneous)	การวิเคราะห์เชิงตัวเลขอย่างง่าย	ดำเนินการระดับไอ/โอกายภาพ	ทริกเกอร์ (Trigger) อย่างง่าย การจัดโครงสร้างข้อมูลใหม่ซับซ้อน	พัฒนาด้วยกลุ่มชิ้นส่วน และมีสื่อประสมอย่างง่าย
สูงมาก	รหัสคำสั่งแบบเรียกซ้ำ (Recursive) ประมวลผลแบบกระจายเชิงวิวิธพันธุ์ (Heterogeneous)	การวิเคราะห์เชิงตัวเลขอย่างยาก	รูทีนสำหรับวินิจฉัยสัญญาณขัดจังหวะ จัดการเส้นทางการสื่อสาร	ฐานข้อมูลแบบกระจาย ทริกเกอร์ที่ซับซ้อน	ระบบสองและสามมิติที่ซับซ้อน

2.3 กระบวนการเชิงวิวัฒน์ (Evolutionary Process)

ความต้องการของซอฟต์แวร์มีการเปลี่ยนแปลงบ่อยครั้งในระหว่างการพัฒนาทำให้ต้องการกระบวนการที่ออกแบบมาเพื่อผลิตภัณฑ์ที่เปลี่ยนแปลงตามเวลานี้ **(6)** ซึ่งกระบวนการวิวัฒน์ตอบสนองต่อจุดนี้ โดยมีหลักการคือ เริ่มต้นจากส่งมอบบางสิ่ง เช่น ต้นแบบ แก่ผู้ใช้ขั้นปลาย (End-User) วัดมูลค่าที่ผู้ใช้ได้รับจากสิ่งที่ส่งมอบ และปรับทั้งการออกแบบและจุดประสงค์ให้ตรงตามสิ่งที่สังเกตจากผู้ใช้ **(13)** และกระบวนการเชิงวิวัฒน์เป็นกระบวนการแบบทำซ้ำ (Iterative) ซึ่งเอื้อให้วิศวกรซอฟต์แวร์พัฒนาซอฟต์แวร์ที่มีความสมบูรณ์ขึ้นในแต่ละรอบของการพัฒนา **(6)**



รูปที่ 2.1 กระบวนการแบบเวียน **(14)**

ตัวอย่างของกระบวนการเชิงวิวัฒน์ เช่น การสร้างต้นแบบ (Prototyping) กระบวนการแบบเวียน (Spiral Process) การพัฒนาจวบกัน (Concurrent Development) แต่ในที่นี้จะกล่าวเฉพาะกระบวนการแบบเวียนดังรูปที่ 2.1 ซึ่งกระบวนการนี้จะพิจารณาความต้องการและทำการวนซ้ำทั้งขั้นตอน คือ วิเคราะห์ ออกแบบ สร้างและทดสอบมากกว่าหนึ่งครั้ง **(15)** ซึ่งกระบวนการนี้จะช่วยขจัดความเสี่ยงและสร้างบางส่วนของซอฟต์แวร์ซึ่งลูกค้าสามารถดูและให้ความเห็นได้ โดยกระบวนการแบบนี้เหมาะกับการพัฒนาซอฟต์แวร์ที่มีขนาดใหญ่เนื่องจากซอฟต์แวร์จะมีการพัฒนาเมื่อกระบวนการดำเนินการไปเรื่อย ๆ **(6)**

2.4 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องมี 2 ส่วนหลัก คือ งานวิจัยที่เกี่ยวข้องกับการจัดสรรทรัพยากรสำหรับโครงการเดียว และงานวิจัยที่เกี่ยวข้องกับการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ โดยมีรายละเอียดดังต่อไปนี้

2.4.1 <u>งานวิจัยด้านการจัดสรรทรัพยากรสำหรับโครงการเดียว</u>

การจัดสรรทรัพยากรในการพัฒนาซอฟต์แวร์โดยใช้หลักการมาตรวัดแบบเดิม เช่น การใช้แบบจำลอง COCOMO **(12)** แบบจำลอง Putnam **(16)** นั้น ยังขาดความสามารถในการสนับสนุน

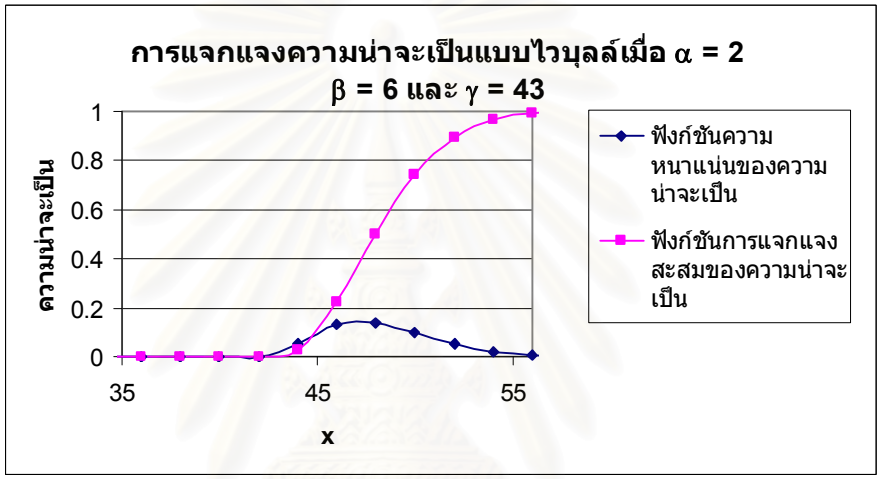
การตัดสินใจเชิงภาวะถ่วงดุล เช่น ถ้าแบบจำลองประมาณว่าจะต้องการพนักงาน 4 คนและใช้เวลา 2 ปีในการพัฒนาโครงการ แต่ถ้าในสถานการณ์จริงโครงการสามารถจ้างพนักงานได้เพียงแค่ 3 คน และเวลาถูกจำกัดที่ 1 ปี ผู้บริหารจะสามารถลดคุณภาพของซอฟต์แวร์ได้หรือไม่หรือจะต้องใช้พนักงานที่มีคุณภาพระดับใดสำหรับการพัฒนาซอฟต์แวร์ภายใต้เงื่อนไขจำกัดนี้ ดังนั้น Fenton และคณะ **(3)** จึงได้เสนอการจัดสรรทรัพยากรสำหรับโครงการซอฟต์แวร์โดยใช้โครงข่ายแบบเบย์ (Bayesian Networks) ซึ่งมีคุณสมบัติในการสนับสนุนการตัดสินใจเชิงภาวะถ่วงดุลแก่ผู้บริหารได้ โดยแต่ละบัพ (Node) ของโครงข่ายแทนทรัพยากรในโครงการซอฟต์แวร์ ดังนั้นผู้บริหารสามารถประมาณทรัพยากรได้โดยใส่เงื่อนไขที่มีอยู่ของทรัพยากร เช่น จำนวน Function Point ที่มีอยู่ และผู้บริหารสามารถรู้ผลกระทบของทรัพยากรนั้นที่มีต่อทรัพยากรอื่น ๆ โดยสังเกตค่าจากบัพที่แทนทรัพยากรอื่น ๆ ว่ามีการเปลี่ยนแปลงอย่างไร

อย่างไรก็ตามวิธีการนี้ยังขาดความสามารถบางประการ เนื่องจากในการหาคำตอบที่ต้องการผู้บริหารจะต้องกำหนดเหตุการณ์ที่ต้องการแล้วจึงหาคำตอบในแต่ละเหตุการณ์ ดังนั้นในกรณีที่ต้องการเปรียบเทียบเหตุการณ์จำนวนมากจะทำให้การหาคำตอบเกิดความซับซ้อนยุ่งยาก ดังนั้นงานวิจัยนี้จึงต้องการเสนอวิธีการที่สามารถแก้ปัญหาในจุดนี้โดยที่ยังคงความสามารถที่ดีของวิธีการเดิม คือ การสนับสนุนการวิเคราะห์เชิงภาวะถ่วงดุล โดยงานวิจัยนี้จะเสนอการใช้แบบจำลองประเมินความเสี่ยงสำหรับการจัดสรรทรัพยากรในโครงการซอฟต์แวร์ ซึ่งแบบจำลองประเมินความเสี่ยงมีหลายตัว เช่น แบบจำลองของ Nogueira และคณะ **(17-19)** ที่ใช้การแจกแจงความน่าจะเป็นแบบไวบุลล์ (Weibull's Family Distribution) แทนความเสี่ยงในโครงการซอฟต์แวร์ แบบจำลองของ Foo และคณะ **(20)** ที่ประเมินค่าความเสี่ยงของโครงการจากแบบสอบถาม แบบจำลองของ Madachy **(21)** ที่ประเมินความเสี่ยงโดยให้ผู้ใช้ระบุค่าตัวประกอบมาตราส่วนและตัวขับต้นทุนของ COCOMO II และกรอบงานของ Roy **(22)** ที่ประเมินความเสี่ยงโดยใช้แบบจำลองต้นไม้ความเสี่ยง (Risk Tree Model) ซึ่งสร้างจากค่าความน่าจะเป็นและระดับความรุนแรงของปัจจัยความเสี่ยงที่ประเมินโดยผู้ใช้ โดยแบบจำลองของ Nogueira และคณะจะเป็นแบบวัตถุวิสัย (Objective) แต่วิธีการที่เหลือจะเป็นแบบอัตวิสัย (Subjective) งานวิจัยนี้เลือกใช้แบบจำลองของ Nogueira และคณะเนื่องจากเป็นแบบวัตถุวิสัยและเป็นสมการคณิตศาสตร์อันเหมาะสมในการนำไปวิเคราะห์ค่าเพื่อหาคำตอบที่เหมาะที่สุดต่อไป

แบบจำลองประเมินความเสี่ยงของ Nogueira และคณะสร้างมาสำหรับกระบวนการเชิงวิวัฒน์ โดยแบบจำลองอยู่ในรูปการแจกแจงความน่าจะเป็นแบบไวบุลล์ซึ่งมีฟังก์ชันความหนาแน่นของความน่าจะเป็น (Probability Density Function: pdf) ดังสมการที่ 2.5 และมีฟังก์ชันการแจกแจงสะสมของความน่าจะเป็น (Cumulative Distribution Function: cdf) ดังสมการที่ 2.6 ตัวอย่างการแจกแจงความน่าจะเป็นแบบไวบุลล์ดังรูปที่ 2.2

$$f(x;\gamma,\alpha,\beta)=\begin{cases}0, x<\gamma \\ \dfrac{\alpha}{\beta^{\alpha}}(x-\gamma)^{\alpha-1}\exp^{-\left(\frac{x-\gamma}{\beta}\right)^{\alpha}}, x\geq\gamma\end{cases} \tag{2.5}$$

$$F(x;\gamma,\alpha,\beta)=\begin{cases}0, x<\gamma \\ 1-\exp^{-\left(\frac{x-\gamma}{\beta}\right)^{\alpha}}, x\geq\gamma\end{cases} \tag{2.6}$$



รูปที่ 2.2 ตัวอย่างการแจกแจงความน่าจะเป็นแบบไวบุลล์

โดยที่ x เป็นตัวแปรสุ่มแทนจำนวนเวลา (เดือน) ที่ใช้ในการพัฒนาซอฟต์แวร์ α เป็นตัวแปรที่กำหนดรูปร่างของการแจกแจงซึ่งใช้แสดงผลิตภาพของพนักงาน (Staff Productivity) β เป็นตัวแปรที่กำหนดความกว้างของการแจกแจงซึ่งใช้แสดงการเปลี่ยนแปลงของความต้องการ (Requirement Volatility) และ γ เป็นตัวแปรที่กำหนดการเลื่อนซ้ายขวาของการแจกแจงซึ่งใช้แสดงความซับซ้อนของซอฟต์แวร์ (Complexity) ซึ่งงานวิจัยของ Nogueira และคณะ **(19)** ได้หาค่าของตัวแปรเหล่านี้จากข้อมูลและพบว่าสามารถปรับค่าตัวแปรได้ดังนี้คือ α จะมีค่าเป็นอัตราส่วนระหว่างเปอร์เซ็นต์ของเวลาทำงานจริง (Direct Work Time) ต่อเปอร์เซ็นต์ของเวลาอยู่เปล่า (Idle Time) ส่วน β จะมีค่าเท่ากับ $INT((BR+DR)/10)$ โดยที่ INT เป็นฟังก์ชันหาเลขจำนวนเต็ม BR (Birth Rate) คือเปอร์เซ็นต์ของความต้องการใหม่ที่เกิดขึ้นในแต่ละรอบของกระบวนการพัฒนา และ DR (Death Rate) คือเปอร์เซ็นต์ของความต้องการที่ลูกค้าตัดทิ้งในแต่ละรอบของกระบวนการพัฒนาเช่นกัน ส่วน γ จะวัดจากมาตรวัด Large Granularity Complexity (LGC) ซึ่งเป็นมาตรวัดที่คำนวณ

จากตัวดำเนินการ (Operator) กระแสข้อมูล (Data Stream) และประเภท (Type) ของข้อกำหนดเชิงรูปนัย (Formal Specification) ในรูปแบบภาษาการออกแบบข้อกำหนดต้นแบบ (Prototyping Specification Design Language) นั่นคือ γ จะเท่ากับ 13·Ln(LGC)-82 โดยที่ LGC มีความสัมพันธ์กับ $KLOC$ ดังนี้คือ $KLOC$=(32·LGC+150)/1000 และค่าความน่าจะเป็นจากฟังก์ชันการแจกแจงสะสมดังสมการที่ 2.6 จะหมายถึงความน่าจะเป็นที่โครงการจะพัฒนาเสร็จสิ้นโดยใช้เวลาน้อยกว่าหรือเท่ากับ x เดือน โดยทั้งสองฟังก์ชันมีเงื่อนไขคือ $x \geq \gamma$ กล่าวคือ เวลาที่ใช้ในการพัฒนาจะมากกว่าหรือเท่ากับเวลาที่เกิดจากความซับซ้อนของซอฟต์แวร์

2.4.2 <u>งานวิจัยด้านการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ</u>

สำหรับวิธีการจัดสรรทรัพยากรสำหรับพัฒนาซอฟต์แวร์หลาย ๆ โครงการพร้อมกันที่พิจารณาปัจจัยด้านความเสี่ยงด้วยนั้น งานวิจัยของ Fewster และคณะ **(5)** ได้ใช้เส้นตายและการแจกแจงความน่าจะเป็นของความเพียรพยายามสำหรับหาค่าความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จและกำหนดเส้นตายใหม่ โดยเงื่อนไขที่ใช้ในการกำหนดความสำเร็จของกลุ่มโครงการคือ โครงการที่ 1 ต้องเสร็จก่อนเส้นตายของโครงการที่ 1 และโครงการที่ 1 และ 2 ต้องเสร็จก่อนเส้นตายของโครงการที่ 2 และอื่น ๆ โดยที่เส้นตายที่ 1 น้อยกว่าเส้นตายที่ 2 ซึ่งน้อยกว่าเส้นตายที่ 3 และอื่น ๆ ถ้ามีพนักงานจำนวน E คน ความเพียรพยายามทั้งหมด (คน-วัน) ตั้งแต่เริ่มโครงการจนถึงเส้นตาย d วันคือ E·d ซึ่งทำให้เงื่อนไขของกลุ่มโครงการเป็นดังสมการที่ 2.7

$$\begin{pmatrix} Y_1 \\ Y_1+Y_2 \\ \vdots \\ Y_1+Y_2+\ldots+Y_n \end{pmatrix} \leq E \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} \qquad (2.7)$$

n แทนจำนวนของโครงการ Y_i แทนความเพียรพยายามที่ต้องการสำหรับโครงการที่ i ถ้า $Y_1+\ldots+Y_i > E \cdot d_i$ แล้วจะถือว่าเส้นตาย d_i ไม่เหมาะสม การคำนวณความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จหาได้จากการจำลองสถานการณ์ (Simulation) โดยการจำลองสถานการณ์จะสร้างค่า $Y_1, \ldots, Y_n$ แบบสุ่มตามการแจกแจงความน่าจะเป็นของความเพียรพยายามที่ได้กำหนดไว้ และถ้าค่า $Y_1, \ldots, Y_n$ นั้นตรงตามเงื่อนไขสมการ 2.7 จะถือว่าชุด Y นั้นเหมาะสม และความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจะเท่ากับอัตราส่วนของจำนวนชุด Y ที่เหมาะสมต่อจำนวนรอบทั้งหมดของการจำลองสถานการณ์

ถึงแม้ว่าวิธีการนี้จะช่วยสนับสนุนการตัดสินใจได้ดี แต่วิธีการยังมีข้อจำกัดคือ จำนวนปัจจัยที่ใช้ในการปรับปรุงความน่าจะเป็นที่โครงการจะสำเร็จมีแค่สองตัว คือ จำนวนพนักงานและเส้นตาย ซึ่งค่อนข้างน้อย ทำให้จำนวนทางเลือกในการตัดสินใจถูกจำกัด ดังนั้นงานวิจัยนี้จึงได้เสนอการใช้แบบจำลองต้นทุนคือ COCOMO II ผนวกกับวิธีการของ Fewster และคณะเพื่อเพิ่มตัวเลือกด้านทรัพยากรสำหรับการตัดสินใจจัดสรรทรัพยากรในกลุ่มโครงการโดยประยุกต์ใช้วิธีการ NOSTROMO (National Obscurity Statistical Risk Observation Model) **(23)** สำหรับการผนวกนี้

หลักการของ NOSTROMO คือจะสมมุติระดับความไม่แน่นอนของแต่ละปัจจัยใน COCOMO II ที่ประกอบด้วยตัวประกอบมาตราส่วน 5 ตัว ตัวขับต้นทุน 17 ตัว และขนาดของซอฟต์แวร์ โดยแบ่งความไม่แน่นอนออกเป็น 4 ระดับ คือ ระดับที่ 1 หมายถึงไม่มีความไม่แน่นอน ระดับที่ 2 หมายถึงความไม่แน่นอนต่ำ ระดับที่ 3 หมายถึงความไม่แน่นอนปานกลาง และระดับที่ 4 หมายถึงความไม่แน่นอนสูง โดยใช้การแจกแจงความน่าจะเป็นแสดงถึงระดับความไม่แน่นอนเหล่านี้ กล่าวคือค่าที่มีระดับความไม่แน่นอนสูงจะแทนด้วยการแจกแจงความน่าจะเป็นที่มีความแปรปรวนสูง ดังนั้นค่าคงที่แสดงถึงการไม่มีความไม่แน่นอน การแจกแจงแบบปรกติ (Normal Distribution) แสดงถึงความไม่แน่นอนต่ำ การแจกแจงสามเหลี่ยม (Triangular Distribution) แสดงถึงความไม่แน่นอนปานกลาง และการแจกแจงเอกรูป (Uniform Distribution) แสดงถึงความไม่แน่นอนสูง และค่าช่วงของแต่ละปัจจัยกำหนดตามความไม่แน่นอน ยกเว้นขนาดของซอฟต์แวร์ที่กำหนดตามเฟสการพัฒนา (Development Phase) ดังตารางที่ 2.4 **(24)** หลังจากนั้นใช้การจำลองสถานการณ์มอนติคาร์โล (Monte Carlo Simulation) เพื่อหาการแจกแจงความน่าจะเป็นของความเพียรพยายามและเวลา

ตารางที่ 2.4 ความไม่แน่นอนของขนาดของซอฟต์แวร์ตามเฟสการพัฒนา (X คือ ขนาดของซอฟต์แวร์ในหน่วย KLOC)

เฟส	ขอบเขตล่าง	ขอบเขตบน
Feasibility	$0.25X$	$4X$
Concept of operation	$0.5X$	$2X$
Requirements specification	$0.667X$	$1.5X$
Product design specification	$0.8X$	$1.25X$
Detail design specification	$0.91X$	$1.1X$
Development and test	X	X

บทที่ 3

แนวคิดการวิจัย

งานวิจัยนี้แบ่งออกเป็น 2 ส่วนหลัก คือ การจัดสรรทรัพยากรสำหรับโครงการเดียวและการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ ซึ่งในแต่ส่วนมีแนวคิดของการวิจัยดังนี้

3.1 แนวคิดการจัดสรรทรัพยากรสำหรับโครงการเดียว

งานวิจัยนี้เสนอการใช้แบบจำลองประเมินความเสี่ยงเพื่อช่วยในการตัดสินใจด้านการจัดสรรทรัพยากรในการพัฒนาซอฟต์แวร์ ซึ่งการใช้แบบจำลองนี้ช่วยเพิ่มความสามารถในการหาคำตอบที่เหมาะที่สุดภายใต้เงื่อนไขได้โดยง่าย และยังคงคุณสมบัติการวิเคราะห์ทรัพยากรเชิงภาวะถ่วงดุลในการช่วยสนับสนุนการตัดสินใจแก่ผู้บริหารไว้เช่นเดิม โดยแบบจำลองประเมินความเสี่ยงที่เลือกใช้นั้น คือ แบบจำลองประเมินความเสี่ยงของ Nogueira และคณะ เนื่องจากแบบจำลองนี้คำนึงถึงความน่าจะเป็นที่โครงการจะเสร็จสิ้นและสนับสนุนการตัดสินใจเชิงภาวะถ่วงดุลได้ นอกจากนี้ค่าของแบบจำลองมีลักษณะต่อเนื่องจึงทำให้ไม่ต้องการการแบ่งช่วงของตัวเลขที่ทำให้สูญเสียความแม่นยำ โดยผู้บริหารสามารถที่จะกำหนดมาตราที่ต้องการใช้ได้เองตามความเหมาะสม และค่าของแบบจำลองเป็นเชิงวัตถุวิสัยทำให้สามารถถูกเก็บและประมวลผลโดยอัตโนมัติ

แบบจำลองประเมินความเสี่ยงนอกจากช่วยในการจัดสรรทรัพยากรเชิงภาวะถ่วงดุลนี้แล้ว ยังสามารถช่วยในการหาคำตอบที่เหมาะที่สุดในกรณีที่ปัญหานั้นมีความซับซ้อนได้โดยง่าย โดยแปลงแบบจำลองประเมินความเสี่ยงในรูปแบบกำหนดการแบบไม่เชิงเส้น (Nonlinear Programming) เพื่อหารูปแบบการจัดสรรทรัพยากรในโครงการที่เหมาะที่สุดภายใต้ข้อจำกัดด้านทรัพยากรต่าง ๆ ซึ่งจะสังเกตได้ว่าการจัดสรรทรัพยากรในกรณีที่ไม่มีปัญหาในข้อจำกัดด้านทรัพยากรนั้นไม่มีความยุ่งยาก คือ ถ้าต้องการให้ความน่าจะเป็นที่โครงการจะเสร็จสิ้นสูงย่อมจะต้องจัดสรรให้เวลาและผลิตภาพของพนักงานนั้นมีค่ามากที่สุด และความซับซ้อนของซอฟต์แวร์และการเปลี่ยนแปลงของความต้องการมีค่าน้อยที่สุดนั่นเอง อย่างไรก็ตามในความเป็นจริงทรัพยากรเหล่านี้มีจำกัดและมีความเกี่ยวข้องสัมพันธ์กัน ทำให้ไม่สามารถลดทรัพยากรหนึ่งโดยที่ไม่ส่งผลกับทรัพยากรอื่นได้ ดังนั้นวิธีการนี้ได้ผนวกต้นทุนของทรัพยากรเข้าไปในปัญหาด้วย นอกเหนือจากข้อจำกัดทางด้านทรัพยากร ทำให้เป้าหมายของปัญหานี้คือจะจัดสรรทรัพยากรอย่างไรเพื่อที่จะให้ผลตอบแทนมากที่สุดภายใต้เงื่อนไขด้านทรัพยากรที่กำหนด ตัวอย่างเช่น ภายใต้

ข้อจำกัดด้านความซับซ้อนของซอฟต์แวร์และการเปลี่ยนแปลงของความต้องการ ผู้บริหารควรจะจัดสรรเวลาและผลิตภาพของพนักงานอย่างไรเพื่อให้ผลตอบแทนสูงสุด

การทดลองนี้ได้จำลองปัญหาหนึ่งมาเป็นตัวอย่างซึ่งผู้ใช้สามารถจำลองปัญหาของผู้ใช้ให้เป็นไปในรูปแบบที่ต้องการได้โดยการประยุกต์ใช้ขั้นตอนดังที่จะกล่าว ตัวอย่างของปัญหาในการทดลองนี้ คือ สมมุติว่าค่าใช้จ่ายสำหรับพนักงานเท่ากับ s บาท/เดือน ต่อผลิตภาพของพนักงาน (α) ที่มีค่าเป็น 1 และการเพิ่มของผลิตภาพทำให้ต้นทุนด้านพนักงานเพิ่มแบบเชิงเส้น กล่าวคือ ไม่มีค่าใช้จ่ายพิเศษสำหรับทำงานล่วงเวลา เป็นต้น ดังนั้นต้นทุนด้านพนักงานสำหรับการพัฒนาที่ใช้เวลา x เดือนจะเป็นดังสมการที่ 3.1

$$cost = s \cdot x \cdot \alpha \tag{3.1}$$

และสมมุติว่าเมื่อโครงการนี้เสร็จจะได้รับค่าตอบแทนเท่ากับ i แต่เนื่องจากความสำเร็จของโครงการมีความไม่แน่นอนดังนั้นค่าตอบแทนคาดหมาย (Expected Income) เมื่อความน่าจะเป็นที่โครงการจะเสร็จสิ้นด้วยความน่าจะเป็น p จะเป็นดังสมการที่ 3.2

$$expin = p \cdot i \tag{3.2}$$

และค่าตอบแทนคาดหมายสุทธิของโครงการจะเป็นดังสมการที่ 3.3

$$E(return) = expin - cost \tag{3.3}$$

หลังจากนั้นจะแปลงปัญหาเป็นกำหนดการแบบไม่เชิงเส้นเพื่อหาคำตอบที่เหมาะที่สุดภายใต้สถานการณ์ทรัพยากรที่จำกัด ตัวอย่างเช่น โครงการนี้มีพนักงาน 3 คน แต่ละคนมีเงินเดือน 25,000 บาท/เดือน ทำให้ค่าใช้จ่ายด้านพนักงานเป็น 75,000 บาท/เดือน และถ้าโครงการนี้เสร็จสิ้นจะได้รับค่าตอบแทน 1,000,000 บาท แต่มีข้อจำกัดที่โครงการจะต้องเสร็จภายใน 12 เดือน และคาดว่าการเปลี่ยนแปลงของความต้องการเป็น 6 และความซับซ้อนของซอฟต์แวร์เป็น 0 เดือนเนื่องจากการพัฒนาอยู่ในรอบแรก ดังนั้นค่าตอบแทนคาดหมายสุทธิของโครงการจะเป็นดังสมการที่ 3.4

$$E(return) = 1{,}000{,}000\left(1 - \exp^{-\left(\frac{x}{6}\right)^{\alpha}}\right) - 75{,}000x\alpha, x \geq 0 \quad (3.4)$$

และถ้าต้องการรู้ว่าควรจะให้ผลิตภาพของพนักงานและเวลาที่ใช้พัฒนาเป็นเท่าใดจึงจะทำให้ได้ค่าตอบแทนมากที่สุดสามารถกระทำได้โดยจำลองปัญหานี้ให้อยู่ในรูปกำหนดการแบบไม่เชิงเส้น คือ

Maximize $E(return)$

Subject to

$0.8 \leq \alpha \leq 6$

$0 \leq x \leq 12$

หลังจากนั้นจะหาคำตอบของกำหนดการแบบไม่เชิงเส้นนี้ ซึ่งค่าคำตอบที่ได้จะเป็นการจัดสรรทรัพยากรที่เหมาะที่สุดสำหรับปัญหาในการจัดสรรทรัพยากรนี้ เช่น คำตอบที่ได้จากตัวอย่าง คือ ผลิตภาพของพนักงานและเวลาในการพัฒนาที่ควรใช้เพื่อให้ได้ผลตอบแทนสูงสุดภายใต้ข้อจำกัดที่โครงการต้องเสร็จภายใน 12 เดือน

3.2 แนวคิดการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ

การจัดการซอฟต์แวร์แบบกลุ่มโครงการนั้นคือการจัดการที่คำนึงถึงผลรวมของทุกโครงการ ซึ่งในการตัดสินใจที่จะทำการพัฒนาซอฟต์แวร์นั้นการประมาณต้นทุนของซอฟต์แวร์สามารถใช้เป็นเงื่อนไขในการจัดการแบบหลายโครงการได้ โดยมีเงื่อนไขคือทรัพยากรที่มีอยู่ควรเพียงพอกับทรัพยากรที่ประมาณได้ของโครงการ ซึ่งวิธีการที่ใช้การประมาณต้นทุนของซอฟต์แวร์เพื่อหาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้แก่ งานวิจัยของ Fewster และคณะซึ่งได้ใช้เส้นตายสำหรับการจัดการกลุ่มโครงการ พร้อมทั้งใช้การแจกแจงความน่าจะเป็นของความเพียรพยายามแทนความไม่แน่นอนของความเพียรพยายามที่เหมาะสมกับการพัฒนาสำหรับใช้ในการหาค่าความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ

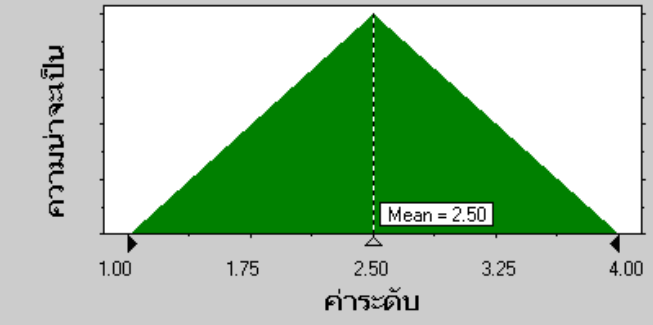
อย่างไรก็ตามวิธีการของ Fewster และคณะจะปรับเปลี่ยนความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จโดยการปรับเพียงแค่สองปัจจัย คือ เส้นตายและจำนวนพนักงานเท่านั้น งานวิจัยนี้จึงเสนอการใช้แบบจำลองต้นทุนมาช่วยแก้ปัญหาในจุดนี้ กล่าวคือ ช่วยให้สามารถปรับเปลี่ยนปัจจัยอื่น ๆ นอกจากเส้นตายและจำนวนพนักงานได้ เช่น เครื่องมือ คุณภาพของบุคลากร และรูปแบบการสื่อสาร เป็นต้น สำหรับปรับปรุงความน่าจะเป็นที่กลุ่มโครงการพัฒนาซอฟต์แวร์จะสำเร็จ โดย

แบบจำลองต้นทุนที่ใช้ในงานวิจัยนี้คือ COCOMO II เนื่องจากเป็นแบบจำลองมาตรฐานที่นิยมใช้และยังประกอบไปด้วยปัจจัยในการพัฒนาซอฟต์แวร์ที่น่าสนใจหลายตัวซึ่งสามารถช่วยเพิ่มทางเลือกในการตัดสินใจแก่ผู้บริหารได้

ตามวิธีการของ Fewster จะกำหนดการแจกแจงความน่าจะเป็นของความเพียรพยายาม (*Y*) จากผู้เชี่ยวชาญหรือข้อมูลเก่าในอดีต งานวิจัยนี้จะประยุกต์ใช้ NOSTROMO ในการหาการแจกแจงความน่าจะเป็นของความเพียรพยายามและเวลาในการพัฒนา โดยมีหลักการคือ ใช้ปัจจัยหลักที่สำคัญเพียง 23 ตัวของ COCOMO II คือ ตัวประกอบมาตราส่วน 5 ตัว ตัวขับต้นทุน 17 ตัวและขนาดของซอฟต์แวร์ หลังจากนั้นกำหนดการแจกแจงความน่าจะเป็นตามวิธีของ NOSTROMO พร้อมทั้งกำหนดค่าพารามิเตอร์ที่จำเป็นสำหรับการแจกแจงความน่าจะเป็น คือ ค่ากลาง (Center) ค่าต่ำสุด (Minimum) และค่าสูงสุด (Maximum) โดยค่าของพารามิเตอร์เหล่านี้กำหนดจากตัวเลขซึ่งแปลงมาจากค่าระดับของปัจจัยใน COCOMO II ที่ผู้ใช้ประเมินให้ดังตารางที่ 3.1 หลังจากนั้นใช้การจำลองสถานการณ์มอนติคาร์โลสร้างการแจกแจงของความเพียรพยายามและเวลา ตัวอย่างเช่น สมมุติให้ปัจจัยที่พิจารณาคือ TOOL ซึ่งผู้ใช้ประเมินว่ามีความไม่แน่นอนปานกลาง ค่าของ TOOL อยู่ในระดับต่ำ และความคลาดเคลื่อนของการประเมินเท่ากับ 1 ดังนั้นรูปแบบของการแจกแจงความน่าจะเป็นคือสามเหลี่ยม และความคลาดเคลื่อนของการประเมินเท่ากับ 1 หมายถึงค่าระดับจะอยู่ภายใน ค่ากลาง±1 ทำให้ค่าต่ำสุดและค่าสูงสุดของค่าระดับเท่ากับ ค่ากลาง-1 และ ค่ากลาง+1 ตามลำดับ ในตัวอย่างนี้ค่าของ TOOL อยู่ในระดับต่ำ ดังนั้นค่ากลางจะเป็นระดับต่ำ (ตามค่าระดับ) ค่าต่ำสุดและค่าสูงสุดจะเป็นต่ำมาก (ต่ำ-1) และปานกลาง (ต่ำ+1) ตามลำดับ หลังจากกำหนดค่าของ TOOL แล้วก็จะกำหนดเช่นนี้ไปเรื่อย ๆ จนครบทุกปัจจัย ตัวอย่างการแจกแจงความน่าจะเป็นของ TOOL แสดงดังรูปที่ 3.1

ตารางที่ 3.1 ตัวเลขแทนค่าระดับของปัจจัยใน COCOMO II

ค่าระดับ	ค่าต่ำสุด	ค่ากลาง	ค่าสูงสุด
ต่ำพิเศษ	0	0.5	1
ต่ำมาก	1	1.5	2
ต่ำ	2	2.5	3
ปานกลาง	3	3.5	4
สูง	4	4.5	5
สูงมาก	5	5.5	6
สูงพิเศษ	6	6.5	7



รูปที่ 3.1 ตัวอย่างการแจกแจงความน่าจะเป็นของ TOOL

หลังจากนั้นจะนำการแจกแจงความน่าจะเป็นของความเพียรพยายามและเวลามาหาความน่าจะเป็นที่โครงการจะสำเร็จโดยใช้เงื่อนไขที่กำหนด โดยเงื่อนไขที่ใช้นี้จะใช้เช่นเดียวกับเงื่อนไขที่ได้กล่าวไปแล้วของ Fewster แต่เนื่องจากหน่วยเวลาของเงื่อนไขเดิมเป็นวันแต่ค่าที่จะใช้เป็นเดือน ดังนั้นจะแปลงสมการให้อยู่ในหน่วยเดือนได้ดังสมการที่ 3.5

$$\begin{pmatrix} X_1 \\ X_1 + X_2 \\ \vdots \\ X_1 + X_2 + \ldots + X_n \end{pmatrix} \leq E \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix} \tag{3.5}$$

โดยที่ n เป็นจำนวนของโครงการ X_i เป็นความเพียรพยายามในหน่วยเดือนและ m_i เป็นเส้นตายของโครงการ i นอกจากนี้ COCOMO II ยังให้ผลลัพธ์ของจำนวนเวลาที่ใช้ในการพัฒนานอกจากความเพียรพยายามด้วยดังนั้นในงานวิจัยนี้จึงแปลงสมการที่ 3.5 ให้อยู่ในรูปเวลาจะได้สมการที่ 3.6

$$\begin{pmatrix} T_1 \\ T_1 + T_2 \\ \vdots \\ T_1 + T_2 + \ldots + T_n \end{pmatrix} \leq \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix} \tag{3.6}$$

โดยที่ n เป็นจำนวนของโครงการ T_i เป็นจำนวนเดือนที่ใช้ในการพัฒนา m_i เป็นเส้นตายของโครงการ i สังเกตได้ว่าสมการนี้จะไม่มีค่า E เนื่องจาก COCOMO II สมมุติจำนวนพนักงาน

เป็นค่าจำนวนพนักงานที่ควรใช้โดยทั่วไป **(15)** โดยการทดลองจะใช้ทั้งสองเงื่อนไขนี้คือทั้งเงื่อนไขความเพียรพยายามและเงื่อนไขเวลา

สำหรับการทดลองปรับเปลี่ยนปัจจัยใน COCOMO II เพื่อดูว่าการปรับเปลี่ยนปัจจัยอื่น ๆ นอกจากจำนวนพนักงานและเส้นตายนั้นส่งผลต่อกลุ่มโครงการหรือไม่ จะศึกษาว่าถ้ากำหนดให้ระดับของปัจจัยใน COCOMO II นั้นแตกต่างกันแล้วเช่น มีค่ามากที่สุดและต่ำที่สุด ค่าความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จสำหรับปัจจัยที่มีระดับแตกต่างจะมีการเปลี่ยนแปลงไปอย่างไร

นอกจากการศึกษาการปรับเปลี่ยนความสำเร็จของกลุ่มโครงการโดยการปรับเปลี่ยนปัจจัยต่าง ๆ และศึกษาการสร้างแบบจำลองโดยใช้เงื่อนไขความเพียรพยายามและเวลาแล้ว งานวิจัยนี้ยังได้ทดลองประโยชน์ที่ได้รับจากการบริหารแบบกลุ่มโครงการ การบริหารแบบกลุ่มโครงการโดยปกติจะช่วยควบคุมความไม่แน่นอนและความเสี่ยงได้ดีเนื่องจากผู้บริหารสามารถนำทรัพยากรจากหลายโครงการมาใช้ร่วมกันได้ ดังนั้นโครงการที่มีความไม่แน่นอนสูงควรจะได้รับประโยชน์จากการบริหารแบบกลุ่มโครงการมากกว่าโครงการที่มีความไม่แน่นอนต่ำ ประโยชน์ที่ได้นี้สามารถวัดได้จากอัตราส่วนของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จโดยการบริหารแบบกลุ่มโครงการต่อความน่าจะเป็นที่จะสำเร็จโดยการบริหารโครงการอย่างเป็นอิสระต่อกัน (Individual Project) ดังตารางที่ 3.2 โดยที่ l_i เป็นความน่าจะเป็นที่กลุ่มโครงการที่มีความไม่แน่นอนต่ำจะสำเร็จโดยใช้การบริหารโครงการอย่างเป็นอิสระต่อกัน กล่าวคือโครงการที่ 1 2 3 4 5 6 7 และ 8 สำเร็จทั้งหมด และ l_p เป็นความน่าจะเป็นที่กลุ่มโครงการที่มีความไม่แน่นอนต่ำจะสำเร็จโดยใช้การบริหารแบบกลุ่มโครงการ ส่วน h_i และ h_p เป็นความน่าจะเป็นที่กลุ่มโครงการที่มีความไม่แน่นอนสูงจะสำเร็จโดยใช้การบริหารโครงการอย่างเป็นอิสระต่อกันและการบริหารแบบกลุ่มโครงการตามลำดับ ตามสมมุติฐานที่ได้กล่าว $ratio_h$ ควรมากกว่า $ratio_l$

ตารางที่ 3.2 การทดลองประโยชน์ของการบริหารแบบกลุ่มโครงการ

ความน่าจะเป็นที่จะสำเร็จ	โครงการเป็นอิสระต่อกัน	กลุ่มโครงการ	ประโยชน์
ความไม่แน่นอนต่ำ	l_i	l_p	$ratio_l=l_p/l_i$
ความไม่แน่นอนสูง	h_i	h_p	$ratio_h=h_p/h_i$

สมการที่ 3.7 แสดงเงื่อนไขสำหรับการบริหารโครงการอย่างเป็นอิสระต่อกันโดยใช้เงื่อนไขความเพียรพยายามและสมการที่ 3.8 แสดงเงื่อนไขสำหรับการบริหารโครงการอย่างเป็นอิสระต่อกันโดยใช้เงื่อนไขเวลา

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \leq E \begin{pmatrix} m_1 \\ m_2 - m_1 \\ \vdots \\ m_n - m_{n-1} \end{pmatrix} \tag{3.7}$$

$$\begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{pmatrix} \leq \begin{pmatrix} m_1 \\ m_2 - m_1 \\ \vdots \\ m_n - m_{n-1} \end{pmatrix} \tag{3.8}$$

เนื่องจากผู้บริหารสามารถปรับเปลี่ยนปัจจัยด้านทรัพยากรนอกจากเส้นตายและจำนวนพนักงานเพื่อปรับเปลี่ยนค่าความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ โดยปัจจัยด้านทรัพยากรเหล่านี้ เช่น ความสามารถในการสื่อสาร ความสามารถของบุคลากร เป็นต้น อาจปรับเปลี่ยนความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากน้อยแตกต่างกัน และความเข้าใจนี้อาจช่วยผู้บริหารในการเลือกปัจจัยต่าง ๆ ที่เหมาะสมได้ ผู้วิจัยจึงทดลองและวิเคราะห์ความไวของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จต่อแต่ละปัจจัยด้านทรัพยากร เพื่อดูว่าแต่ละปัจจัยมีผลต่อการเปลี่ยนแปลงของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากน้อยเพียงใด โดยในที่นี้จะสมมุติว่าผู้บริหารเลือกใช้ปัจจัยนั้นสำหรับทุกโครงการในกลุ่มโครงการ เช่น สมมุติว่ากลุ่มโครงการประกอบด้วยโครงการ 8 โครงการ ถ้าผู้บริหารเลือกใช้ TOOL เป็นปัจจัยที่ปรับเปลี่ยนความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ ดังนั้นโครงการทั้ง 8 โครงการนี้จะถูกปรับด้วยค่า TOOL เหมือนกัน

โดยการวิเคราะห์ความไวของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จต่อแต่ละปัจจัยทรัพยากรจะกำหนดค่าพื้นฐานซึ่งเป็นค่าความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จในเหตุการณ์ที่เป็นกลาง และกำหนดค่าระดับที่มากที่สุดและน้อยที่สุดของแต่ละปัจจัยจากตารางที่ 2.1 2.2 และ 2.3 ซึ่งสามารถสรุปได้ดังตารางที่ 3.3 หลังจากนั้นจะหาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเมื่อปัจจัย i มีค่าระดับที่น้อยที่สุด ($p_{i,\,fmin}$) และความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเมื่อปัจจัย i มีค่าระดับที่มากที่สุด ($p_{i,\,fmax}$) ดังนั้นค่าความน่าจะเป็นที่น้อยที่สุด ($p_{i,\,min}$) และมากที่สุด ($p_{i,\,max}$) ของปัจจัย i คือ $min(p_{i,\,fmin}, p_{i,\,fmax})$ และ$max(p_{i,\,fmin}, p_{i,\,fmax})$ ตามลำดับ โดยถ้า $p_{i,\,min}$ มีค่าน้อยแสดงว่าปัจจัย i ลดความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้มาก และถ้า $p_{i,\,max}$ มีค่ามากแสดงว่าปัจจัย i เพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้มาก หลังจากนั้นหาช่วงห่างของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเนื่องจากปัจจัย i (แทนด้วย r_i) จากสมการ $r_i = p_{i,\,max} - p_{i,\,min}$ โดยถ้า r_i มีค่ามากจะแสดงว่าปัจจัย i ส่งผล

มากต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ ในทางกลับกันถ้า r_i มีค่าน้อยจะแสดงว่าปัจจัย i ส่งผลน้อยต่อต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ

ตารางที่ 3.3 ค่ามากและน้อยที่สุดสำหรับแต่ละปัจจัย

ปัจจัย	ค่ามากที่สุด	ค่าน้อยที่สุด
PREC	ต่ำมาก	สูงพิเศษ
FLEX	ต่ำมาก	สูงพิเศษ
RESL	ต่ำมาก	สูงพิเศษ
TEAM	ต่ำมาก	สูงพิเศษ
PMAT	ต่ำมาก	สูงพิเศษ
RELY	ต่ำมาก	สูงมาก
DATA	ต่ำ	สูงมาก
CPLX	ต่ำมาก	สูงพิเศษ
RUSE	ต่ำ	สูงพิเศษ
DOCU	ต่ำมาก	สูงมาก
TIME	ปานกลาง	สูงพิเศษ
STOR	ปานกลาง	สูงพิเศษ
PVOL	ต่ำ	สูงมาก
ACAP	ต่ำมาก	สูงมาก
PCAP	ต่ำมาก	สูงมาก
PCON	ต่ำมาก	สูงมาก
APEX	ต่ำมาก	สูงมาก
PLEX	ต่ำมาก	สูงมาก
LTEX	ต่ำมาก	สูงมาก
TOOL	ต่ำมาก	สูงมาก
SITE	ต่ำมาก	สูงพิเศษ
SCED	ต่ำมาก	สูงมาก

บทที่ 4

การทดลองและวิเคราะห์ผล

การทดลองแบ่งออกเป็น 2 ส่วนตามแนวคิดของงานวิจัย คือ การทดลองการจัดสรรทรัพยากรสำหรับโครงการเดียวและการทดลองการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ โดยมีรายละเอียดการทดลองและผลที่ได้ดังต่อไปนี้

4.1 การทดลองการจัดสรรทรัพยากรสำหรับโครงการเดียว

กำหนดค่าคงที่ของตัวแปรตามเหตุการณ์ 3 แบบ คือ เหตุการณ์ที่ตัวแปรมีค่าต่ำ กลางและสูง เพื่อใช้ในทดลอง โดยมีวิธีกำหนดค่าคงที่ของแต่ละตัวแปรดังต่อไปนี้

1. ผลิตภาพของพนักงาน (α)

จากงานวิจัยของ Nogueira และคณะ **(19)** พบว่าค่าผลิตภาพของพนักงานสำหรับเหตุการณ์ที่ผลิตภาพของพนักงานต่ำคือ $0.8<\alpha<2.0$ และสำหรับเหตุการณ์ที่ผลิตภาพของพนักงานสูงคือ $2.0<\alpha<6.0$ ดังนั้นกำหนดให้จุดกึ่งกลางในแต่ละช่วงเป็นตัวแทนของช่วงทำให้ได้ค่าผลิตภาพของพนักงานสำหรับเหตุการณ์ต่ำคือ 1.4 สำหรับเหตุการณ์กลางคือ 2.0 และ สำหรับเหตุการณ์สูงคือ 4.0

2. การเปลี่ยนแปลงของความต้องการ (β)

Nogueira และคณะ **(19)** กำหนดช่วงของค่าอัตราการเกิดและอัตราการตายของความต้องการในกระบวนการพัฒนาเป็น 4 ช่วง คือ ช่วงเติบโต (Growing) ช่วงเปลี่ยนแปลง (Volatile) ช่วงเสถียร (Stable) และช่วงหดตัว (Shrinking) โดยช่วงของค่าอัตราการเกิดและตายของความต้องการในแต่ละช่วงของการพัฒนาคือ

ช่วงเติบโต: 10%<อัตราการเกิด<100% และ 0%<อัตราการตาย<10%

ช่วงเปลี่ยนแปลง: 10%<อัตราการเกิด<100% และ 10%<อัตราการตาย<100%

ช่วงเสถียร: 0%<อัตราการเกิด<10% และ 0%<อัตราการตาย<10%

ช่วงหดตัว: 0%<อัตราการเกิด<10% และ 10%<อัตราการตาย<100%

กำหนดให้ค่ากลางในแต่ละช่วงอัตราการเกิดและช่วงอัตราการตายเป็นตัวแทนของช่วงนั้น และคำนวณหาค่าการเปลี่ยนแปลงของความต้องการจากค่ากลางทั้งสอง โดยค่ากลางและการเปลี่ยนแปลงของความต้องการสรุปได้ดังตารางที่ 4.1

ตารางที่ 4.1 ค่ากลางของช่วงอัตราการเกิดและตายและการเปลี่ยนแปลงของความต้องการ

ช่วง	อัตราการเกิด	อัตราการตาย	การเปลี่ยนแปลงของความต้องการ
เติบโต	55%	5%	6
เปลี่ยนแปลง	55%	55%	11
เสถียร	5%	5%	1
หดตัว	5%	55%	6

จากตารางที่ 4.1 จะเห็นว่าช่วงที่มีค่าการเปลี่ยนแปลงของความต้องการต่ำสุดคือช่วงเสถียร มีค่าปานกลางคือช่วงเติบโตและหดตัว และมีค่ามากสุดคือช่วงเปลี่ยนแปลง ดังนั้นค่าการเปลี่ยนแปลงของความต้องการเรียงจากน้อยไปหามากคือ 1 6 และ 11

3. <u>ความซับซ้อนของซอฟต์แวร์ (γ)</u>

งานวิจัยของ Nogueira และคณะ **(19)** ศึกษาโครงการที่มีค่า LGC ระหว่าง 1,000 ถึง 30,000 ดังนั้นสมมุติให้ค่า LGC สำหรับเหตุการณ์ต่ำคือ 1,000 และสำหรับเหตุการณ์สูง คือ 30,000 และกำหนด LGC สำหรับเหตุการณ์กลางจากจุดกึ่งกลางของช่วงซึ่งเท่ากับ 15,500 หลังจากนั้นคำนวณหาค่าความซับซ้อนของซอฟต์แวร์จาก LGC ได้ค่าคือ 8 43 และ 52 เดือน สำหรับเหตุการณ์ต่ำ กลาง และสูงตามลำดับ

4. <u>ขนาดของซอฟต์แวร์</u>

ขนาดของซอฟต์แวร์ในหน่วย KLOC คำนวณจากค่า LGC ที่เป็นตัวเดียวกับที่ใช้ในการหาค่าความซับซ้อนของซอฟต์แวร์ โดยค่าขนาดของซอฟต์แวร์ที่ได้คือ 32 496 และ 960 สำหรับเหตุการณ์ต่ำ กลาง และสูง ตามลำดับ

5. <u>เวลาที่ใช้ในการพัฒนา (x)</u>

จากขั้นตอนที่ผ่านมาได้กำหนดค่าคงที่ของ α, β และ γ ไว้แล้ว และจากสมการ 2.6 ความสัมพันธ์ของ x กับ α, β, γ และ F คือ $x = \gamma + \beta\,(-\ln(1-F))^{1/\alpha}$ ดังนั้นจะหาค่า x ได้เมื่อรู้ค่าความน่าจะเป็นที่โครงการจะเสร็จสิ้น สมมุติว่าค่า x ควรเป็นค่าในกรณีที่โครงการมีโอกาสเสร็จสิ้นสูง

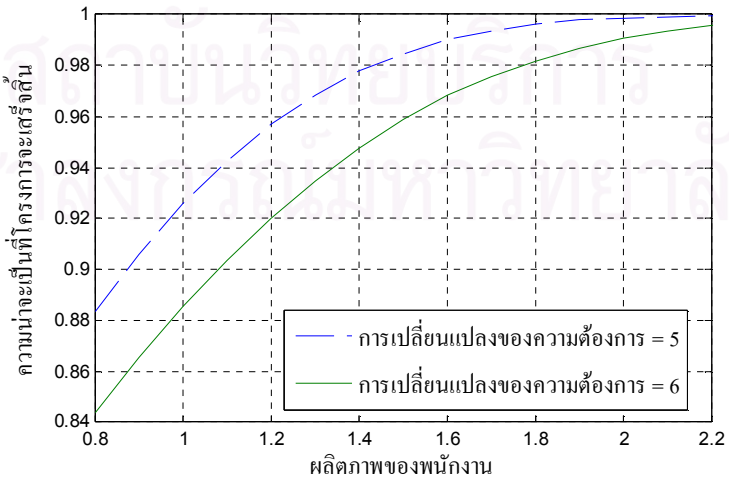
ดังนั้นกำหนดให้ความน่าจะเป็นที่โครงการจะเสร็จสิ้นเป็นค่าที่สูงคือ 99% แล้วคำนวณหาค่าเวลาที่ใช้ในการพัฒนา

จากขั้นตอนการคำนวณค่าคงที่ของตัวแปรทั้งหมดที่กล่าวมา สรุปเป็นค่าคงที่ที่ใช้ในการทดลองได้ดังตารางที่ 4.2

ตารางที่ 4.2 ค่าคงที่ของตัวแปรที่ใช้ในการทดลอง

ตัวแปร	ต่ำ	กลาง	สูง
α	1.4	2.0	4.0
β	1	6	11
γ(เดือน)	8	43	52
KLOC	32	496	960
x (เดือน)	11	56	68

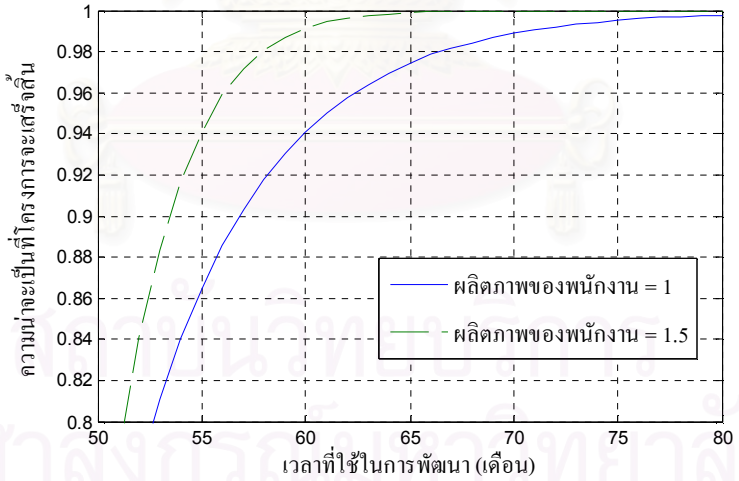
หลังจากกำหนดค่าคงที่ของตัวแปรแล้วทดลองประยุกต์ใช้แบบจำลองประเมินความเสี่ยงสำหรับการจัดสรรทรัพยากรของโครงการซอฟต์แวร์เชิงภาวะถ่วงดุล ตัวอย่างที่หนึ่ง สมมุติว่าโครงการจะต้องพัฒนาให้เสร็จสิ้นภายในเวลา 56 เดือน โครงการมีความซับซ้อนของซอฟต์แวร์ซึ่งหาได้จาก *KLOC* เท่ากับ 43 เดือน (γ=13·Ln((1000·*KLOC*-150)/32)-82) และต้องการความน่าจะเป็นที่โครงการจะเสร็จสิ้นภายใน 56 เดือนเท่ากับ 99%



รูปที่ 4.1 ผลิตภาพของพนักงานที่ต้องการ ณ การเปลี่ยนแปลงของความต้องการที่ระดับต่าง ๆ

ตัวอย่างการจัดสรรทรัพยากรเชิงภาวะถ่วงดุลแสดงได้ดังรูปที่ 4.1 ซึ่งจะเห็นว่าถ้าความน่าจะเป็นที่โครงการจะเสร็จสิ้นเท่ากับ 99% และการเปลี่ยนแปลงของความต้องการเท่ากับ 6 ผู้บริหารจะต้องควบคุมให้ผลิตภาพของพนักงานเท่ากับ 2 ซึ่งหมายถึง เวลาทำงานจริงเป็น 2 เท่าของเวลาอยู่เปล่า แต่สมมุติถ้าผลิตภาพของพนักงานไม่สามารถเท่ากับ 2 เนื่องจากข้อจำกัดด้านการปฏิบัติงานทำให้ผลิตภาพของพนักงานอยู่ที่ 1.6 ผู้บริหารอาจยอมลดการเปลี่ยนแปลงของความต้องการให้เหลือเพียง 5 เพื่อให้บรรลุตามจุดประสงค์ที่ต้องการ แต่ถ้าผู้บริหารไม่สามารถลดการเปลี่ยนแปลงของความต้องการนี้ลงได้ ผู้บริหารอาจยอมให้ความน่าจะเป็นที่โครงการจะเสร็จสิ้นลดลงเหลือเพียง 97%

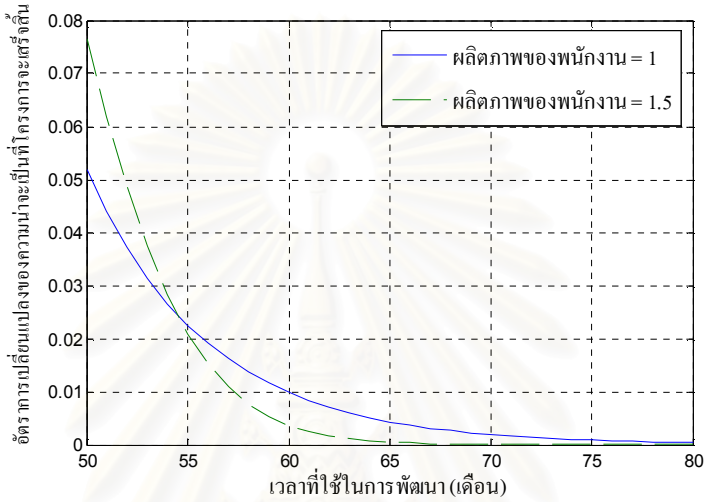
ตัวอย่างที่สองของการจัดสรรทรัพยากรเชิงภาวะถ่วงดุลแสดงดังรูปที่ 4.2 สมมุติว่าโครงการหนึ่งมีการเปลี่ยนแปลงของความต้องการเท่ากับ 6 ความซับซ้อนของซอฟต์แวร์เท่ากับ 43 เดือน และต้องการความน่าจะเป็นที่โครงการจะเสร็จสิ้นที่ 99% ถ้าผลิตภาพของพนักงานเท่ากับ 1 โครงการนี้จะใช้เวลาในการพัฒนาประมาณ 70 เดือน แต่สมมุติว่าโครงการนี้ต้องการเร่งให้เสร็จเร็วขึ้นคือเสร็จภายในเวลา 60 เดือน ผู้บริหารควรเพิ่มผลิตภาพของพนักงานเป็น 1.5 แต่ถ้าผู้บริหารไม่สามารถเพิ่มผลิตภาพของพนักงานได้ผู้บริหารอาจยอมลดความน่าจะเป็นที่โครงการนี้จะเสร็จสิ้นเหลือเพียง 94%



รูปที่ 4.2 เวลาที่ใช้ในการพัฒนา ณ ผลิตภาพของพนักงานที่ระดับต่าง ๆ

อัตราการเปลี่ยนแปลงของความน่าจะเป็นที่โครงการจะเสร็จสิ้นสามารถช่วยในการตัดสินใจได้เช่นกัน ดังรูปที่ 4.3 คือ ถ้าผลิตภาพของพนักงานเท่ากับ 1 เมื่อเพิ่มเวลาอีก 1 เดือนที่เวลา 65 เดือน ความน่าจะเป็นที่เพิ่มขึ้นน้อยมาก คือ น้อยกว่า 1% แต่ถ้าเพิ่มเวลาอีก 1 เดือนที่เวลา

50 เดือน ความน่าจะเป็นที่โครงการจะเสร็จสิ้นเพิ่มขึ้นถึง 5% และได้ถึง 7.5% ถ้าผลิตภาพของพนักงานเท่ากับ 1.5 ซึ่งวิเคราะห์ได้ว่าเมื่อเวลาที่ใช้ในการพัฒนานั้นมีจำกัด การเพิ่มเวลาหรือผลิตภาพของพนักงานสามารถเพิ่มความน่าจะเป็นที่โครงการจะเสร็จสิ้นได้ แต่ยิ่งเวลาในการพัฒนามีค่ามากขึ้นการเพิ่มความน่าจะเป็นที่โครงการจะเสร็จสิ้นยิ่งลดลงอันเนื่องจากโครงการที่ใช้เวลามากนั้นมีโอกาสที่จะเสร็จสิ้นสูงอยู่แล้ว

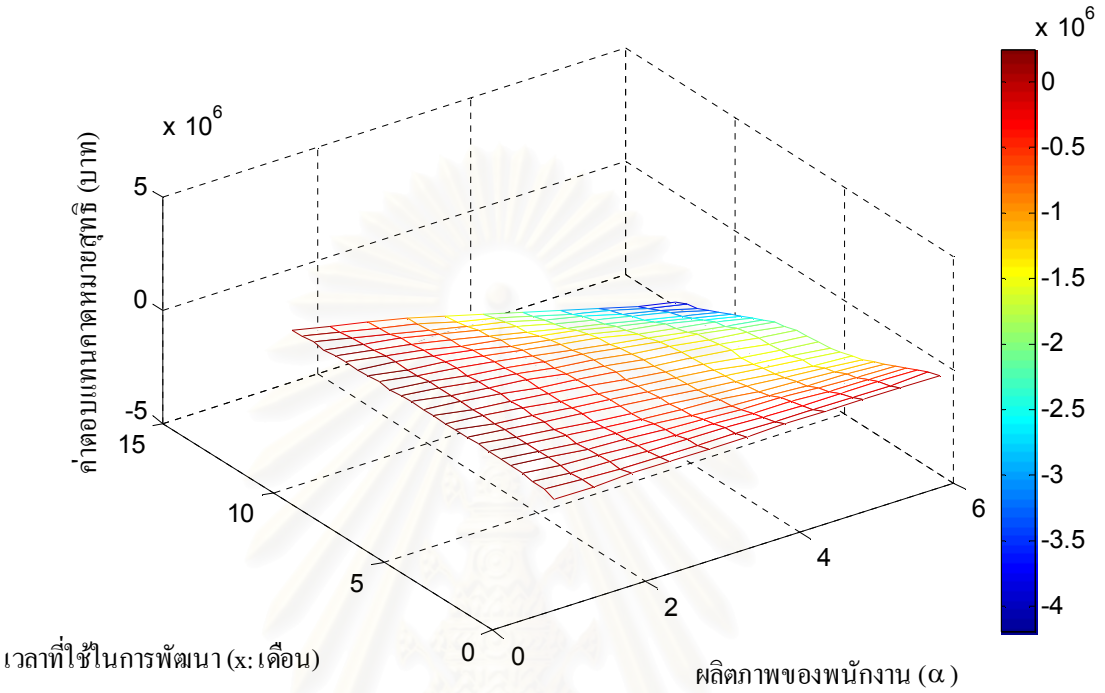


รูปที่ 4.3 อัตราการเปลี่ยนแปลงของความน่าจะเป็นที่โครงการจะเสร็จสิ้นและเวลาที่ใช้ในการพัฒนา

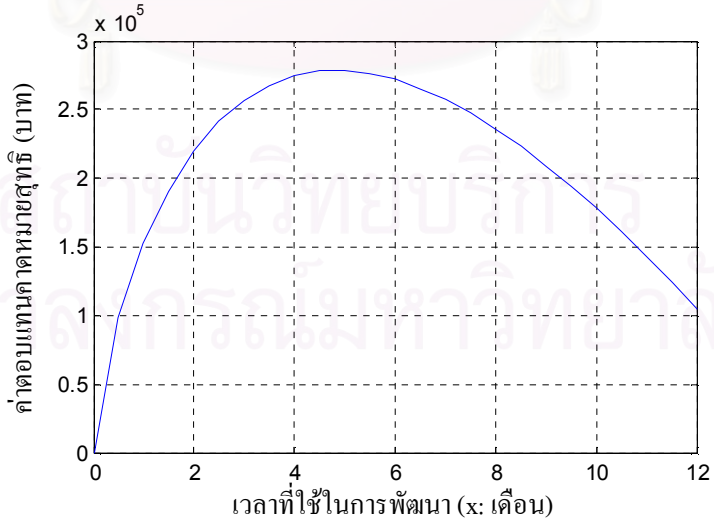
จากตัวอย่างที่กล่าวมาจะเห็นได้ว่าแบบจำลองประเมินความเสี่ยงสามารถสนับสนุนการตัดสินใจเชิงภาวะถ่วงดุลได้ และให้ผลลัพธ์ในรูปแบบของความน่าจะเป็นที่โครงการจะเสร็จสิ้น ความน่าจะเป็นนี้ยังแสดงถึงภาพรวมของการตัดสินใจจัดสรรทรัพยากรในโครงการได้เนื่องจากค่าความน่าจะเป็นนี้คำนวณมาจากทรัพยากรหลายตัวในโครงการ

การทดลองหาการจัดสรรทรัพยากรที่เหมาะที่สุดสำหรับโครงการนั้นจะใช้ตัวอย่างปัญหาในส่วนแนวคิดของงานวิจัยที่แสดงดังสมการที่ 3.4 โดยใช้เครื่องมือสำหรับแก้กำหนดการแบบไม่เชิงเส้นเพื่อหาคำตอบที่เหมาะที่สุด คำตอบที่ได้คือ $\alpha = 0.8$ และ $x = 4.84$ แสดงในรูปที่ 4.4 4.5 และ 4.6 โดยจุดที่ $\alpha = 0.8$ และ $x = 4.84$ จะเป็นจุดที่สูงที่สุดของกราฟภายในขอบเขตที่กำหนด ดังนั้นถ้าต้องการค่าตอบแทนคาดหมายสุทธิที่สูงที่สุด ผู้บริหารควรใช้ผลิตภาพของพนักงานที่ 0.8 และเวลาในการพัฒนาเท่ากับ 4.84 เดือน โดยได้ค่าตอบแทนคาดหมายสุทธิเท่ากับ 278,788 บาท สังเกตว่าผลิตภาพของพนักงานจากคำตอบเท่ากับ 0.8 ซึ่งเป็นค่าขอบของช่วง ทั้งนี้อาจเนื่องมาจากที่

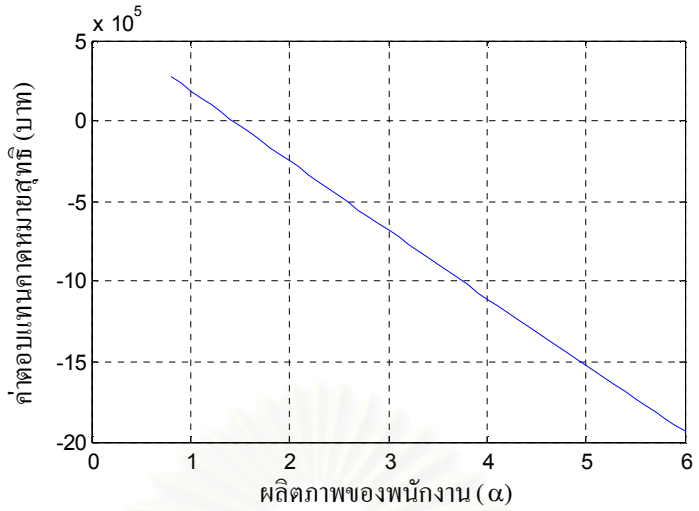
เวลาที่ใช้ในการพัฒนามีอัตราการเพิ่มความน่าจะเป็นที่โครงการจะเสร็จสิ้นมากกว่าผลิตภาพของพนักงานทำให้การเพิ่มเวลาที่ใช้ในการพัฒนาเป็นสิ่งที่เหมาะสมมากกว่าการเพิ่มผลิตภาพของพนักงาน



รูปที่ 4.4 ค่าตอบแทนคาดหมายสุทธิ ณ เวลาและผลิตภาพของพนักงาน



รูปที่ 4.5 ค่าตอบแทนคาดหมายสุทธิและเวลา



รูปที่ 4.6 ค่าตอบแทนคาดหมายสุทธิและผลิตภาพของพนักงาน

ความถูกต้องของคำตอบที่ได้ขึ้นอยู่กับความถูกต้องของแบบจำลองประเมินความเสี่ยงที่ใช้ ซึ่งแบบจำลองของ Nogueira และคณะที่นำมาประยุกต์ใช้นี้มีการตรวจสอบความสมเหตุสมผลกับแบบจำลองต้นทุน COCOMO 81 **(24)** และ Putnam **(25)** แล้ว อย่างไรก็ตามในปัจจุบัน COCOMO 81 เป็นแบบจำลองเก่าที่แทนที่ด้วย COCOMO II ซึ่งเป็นแบบจำลองที่ใหม่กว่า ดังนั้นเพื่อทดสอบว่าแบบจำลองยังคงเหมาะสมกับสภาพแวดล้อมในปัจจุบันจึงได้ตรวจสอบความสมเหตุสมผลของแบบจำลองประเมินความเสี่ยงใหม่โดยใช้ COCOMO II โดยมีสมมุติฐานในการตรวจสอบคือถ้าให้ค่าพารามิเตอร์ที่สมมูลกันแก่ COCOMO II และแบบจำลองประเมินความเสี่ยง ค่าความน่าจะเป็นที่โครงการจะเสร็จสิ้นโดยใช้เวลาที่คำนวณจาก COCOMO II ควรมีค่าสูงเนื่องจากค่าเวลาที่คำนวณจาก COCOMO II เป็นค่าที่เหมาะสมสำหรับการพัฒนาซอฟต์แวร์ซึ่งควรจะให้ความน่าจะเป็นที่โครงการจะเสร็จสิ้นสูง

การตรวจสอบความสมเหตุสมผลมีขั้นตอนดังนี้ ขั้นแรกกำหนดให้ค่าพารามิเตอร์เป็นค่าสำหรับเหตุการณ์กลางดังในตารางที่ 4.2 กล่าวคือ ผลิตภาพของพนักงาน (α) เท่ากับ 2 การเปลี่ยนแปลงของความต้องการ (β) เท่ากับ 6 และ KLOC เท่ากับ 496 (สมมูลกับ LGC ที่ 15,500 หรือความซับซ้อนของซอฟต์แวร์ที่ 43 เดือน) หลังจากนั้นคำนวณเวลาที่ใช้ในการพัฒนาจาก COCOMO II โดยแปลงค่าพารามิเตอร์ α และ β ให้เหมาะสมกับ COCOMO II โดยพารามิเตอร์สำหรับความสามารถของนักวิเคราะห์และนักเขียนโปรแกรมใน COCOMO II คือ ACAP และ PCAP ตามลำดับ และพารามิเตอร์สำหรับการเปลี่ยนแปลงและวิวัฒนาการของความต้องการใน COCOMO II คือ REVL จึงแปลงค่าของ α เป็น ACAP และ PCAP และแปลง β เป็น REVL โดยให้ทั้ง ACAP และ PCAP มีค่าปานกลางซึ่งหมายถึงความสามารถของนักวิเคราะห์และนักเขียน

โปรแกรมอยู่ระดับกลาง และให้ REVL มีค่าเป็น 60 ซึ่งหมายถึงมีคำสั่งโปรแกรมที่ถูกตัดทิ้งเป็นจำนวน 60% หลังจากกำหนดค่าพารามิเตอร์เหล่านี้แล้วใช้ COCOMO II คำนวณเวลาได้ค่าประมาณ 53 เดือน หลังจากนั้นนำเวลาคือ 53 เดือนไปคำนวณหาค่าความน่าจะเป็นที่โครงการจะเสร็จสิ้นโดยใช้แบบจำลองประเมินความเสี่ยง ซึ่งได้ค่าความความน่าจะเป็นเท่ากับ 94% ซึ่งเป็นค่าที่ค่อนข้างสูง หลังจากนั้นทำกลับกันคือกำหนดให้ค่าความน่าจะเป็นที่โครงการจะเสร็จสิ้นในแบบจำลองประเมินความเสี่ยงเท่ากับ 99% แล้วคำนวณหาเวลาซึ่งได้เท่ากับ 56 เดือน เห็นได้ว่าค่านี้ใกล้เคียงกับค่า 53 เดือนที่ได้จาก COCOMO II ดังนั้นผลลัพธ์จากแบบจำลองประเมินความเสี่ยงมีความใกล้เคียงกับผลลัพธ์จาก COCOMO II สำหรับเหตุการณ์กลาง

ตัวอย่างทั้งหมดนี้แสดงให้เห็นถึงการใช้แบบจำลองประเมินความเสี่ยงเพื่อจัดสรรทรัพยากรเชิงภาวะถ่วงดุลในโครงการซอฟต์แวร์ และหาการจัดสรรทรัพยากรในโครงการซอฟต์แวร์ที่เหมาะที่สุดภายใต้ข้อจำกัดด้านทรัพยากรที่มีอยู่ในโครงการได้

4.2 การทดลองการจัดสรรทรัพยากรสำหรับกลุ่มโครงการ

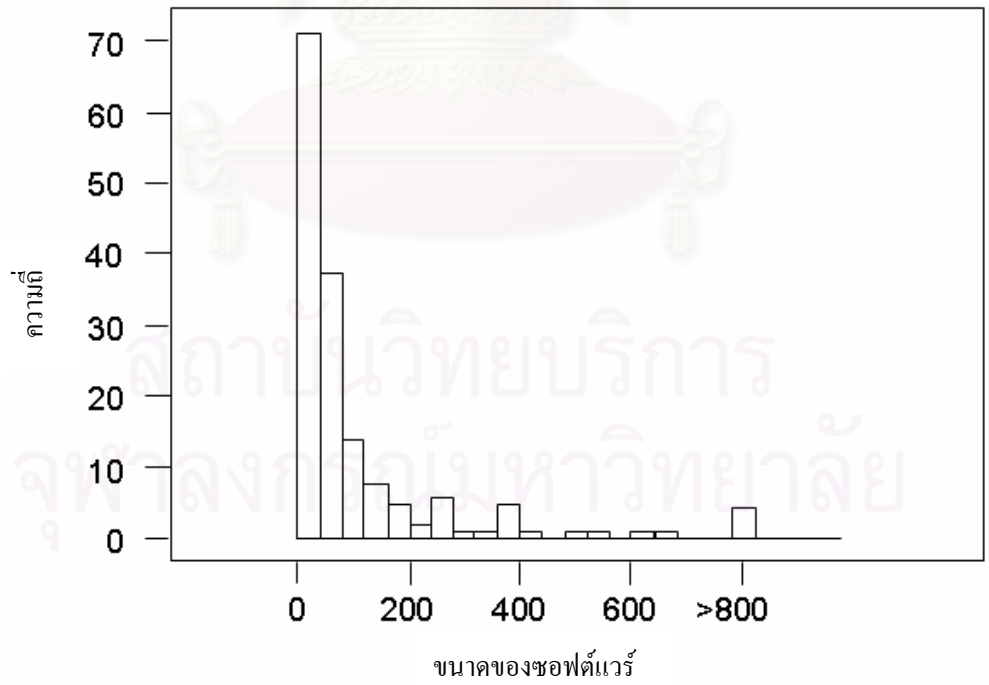
การทดลองการจัดสรรทรัพยากรสำหรับกลุ่มโครงการจะกระทำตามแนวคิดของงานวิจัยที่ได้เสนอ คือ ขั้นแรกจัดเตรียมข้อมูลของกลุ่มโครงการสำหรับการทดลองโดยใช้ข้อมูลจากโครงงานมหาบัณฑิตของกวี บูรพสายันห์ **(26)** ซึ่งได้ศึกษาการประเมินต้นทุนโดยใช้แบบจำลองต้นทุนต่าง ๆ เช่น COCOMO II สำหรับโครงการจำนวน 8 โครงการของบริษัท Progress Software จำกัด ข้อมูลประกอบด้วยค่าระดับของตัวประกอบมาตราส่วนและตัวขับต้นทุน ขนาดของซอฟต์แวร์ และวันที่เริ่มต้นและสิ้นกำหนดของแต่ละโครงการ

ตารางที่ 4.3 ค่าระดับโดยปริยายของตัวประกอบมาตราส่วนและตัวขับต้นทุน

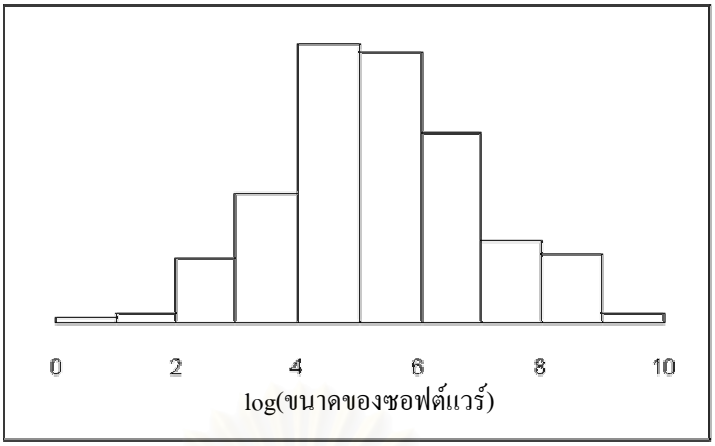
ตัวประกอบมาตราส่วน / ตัวขับต้นทุน	ความไม่แน่นอน
RESL, DATA, ACAP, PCAP	สูง
TEAM, PMAT, STOR, DOCU, TOOL, SCED	ปานกลาง
PREC, FLEX, TIME, PVOL, RELY, CPLX, SITE	ต่ำ
RUSE, APEX, PLEX, LTEX, PCON	ไม่มี

ขั้นที่สองกำหนดรูปแบบการแจกแจงความน่าจะเป็นของตัวประกอบมาตราส่วน ตัวขับต้นทุน และขนาดของซอฟต์แวร์ตามระดับความไม่แน่นอนโดยระดับความไม่แน่นอนของแต่ละปัจจัยใน COCOMO II กำหนดตามระดับความไม่แน่นอนโดยปริยาย (Default Level) ที่เสนอโดย McDonald **(27)** ดังตารางที่ 4.3

การแจกแจงความน่าจะเป็นของขนาดของซอฟต์แวร์กำหนดตามรูปแบบการแจกแจงความน่าจะเป็นของขนาดของซอฟต์แวร์ที่เสนอโดย Madachy **(28)** ซึ่งมีรูปแบบสามเหลี่ยม (Triangular) ปรกติ (Normal) ล็อกปรกติ (Lognormal) และล็อกปรกติแบบตัดปลาย (Truncated Lognormal) แต่การทดลองนี้จะไม่พิจารณาการแจกแจงแบบสามเหลี่ยมและปรกติเนื่องจากการแจกแจงของขนาดของซอฟต์แวร์ในงานวิจัยของ Chulani **(29)** ที่ได้วิเคราะห์เป็นแบบการแจกแจงแบบล็อกปรกติ ดังจะเห็นได้จากข้อมูลที่มีลักษณะเบ้ขวา (Postitive Skew) ดังรูปที่ 4.7 และค่าลอการิทึมของข้อมูลมีลักษณะคล้ายการแจกแจงแบบปรกติดังรูปที่ 4.8 ซึ่งลักษณะทั้งสองแสดงถึงการแจกแจงแบบล็อกปรกติและล็อกปรกติแบบตัดปลาย อย่างไรก็ตามล็อกปกติแบบตัดปลายเหมาะกับข้อมูลที่ค่าเฉลี่ยมีค่าใกล้ศูนย์แต่ค่าเฉลี่ยของข้อมูลที่ใช้ในการทดลองนี้มีค่าเฉลี่ยไม่เข้าใกล้ศูนย์ ด้วยเหตุนี้จึงไม่พิจารณาล็อกปกติแบบตัดปลาย ดังนั้นจากเหตุผลที่กล่าวมาทั้งหมดการแจกความน่าจะเป็นที่เหมาะสมของขนาดของซอฟต์แวร์สำหรับการทดลองนี้คือล็อกปรกติ

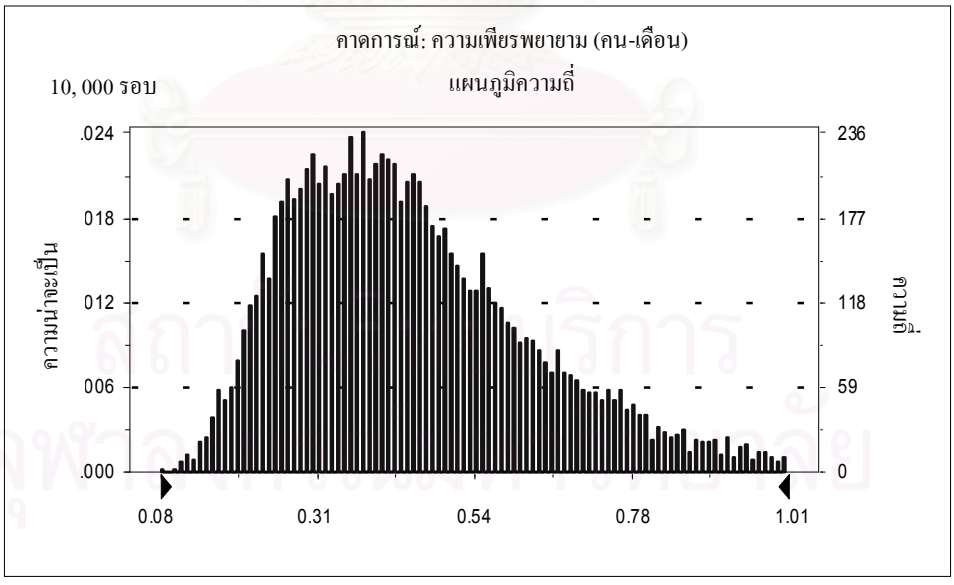


รูปที่ 4.7 ขนาดของซอฟต์แวร์

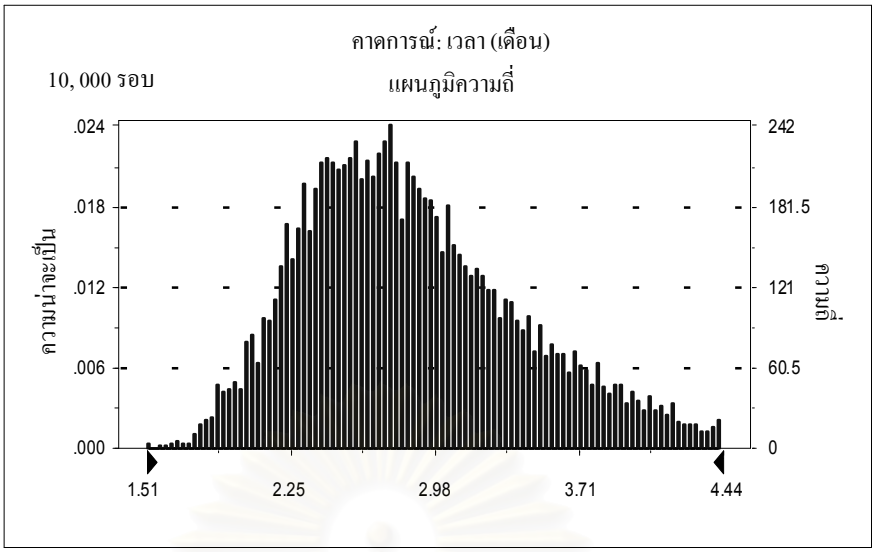


รูปที่ 4.8 ค่าลอการิทึมของขนาดของซอฟต์แวร์

ขั้นที่สามกำหนดค่าพารามิเตอร์ของการแจกแจงความน่าจะเป็นสำหรับตัวประกอบมาตราส่วนและตัวขับต้นทุน โดยสมมุติให้ความคลาดเคลื่อนเท่ากับ 1 และในขั้นตอนสุดท้ายใช้เครื่องมือจำลองสถานการณ์มอนติคาร์โลจำนวน 10,000 รอบทำให้ได้ข้อมูลความเพียรพยายามและเวลาของโครงการที่ 1 ถึง 8 จำนวน 10,000 ชุด รูปที่ 4.9 และ 4.10 แสดงตัวอย่างของการแจกแจงความน่าจะเป็นของความเพียรพยายามและเวลาตามลำดับสำหรับโครงการที่ 1



รูปที่ 4.9 ตัวอย่างการแจกแจงความน่าจะเป็นของความเพียรพยายาม



รูปที่ 4.10 ตัวอย่างการแจกแจงความน่าจะเป็นของเวลา

ขั้นสุดท้ายหาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จโดยใช้เงื่อนไขความเพียรพยายามและเวลา โดยสมมุติให้จำนวนพนักงานเท่ากับ 3 ตามข้อมูล และจากข้อมูลความเพียรพยายามและเวลาจำนวน 10, 000 ชุดที่ได้จากการจำลองสถานการณ์มอนติคาร์โล จะนับจำนวนชุดข้อมูลที่ผ่านเงื่อนไขตามสมการที่ 3.5 และ 3.6 สำหรับเงื่อนไขความเพียรพยายามและเวลาตามลำดับ โดยความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเท่ากับจำนวนชุดข้อมูลที่ผ่านเงื่อนไขหารด้วยจำนวนรอบของการจำลองสถานการณ์ (ในที่นี้คือ 10,000 รอบ) ตัวอย่างเช่น ถ้าจำนวนชุดข้อมูลที่ผ่านเงื่อนไขความเพียรพยายามเท่ากับ 9,993 ชุด ดังนั้นความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเท่ากับ 9,993/10,000=0.9993 ตารางที่ 4.4 แสดงผลลัพธ์ของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จสำหรับเงื่อนไขความเพียรพยายามและเงื่อนไขเวลา

ตารางที่ 4.4 ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ

	เงื่อนไขความเพียรพยายาม	**เงื่อนไขเวลา**
ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ	99.93%	90.54%

จากตารางจะเห็นว่าความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จโดยใช้เงื่อนไขเวลามีค่าน้อยกว่าความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จโดยใช้เงื่อนไขความเพียรพยายาม ซึ่งอาจเกิดเนื่องจากความสัมพันธ์ระหว่างเวลาและความเพียรพยายามสำหรับเงื่อนไขความเพียรพยายาม (สมการที่ 3.5) เป็นแบบเชิงเส้น (*Effort=E·Time*) แต่สำหรับเงื่อนไขเวลา (สมการที่ 3.6) จะเป็นแบบไม่เชิงเส้นซึ่ง

เวลาจะหาได้จากสมการหาเวลาของ COCOMO II (สมการที่ 2.2) ดังนั้นสมมุติว่ามีโครงการหนึ่งต้องการความเพียรพยายาม 120 คน-เดือน ถ้าพิจารณาแบบเงื่อนไขความเพียรพยายามจะหมายความว่าถ้าโครงการนี้มีพนักงาน 10 คน พนักงาน10 คนนี้จะต้องทำงานตลอด 12 เดือน (10x12=120 คน-เดือน) แต่ถ้าพิจารณาแบบเงื่อนไขเวลาโครงการนี้จะต้องการเวลามากกว่า 12 เดือน **(15)** จากเหตุผลนี้ทำให้เงื่อนไขแบบเวลาต้องการเวลามากกว่าจึงทำให้ความน่าจะเป็นที่จะสำเร็จมีค่าน้อยกว่า อย่างไรก็ตามเงื่อนไขแบบเวลานี้มีลักษณะคล้ายความเป็นจริงมากกว่า

สำหรับการทดลองปรับเปลี่ยนปัจจัยใน COCOMO II เพื่อปรับเปลี่ยนความน่าจะเป็นที่โครงการจะสำเร็จจะทดลองโดยใช้ปัจจัย TOOL เป็นตัวอย่าง ซึ่งคือการศึกษาว่าการใช้เครื่องมือในระดับที่ต่างกันมีผลต่อความสำเร็จของกลุ่มโครงการอย่างไร โดยทำการทดลอง โดยสร้างกลุ่มโครงการสองกลุ่ม กลุ่มที่หนึ่งกำหนดให้ TOOL ของทุกโครงการมีค่าระดับสูงมาก ส่วนกลุ่มที่สองจะกลับกันคือมีค่าระดับต่ำมาก ผลการทดลองแสดงดังตารางที่ 4.5

ตารางที่ 4.5 ผลการทดลองของการปรับเปลี่ยน TOOL

TOOL	**ความน่าจะเป็นที่จะสำเร็จ**	
	เงื่อนไขความเพียรพยายาม	**เงื่อนไขเวลา**
ต่ำมาก	98.80%	82.71%
สูงมาก	99.97%	94.84%

จากตารางเห็นได้ว่าค่าความน่าจะเป็นที่โครงการจะสำเร็จมีค่าแตกต่างกันโดยค่าความน่าจะเป็นที่ได้จากโครงการที่ TOOL มีค่าระดับสูงมากจะมีค่ามากกว่าโครงการที่ TOOL มีค่าระดับต่ำมาก ดังนั้นการใช้เครื่องมือที่ดีขึ้นจะทำให้ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จสูงขึ้นทั้งเงื่อนไขแบบความเพียรพยายามและเวลา และชี้ให้เห็นได้ว่าแบบจำลองที่เสนอนี้สามารถรองรับการปรับเปลี่ยนปัจจัยอื่นนอกจากเส้นตายและจำนวนพนักงานซึ่งมีผลต่อความสำเร็จของกลุ่มโครงการได้

สำหรับการทดลองประโยชน์ที่ได้จากการบริหารแบบกลุ่มโครงการ จะทดลองโดยสร้างกลุ่มโครงการสองกลุ่ม กลุ่มที่หนึ่งกำหนดให้การแจกแจงความน่าจะเป็นของทุกปัจจัยเป็นแบบปรกติซึ่งแทนความไม่แน่นอนต่ำ และกลุ่มที่สองให้การแจกแจงความน่าจะเป็นของทุกปัจจัยเป็นแบบเอกรูปซึ่งแทนความไม่แน่นอนสูง ผลลัพธ์ของการทดลองแสดงดังตารางที่ 4.6 สำหรับเงื่อนไขความเพียรพยายามและตารางที่ 4.7 สำหรับเงื่อนไขเวลา

ตารางที่ 4.6 ประโยชน์จากการบริหารแบบกลุ่มโครงการโดยใช้เงื่อนไขความเพียรพยายาม

ความไม่แน่นอน	ความน่าจะเป็นที่จะสำเร็จ		ประโยชน์
	โครงการเป็นอิสระต่อกัน	กลุ่มโครงการ	
ต่ำ	95.65%	99.72%	1.04
สูง	91.08%	99.21%	1.09

ตารางที่ 4.7 ประโยชน์จากการบริหารแบบกลุ่มโครงการโดยใช้เงื่อนไขเวลา

ความไม่แน่นอน	ความน่าจะเป็นที่จะสำเร็จ		ประโยชน์
	โครงการเป็นอิสระต่อกัน	กลุ่มโครงการ	
ต่ำ	10.99%	90.34%	8.22
สูง	5.42%	81.28%	15

จากผลลัพธ์จะเห็นว่ากลุ่มโครงการที่ปัจจัยด้านทรัพยากรมีความไม่แน่นอนสูงจะได้รับประโยชน์ซึ่งในที่นี้คือความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ จากการบริหารแบบกลุ่มโครงการมากกว่า ดังนั้นการบริหารแบบกลุ่มโครงการจึงมีประโยชน์มากสำหรับกลุ่มโครงการที่ปัจจัยด้านทรัพยากรมีความไม่แน่นอนสูง

หลังจากนั้นจะวิเคราะห์ความไวของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จต่อแต่ละปัจจัยด้านทรัพยากรเพื่อดูว่าการปรับเปลี่ยนแต่ละปัจจัยด้านทรัพยากรส่งผลต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากน้อยเพียงใด โดยขั้นแรกจะสร้างกลุ่มโครงการของเหตุการณ์ที่เป็นกลาง เพื่อใช้เปรียบเทียบว่าการปรับเปลี่ยนนั้นมีผลแตกต่างอย่างไรจากกลุ่มโครงการของเหตุการณ์ที่เป็นกลางนี้ โดยในการทดลองนี้จะสร้างกลุ่มโครงการในสถานการณ์กลางที่ประกอบด้วยโครงการ 8 โครงการ โดยกำหนดขนาดของซอฟต์แวร์เท่ากับ 200 KLOC ซึ่งเป็นค่าของซอฟต์แวร์ที่มีขนาดกลางของ Boehm **(24)** ระดับความไม่แน่นอนของปัจจัยเป็นปานกลาง และกำหนดให้กลุ่มโครงการนี้มีความน่าจะเป็นที่จะสำเร็จเท่ากับ 50% ซึ่งหมายถึงมีโอกาสที่กลุ่มโครงการนี้จะสำเร็จและไม่สำเร็จมีค่าเท่ากัน ซึ่งการกำหนดให้ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเท่ากับ 50% นั้นจะใช้วิธีการของ Fewster หาเส้นตายเพื่อให้ได้ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จตามที่ต้องการ ซึ่งวิธีการนี้จะสมมุติว่าความน่าจะเป็นที่แต่ละโครงการจะทำให้กลุ่มโครงการไม่สำเร็จมีค่าเท่ากัน

หรือความน่าจะเป็นที่แต่ละโครงการจะไม่สำเร็จเป็นอันแรก (โครงการที่มีเส้นตายก่อนหน้าโครงการนี้สำเร็จทุกอัน) เท่ากับ 50%/8 = 6.25%

ขั้นตอนในการหาความเพียรพยายามสำหรับเงื่อนไขความเพียรพยายามหรือเส้นตายสำหรับเงื่อนไขเวลามีรายละเอียดดังนี้ ขั้นแรกจะจำลองสถานการณ์มอนติคาร์โลของกลุ่มโครงการที่กล่าวแล้วในข้างต้นเป็นจำนวน 10,000 รอบ ซึ่งจะได้ข้อมูลของความเพียรพยายามหรือเวลาที่ควรใช้ในการพัฒนาจำนวน 10,000 ตัว หลังจากนั้นจะหาความเพียรพยายามหรือเส้นตายที่ทำให้โครงการที่หนึ่งมีความน่าจะเป็นที่จะไม่สำเร็จเท่ากับ 6.25% จากเงื่อนไขที่หนึ่ง (บรรทัดที่หนึ่ง) ในสมการที่ 3.5 หรือ 3.6 จะเห็นว่าถ้าต้องการให้ความน่าจะเป็นนี้เท่ากับ 6.25% จะต้องมีจำนวนความเพียรพยายามหรือเวลาของโครงการที่หนึ่ง (X_1 หรือ T_1) ที่มากกว่า $E_1=E^{\cdot}m_1$ หรือ m_1 ตามลำดับเป็นจำนวน 6.25/100×10,000 = 625 ตัว ฉะนั้นจึงหา E_1 หรือ m_1 ได้โดยการนำข้อมูลความเพียรพยายามหรือเส้นตายมาเรียงลำดับจากมากไปน้อยซึ่งความเพียรพยายามหรือเวลาลำดับที่ 626 ของโครงการที่หนึ่งจะเป็นค่าความเพียรพยายามหรือเส้นตายที่ทำให้ข้อมูลลำดับที่ 1 ถึง 625 เป็นข้อมูลที่ไม่ผ่านตามเงื่อนไขคือใช้ความเพียรพยายามหรือเวลามากกว่า E_1 หรือ m_1 หลังจากนั้นจะหาความเพียรพยายามหรือเส้นตายสำหรับโครงการที่สอง (E_2 และ m_2) โดยใช้วิธีการหาเช่นเดียวกับโครงการที่หนึ่ง คือ ลบข้อมูลลำดับที่ 1 ถึง 625 ของข้อมูลเดิมก่อน แล้วหาข้อมูลลำดับที่ 626 ทำเช่นนี้ไปเรื่อย ๆ จนกระทั่งโครงการที่แปดก็จะได้ความเพียรพยายามหรือเส้นตายของทั้ง 8 โครงการที่จะทำให้กลุ่มโครงการมีความน่าจะเป็นที่จะสำเร็จเท่ากับ 50% ซึ่งความเพียรพยายามหรือเส้นตายที่ได้นี้แสดงดังตารางที่ 4.8

ตารางที่ 4.8 ค่าความเพียรพยายามและเส้นตายที่ทำให้ความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเท่ากับ 50%

	โครงการที่							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
ความเพียรพยายาม (คน-เดือน)	870.58	1434.29	1956.08	2465.10	2960.71	3438.29	3896.02	4361.24
เส้นตาย (เดือน)	36.98	65.23	92.09	118.54	144.72	170.42	195.88	221.08

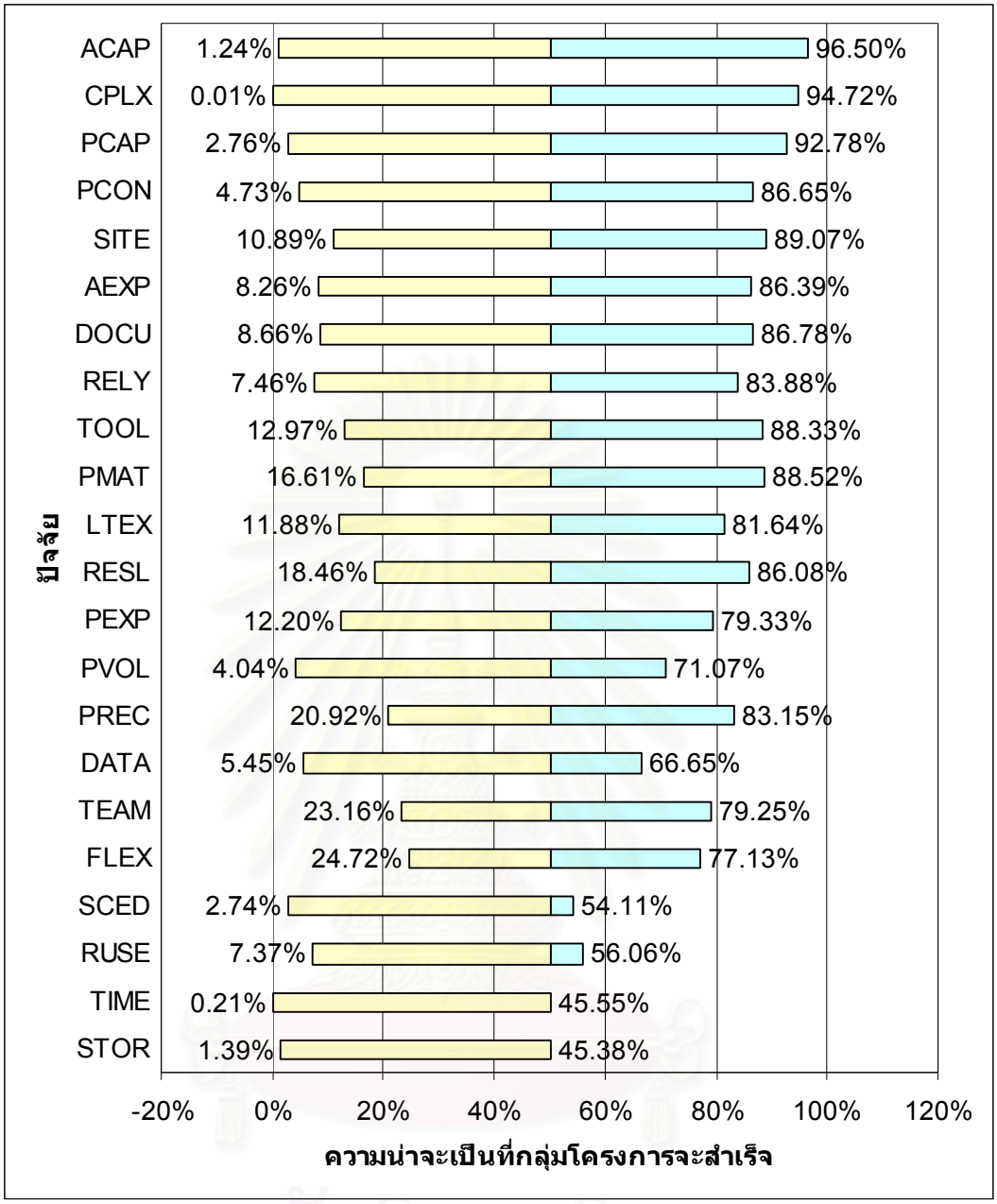
อย่างไรก็ตามเนื่องจากการใช้ข้อมูลลำดับที่ 626 เป็นเส้นตายนี้อาจเกิดปัญหาได้ในกรณีที่ค่าของข้อมูลลำดับที่ 625 และ 626 แตกต่างกันมาก ตัวอย่างเช่น ถ้าข้อมูลเวลาลำดับที่ 625 และ 626

คือ 100 เดือนและ 30 เดือนตามลำดับ ดังนั้นเส้นตายซึ่งเท่ากับ 30 เดือนจะดูไม่เหมาะสม เนื่องจากเส้นตาย 30 เดือนใช้เป็นตัวแบ่ง 100 เดือนและ 30 เดือนแต่เส้นตายกลับมีค่าต่างจาก 100 เดือนมาก ดังนั้นการกำหนดเส้นตายควรพิจารณาด้วยว่าค่าข้อมูลทั้งสองนี้แตกต่างกันมากหรือไม่ ความเพียรพยายามและเวลาลำดับที่ 625 และ 626 ของกลุ่มโครงการในการทดลองนี้แสดงดังตารางที่ 4.9 ซึ่งจะเห็นว่าความเพียรพยายามและเวลาลำดับที่ 625 และ 626 ของทุกโครงการแตกต่างกันไม่เกิน 1 ซึ่งเป็นค่าที่ยอมรับได้สำหรับการทดลองนี้ ดังนั้นเส้นตายที่ได้จากตารางที่ 4.8 จึงเหมาะสมสำหรับการทดลอง

ตารางที่ 4.9 ความเพียรพยายามและเวลาลำดับที่ 625 และ 626 ของกลุ่มโครงการ

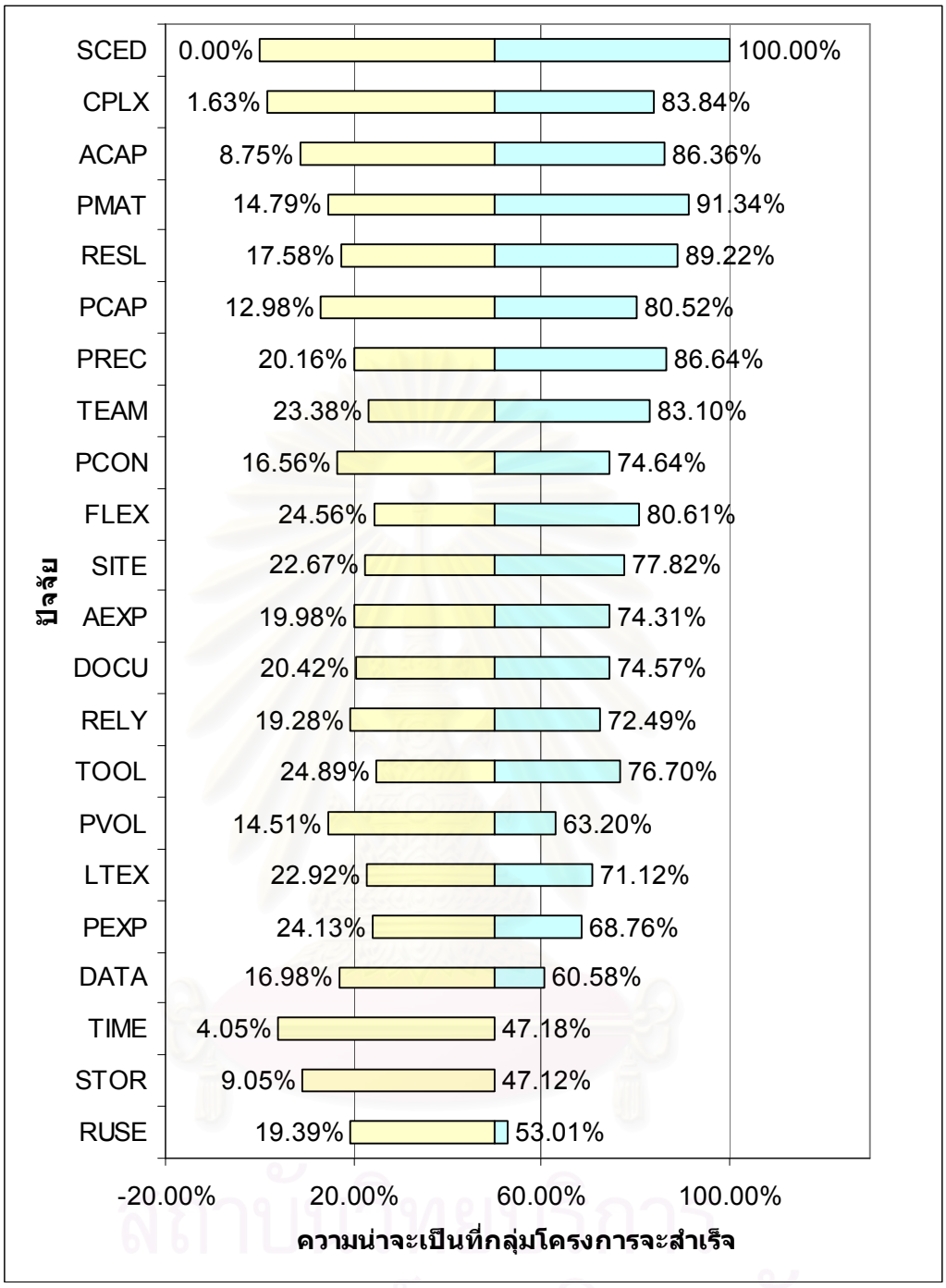
โครงการที่	ข้อมูลลำดับที่			
	ความเพียรพยายาม (คน-เดือน)		เวลา (เดือน)	
	625	626	625	626
1	870.81	870.58	36.99	36.98
2	1,434.44	1,434.29	65.25	65.23
3	1,956.09	1,956.08	92.10	92.09
4	2,465.45	2,465.10	118.56	118.54
5	2,960.76	2,960.71	144.73	144.72
6	3,439.33	3,438.29	171.62	171.62
7	3,896.33	3,896.02	195.89	195.88
8	4,361.71	4,361.24	221.08	221.08

หลังจากนั้นจะวิเคราะห์ความไวของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จต่อแต่ละปัจจัยตามแนวคิดการวิจัย โดยกำหนดค่ามากที่สุดและน้อยที่สุดของแต่ละปัจจัยแก่ทุกโครงการตามตารางที่ 3.3 เช่น ทุกโครงการในกลุ่มโครงการมีปัจจัย PREC เท่ากับต่ำมากหรือสูงพิเศษ ทุกโครงการในกลุ่มโครงการมีปัจจัย FLEX เท่ากับต่ำมากหรือสูงพิเศษ เป็นต้น และจำลองสถานการณ์มอนติคาร์โลเพื่อหาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จและแสดงผลลัพธ์ในแผนภูมิทอร์นาโด (Tornado Chart) ซึ่งเรียงปัจจัยตามช่วงห่างของค่าความน่าจะเป็นดังรูปที่ 4.11 และ 4.12 สำหรับเงื่อนไขความเพียรพยายามและเวลาตามลำดับ และความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จที่น้อยที่สุดและมากที่สุดสำหรับเงื่อนไขความเพียรพยายามและเวลาแสดงดังรูปที่ 4.13 และ 4.14 ตามลำดับ

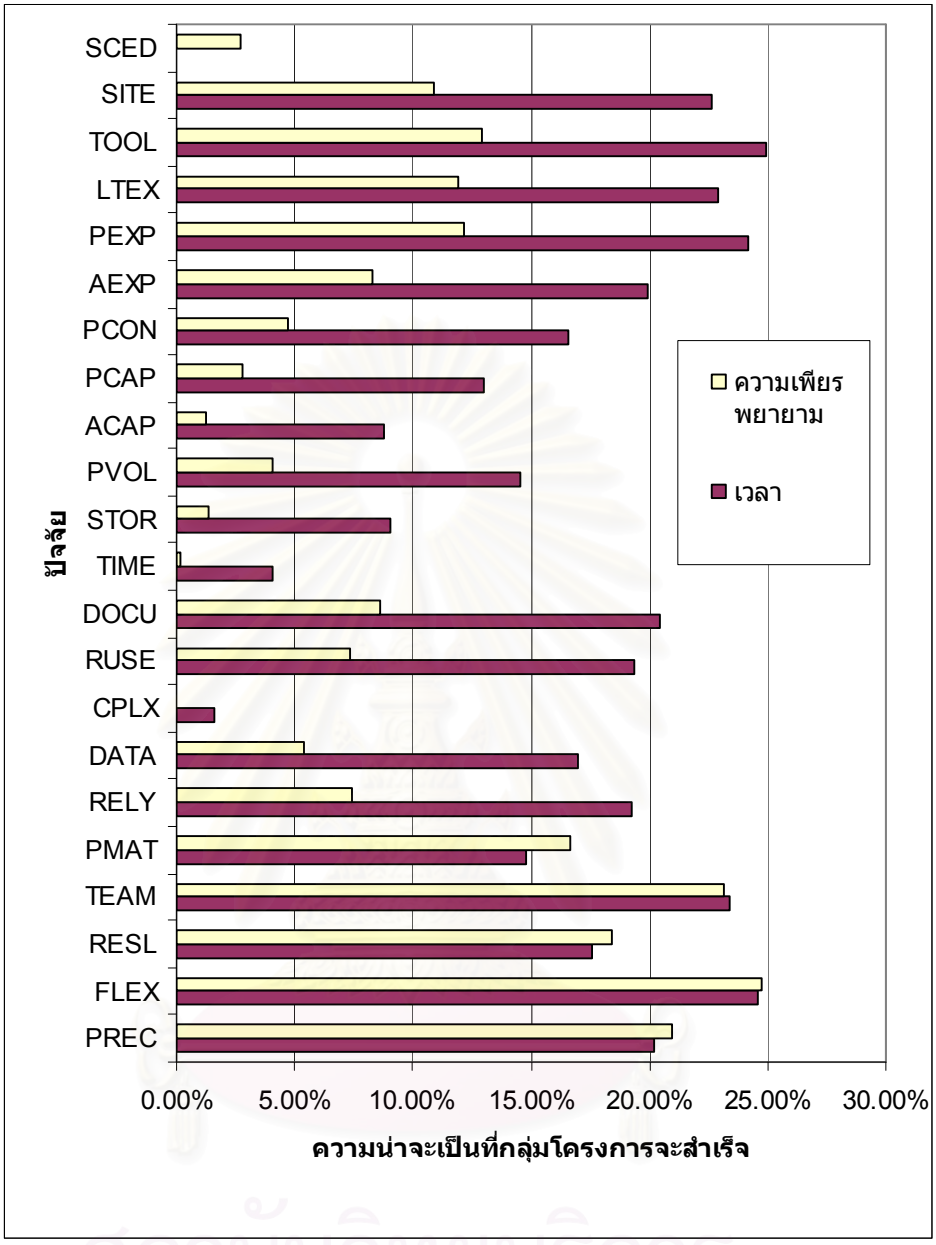


รูปที่ 4.11 แผนภูมิทอร์นาโดของปัจจัยด้านทรัพยากรสำหรับเงื่อนไขความเพียรพยายาม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

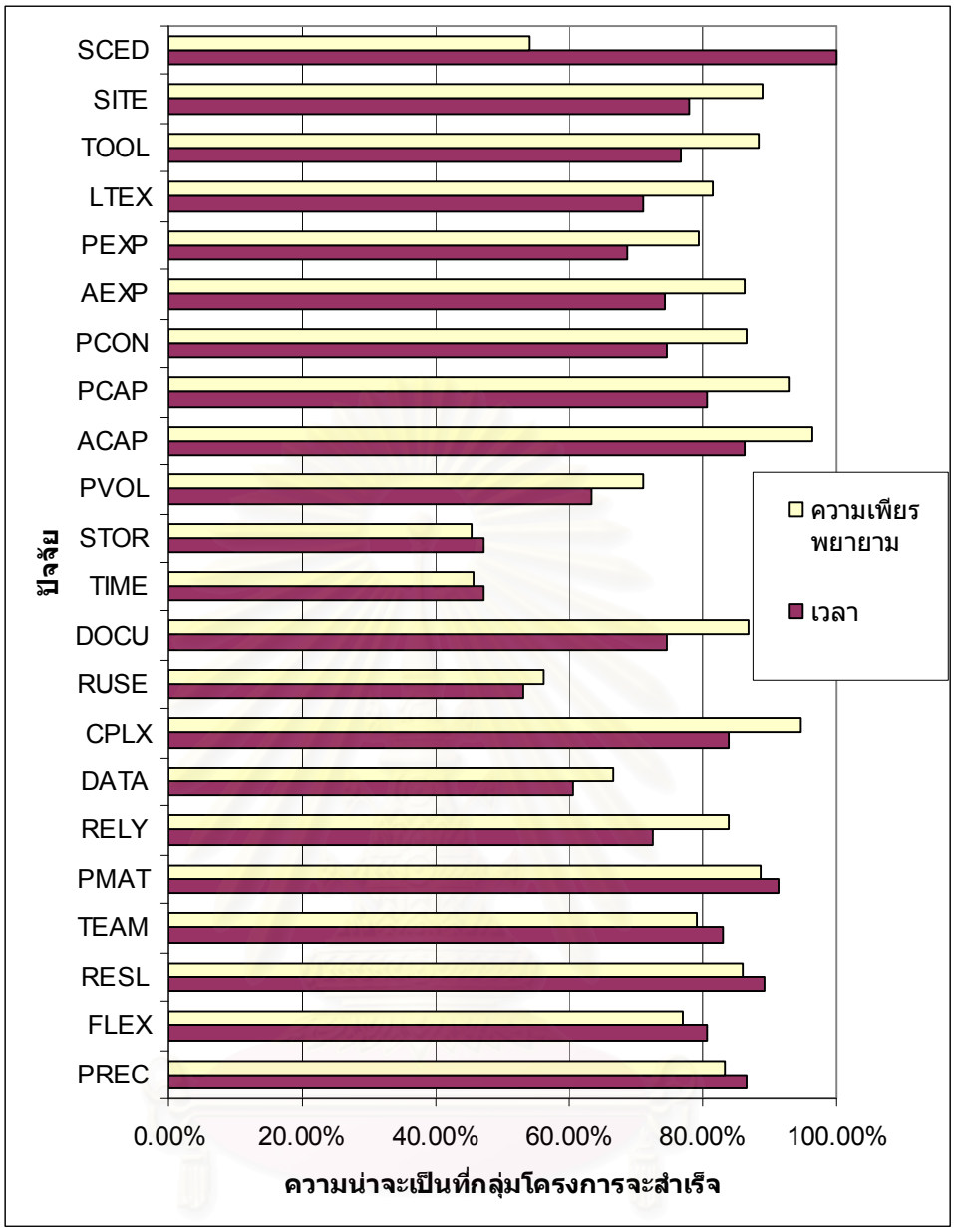


รูปที่ 4.12 แผนภูมิทอร์นาโดของปัจจัยด้านทรัพยากรสำหรับเงื่อนไขเวลา



รูปที่ 4.13 แผนภูมิแท่งของความน่าจะเป็นที่น้อยที่สุดของปัจจัยด้านทรัพยากร

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.14 แผนภูมิแท่งของความน่าจะเป็นที่มากที่สุดของปัจจัยด้านทรัพยากร

จากรูปที่ 4.11 จะเห็นว่าปัจจัยที่มีช่วงห่างของค่าความน่าจะเป็นที่มากที่สุดและน้อยที่สุด คือ ACAP และ STOR ตามลำดับ ซึ่งหมายความว่า เมื่อพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จโดยพิจารณาจากความเพียรพยายามนั้น ความสามารถของนักวิเคราะห์มีผลกระทบมากที่สุด และการใช้หน่วยเก็บหลักของระบบมีผลกระทบน้อยที่สุด นอกจากนี้ถ้าเปรียบเทียบเฉพาะปัจจัยสองตัว เช่น TOOL ซึ่งหมายถึง การใช้เครื่องมือ กับ PCAP ซึ่งหมายถึงความสามารถของนักเขียนโปรแกรม การเพิ่มหรือลดความสามารถของนักเขียนโปรแกรมมีผลกระทบต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากกว่าการใช้เครื่องมือ

จากรูปที่ 4.12 เห็นได้ว่าช่วงของค่าความน่าจะเป็นที่มีค่ามากที่สุดและน้อยที่สุด คือ SCED และ RUSE ตามลำดับ กล่าวคือ ถ้าพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจากเวลานั้น กำหนดการของการพัฒนานั้นมีผลมากที่สุดและการพัฒนาสำหรับความสามารถใช้ซ้ำนั้นมีผลน้อยที่สุด

จากรูปที่ 4.13 เห็นได้ว่าปัจจัยที่ทำให้ค่าความน่าจะเป็นน้อยที่สุดสำหรับเงื่อนไขความเพียรพยายาม คือ CPLX ดังนั้นถ้าพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจากความเพียรพยายามนั้น การเพิ่มความซับซ้อนของผลิตภัณฑ์จะลดความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากที่สุด และปัจจัยที่มีค่าความน่าจะเป็นที่น้อยที่สุดสำหรับเงื่อนไขเวลา คือ SCED ดังนั้นถ้าพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจากเวลานั้น การลดกำหนดการของการพัฒนาจะลดความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากที่สุด

จากรูปที่ 4.14 เห็นได้ว่าปัจจัยที่มีค่าความน่าจะเป็นที่มากที่สุดสำหรับเงื่อนไขความเพียรพยายาม คือ ACAP ดังนั้นถ้าพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจากความเพียรพยายามนั้น การเพิ่มความสามารถของนักวิเคราะห์จะเพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้มากที่สุด และปัจจัยที่มีค่าความน่าจะเป็นที่มากที่สุดสำหรับเงื่อนไขเวลา คือ SCED ดังนั้นถ้าพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจากเวลานั้น การเพิ่มกำหนดการของการพัฒนาจะเพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้มากที่สุด

ดังนั้นถ้าสมมุติว่ากลุ่มโครงการในการทดลองคือกลุ่มโครงการจริงของผู้บริหาร ซึ่งมีความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จเท่ากับ 50% ถ้าผู้บริหารพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจากความเพียรพยายาม และผู้บริหารต้องการเพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ ผู้บริหารควรเพิ่มความสามารถของนักวิเคราะห์เนื่องจากเป็นปัจจัยที่เพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้มากที่สุด และต้องระวังในการเพิ่มความซับซ้อนของผลิตภัณฑ์เนื่องจากเป็นปัจจัยที่ลดความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้มากที่สุด แต่ถ้าผู้บริหารพิจารณาความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จจากเวลา เมื่อผู้บริหารต้องการเพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จผู้บริหารควรเพิ่มกำหนดการของการพัฒนาเนื่องจากเป็นปัจจัยที่เพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากที่สุด และต้องระวังการลดกำหนดการของการพัฒนาด้วยเนื่องจากเป็นปัจจัยที่ลดความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากที่สุด

ดังนั้นวิธีการวิเคราะห์ความไวของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จต่อปัจจัยด้านทรัพยากรนั้นสามารถประยุกต์ใช้กับกลุ่มโครงการในสถานการณ์จริงของบริษัทได้ โดยวิเคราะห์ความไวตามขั้นตอนที่กล่าวในการทดลองนี้ แต่กำหนดข้อมูลตามกลุ่มโครงการจริง หลังจากนั้นผู้บริหารสามารถพิจารณาได้ว่าปัจจัยด้านทรัพยากรในกลุ่มโครงการที่ต่างกันส่งผลต่อความน่าจะ

เป็นที่กลุ่มโครงการจะสำเร็จแตกต่างกันอย่างไร และผู้บริหารควรปรับปรุงปัจจัยด้านทรัพยากรใดที่จะเพิ่มความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากที่สุด และควรระวังปัจจัยด้านทรัพยากรที่อาจทำให้ค่าความน่าจะเป็นลดลงได้

อย่างไรก็ตามจากรูปที่ 4.11 และ 4.12 จะสังเกตว่าการเรียงของปัจจัยด้านทรัพยากรของเงื่อนไขความเพียรพยายามและเงื่อนไขเวลาไม่เหมือนกัน ซึ่งสาเหตุอาจเกิดจากการพิจารณาตัวประกอบมาตราส่วนรวมกับตัวขับต้นทุน ทั้งที่จริงควรพิจารณาตัวประกอบมาตราส่วนแยกจากตัวขับต้นทุนเนื่องจากตัวประกอบมาตราและตัวขับต้นทุนนั้นคำนวณโดยใช้วิธีแตกต่างกัน

ตารางที่ 4.10 ลำดับการเรียงของตัวขับต้นทุนตามค่าช่วงห่างสำหรับเงื่อนไขความเพียรพยายามและเวลา

ตัวขับต้นทุน	ลำดับ	
	เงื่อนไขความเพียรพยายาม	เงื่อนไขเวลา
RELY	8	8
DATA	13	13
CPLX	2	1
RUSE	14	16
DOCU	7	7
TIME	15	14
STOR	16	15
PVOL	12	10
ACAP	1	2
PCAP	3	3
PCON	4	4
AEXP	6	6
PEXP	11	12
LTEX	10	11
TOOL	9	9
SITE	5	5

จากรูปที่ 4.11 และ 4.12 เมื่อพิจารณาเฉพาะการเรียงของตัวประกอบมาตราส่วนพบว่าตัวประกอบมาตราส่วนเรียงลำดับเหมือนกันทั้งเงื่อนไขความเพียรพยายามและเวลา คือ PMAT RESL PREC TEAM และ FLEX ตามลำดับ ดังนั้นจึงพิจารณาเฉพาะการเรียงของแต่ละตัวขับต้นทุนดังตารางที่ 4.10 โดยลำดับที่ 1 หมายถึงปัจจัยมีช่วงห่างของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จมากที่สุด อนึ่งเนื่องจาก SCED หรือกำหนดการของการพัฒนาคำนวณโดยใช้วิธีที่แตกต่างภายใต้สองเงื่อนไขนี้ทำให้มีลำดับที่ไม่เหมือนกันอย่างเห็นได้ชัดจึงไม่พิจารณา SCED ในการเรียงลำดับของตัวขับต้นทุนนี้ด้วย

เมื่อพิจารณาตารางที่ 4.10โดยคร่าวจะเห็นว่าลำดับของตัวขับต้นทุนที่แม้จะแตกต่างกันสำหรับเงื่อนไขความเพียรพยายามและเวลาก็มีค่าใกล้เคียงกัน เช่น CPLX ซึ่งอยู่ในลำดับที่ 2 สำหรับเงื่อนไขความเพียรพยายามซึ่งใกล้กับลำดับที่ 1 สำหรับเงื่อนไขเวลา ดังนั้นการที่ตัวขับต้นทุนนี้เรียงไม่สอดคล้องกันอาจเกิดเนื่องจากตัวขับต้นทุนบางตัวมีผลต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จแตกต่างกันน้อยมากทำให้ความคลาดเคลื่อนจากการจำลองสถานการณ์ส่งผลกระทบได้ ดังนั้นเพื่อทดสอบสมมุติฐานนี้จึงทดสอบว่าลำดับการเรียงของตัวขับต้นทุนมีความแตกต่างอย่างมีนัยสำคัญหรือไม่ในทางสถิติ โดยเลือกใช้วิธีเครื่องหมายลำดับที่ของวิลคอกซันสำหรับการทดสอบแบบจับคู่ (The Wilcoxon Signed Rank Sum Test for the Matched Paired Difference) เนื่องจากเป็นวิธีที่ใช้กับข้อมูลแบบสเกลลำดับ และเป็นการทดสอบแบบจับคู่ที่พิจารณาขนาดของผลต่าง การทดสอบคือ ลำดับของตัวขับต้นทุนสำหรับเงื่อนไขความเพียรพยายามแตกต่างจากเงื่อนไขเวลาที่ระดับนัยสำคัญ 0.05 หรือไม่ สมมุติฐานจึงเป็น

H_0: $M_A=M_B$

H_1: $M_A\neq M_B$

เมื่อ M_A และ M_B คือค่ามัธยฐานของลำดับในเงื่อนไขความเพียรพยายามและเงื่อนไขเวลาตามลำดับ โดยการทดสอบสมมุติฐานจะพิจารณาค่า *p* ซึ่งเป็นระดับนัยสำคัญที่น้อยที่สุดที่จะทำให้ปฏิเสธสมมุติฐาน H_0 ดังนั้นถ้าระดับนัยสำคัญที่ตั้งไว้มากกว่าหรือเท่ากับ *p* (หรือ *p* น้อยกว่าระดับนัยสำคัญที่ตั้งไว้) จะปฏิเสธสมมุติฐาน H_0 เนื่องจากในการทดลองแบบสองหางนี้ค่า *p* ที่ได้เท่ากับ 1.000 ซึ่งมากกว่าระดับนัยสำคัญที่ตั้งไว้คือ 0.05 จึงไม่สามารถปฏิเสธ H_0 ได้ สรุปคือลำดับของตัวขับต้นทุนสำหรับเงื่อนไขความเพียรพยายามและเวลาไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05 หรือที่ระดับความเชื่อมั่น 95%

นอกจากนี้เมื่อพิจารณารูปที่ 4.13 และ 4.14 พบว่าทั้งความน่าจะเป็นที่น้อยที่สุดและมากที่สุดต่างมีการเรียงของปัจจัยด้านทรัพยากรของเงื่อนไขความเพียรพยายามและเงื่อนไขเวลาไม่เหมือนเช่นกัน จึงทดสอบตามวิธีข้างต้นและพบว่าการเรียงตัวประกอบมาตราส่วนของทั้งสอง

เงื่อนไขเหมือนกันคือ PMAT RESL PREC TEAM และ FLEX ตามลำดับ สำหรับทั้งความน่าจะเป็นที่น้อยที่สุดและมากที่สุด และพบว่าลำดับการเรียงของตัวขับต้นทุนซึ่งแสดงดังตารางที่ 4.11 ไม่เหมือนกันในบางปัจจัย

ตารางที่ 4.11 ลำดับการเรียงของตัวขับต้นทุนตามความน่าจะเป็นสำหรับเงื่อนไขความเพียรพยายามและเวลา

ตัวขับต้นทุน	ลำดับ			
	ความน่าจะเป็นที่น้อยที่สุด		ความน่าจะเป็นที่มากที่สุด	
	เงื่อนไขความเพียรพยายาม	เงื่อนไขเวลา	เงื่อนไขความเพียรพยายาม	เงื่อนไขเวลา
RELY	10	9	9	9
DATA	8	8	13	13
CPLX	1	1	2	2
RUSE	9	10	14	14
DOCU	12	12	6	7
TIME	2	2	15	15
STOR	4	4	16	16
PVOL	6	6	12	12
ACAP	3	3	1	1
PCAP	5	5	3	3
PCON	7	7	7	6
AEXP	11	11	8	8
PEXP	15	15	11	11
LTEX	14	14	10	10
TOOL	16	16	5	5
SITE	13	13	4	4

หลังจากนั้นทดสอบทางสถิติด้วยวิธีเดิมได้ค่า p ของทั้งความน่าจะเป็นที่น้อยที่สุดและมากที่สุดเท่ากันคือ 1.00 ดังนั้นลำดับของตัวขับต้นทุนสำหรับเงื่อนไขความเพียรพยายามและเวลาของทั้งสองผลไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05 หรือที่ระดับความเชื่อมั่น 95%

จากการวิเคราะห์ทั้งหมดสรุปได้ว่าการเรียงของปัจจัยด้านทรัพยากรไม่แตกต่างกันอย่างมีนัยสำคัญสำหรับเงื่อนไขความเพียรพยายามและเวลา แต่การที่ปัจจัยเรียงไม่เหมือนกันนั้นเกิดจากการที่พิจารณาตัวประกอบมาตราส่วนและตัวขับต้นทุนรวมเข้าด้วยกัน และอาจเกิดจากผลกระทบจากความคลาดเคลื่อนจากการจำลองสถานการณ์ด้วย ซึ่งการทดสอบทางสถิติพบว่าการเรียงของปัจจัยโดยการแบ่งตามประเภทว่าเป็นตัวประกอบมาตราส่วนหรือตัวขับต้นทุนสำหรับเงื่อนไขความเพียรพยายามและเวลานี้ไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05 หรือที่ระดับความเชื่อมั่น 95%

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

จากการทดลองใช้แบบจำลองประเมินความเสี่ยงสำหรับการจัดสรรทรัพยากรในโครงการเดียวและแบบจำลองต้นทุน COCOMO II สำหรับการจัดสรรทรัพยากรในกลุ่มโครงการสามารถสรุปผลการวิจัยและข้อเสนอแนะดังนี้

5.1 สรุปผลการวิจัย

1. แบบจำลองประเมินความเสี่ยงสามารถนำมาประยุกต์ใช้สำหรับการจัดสรรทรัพยากรในโครงการเดียวได้ โดยสามารถใช้ตัดสินใจทรัพยากรเชิงภาวะถ่วงดุล และสามารถใช้เป็นพื้นฐานในการหารูปแบบการจัดสรรทรัพยากรที่เหมาะสมที่สุดได้

2. แบบจำลองต้นทุน COCOMO II สามารถช่วยเพิ่มปัจจัยด้านทรัพยากรในกลุ่มโครงการนอกเหนือจากเส้นตายและจำนวนพนักงานสำหรับปรับเปลี่ยนความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จได้

3. จากการทดลองประโยชน์จากการบริหารแบบกลุ่มโครงการพบว่าการบริหารแบบกลุ่มโครงการเป็นวิธีการที่มีประโยชน์สำหรับกลุ่มโครงการที่มีความไม่แน่นอนสูง

4. การวิเคราะห์ความไวของความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ ต่อปัจจัยด้านทรัพยากรแสดงให้เห็นว่าแต่ละปัจจัยส่งผลมากน้อยเพียงใดต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ โดยวิธีการนี้สามารถประยุกต์ใช้สำหรับการตัดสินใจเลือกปรับเปลี่ยนปัจจัยทรัพยากรแก่ผู้บริหารได้ กล่าวคือผู้บริหารสามารถศึกษาว่าปัจจัยทรัพยากรใดในกลุ่มโครงการของผู้บริหารส่งผลมากน้อยเพียงใดต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ

5.2 ข้อเสนอแนะ

1. การใช้แบบจำลองประเมินความเสี่ยงสำหรับหารูปแบบการจัดสรรทรัพยากรที่เหมาะสมที่สุดอาจดัดแปลงใช้ในสถานการณ์อื่น ๆ เช่น สถานการณ์ที่ค่าตอบแทนคำนวณจากจำนวนของความต้องการ สถานการณ์ที่เวลาที่ใช้ในการพัฒนาไม่สามารถเกินกำหนดการได้ สถานการณ์ที่พัฒนาเสร็จล่าช้าแล้วต้องเสียค่าปรับ เป็นต้น

2. ในการวิเคราะห์ความไวผู้บริหารอาจใช้ COCOMO II แบบ Early Prototyping หรือ Early Design แทน Post-Architecture ตามความเหมาะสมกับโครงการ

3. ในการวิเคราะห์ความไวอาจเลือกปรับปัจจัยด้านทรัพยากรเฉพาะบางโครงการแทนที่จะปรับในทุกโครงการ เพื่อดูว่าการปรับในเฉพาะบางโครงการส่งผลอย่างไรต่อความน่าจะเป็นที่กลุ่มโครงการจะสำเร็จ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Mendes, E., Mosley, N. and Watson, I. 2002. A Comparison of Case-based Reasoning Approaches to Web Hypermedia Project Cost Estimation. Proceedings of the 11th International World-Wide Web Conference.

2. Shepperd, M., Schofield, C. and Kitchenham, B. 1996. Effort Estimation Using Analogy. Proceedings of the 18th International Conference on Software Engineering.

3. Fenton, N., Marsh, W., Neil, M., Cates, P., Forey, S. and Tailor, M. 2004. Making Resource Decisions for Software Projects. Proceedings of the 26th International Conference on Software Engineering.

4. Fenton, N.E. and Pfleeger, S.L. 1997. Software Metrics: A Rigorous and Practical Approach. International Thomson Computer Press. Boston: PWS.

5. Fewster, R.M. and Mendes, E. 2003. Portfolio Management Method for Deadline Planning. Proceedings of the 9th International Software Metrics Symposium.

6. Pressman, R.S. 2005. Software Engineering: A Practitioner's Approach. (n.p.): McGraw-Hill.

7. Jiamthubthugsin, W. and Sutivong, D. 2006. Resource Decisions in Software Development Using Risk Assessment Model. The 39th Hawaii International Conference on System Sciences.

8. Jiamthubthugsin, W. and Sutivong, D., Portfolio Management of Software Development Projects Using COCOMO II. The 28th International Conference on Software Engineering.

9. Keil, M., Cule, P.E., Lyytinen, K. and Schmidt, R. C. 1998. A Framework for Identifying Software Project Risks. Communications of the ACM 41: 76-83.

10. Boehm, B.W. Software Risk Management: Principles and Practices. IEEE Software 8, 1: 32-41.

11. Boehm, B.W. 2000. Software Cost Estimation with COCOMO II. Upper Saddle River: Prentice Hall.

12. Boehm, B., Clark, B., Horowitz, E. and Westland, C. 1995. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0. Annals of Software Engineering 1: 57-94.

13. Gilb, T. and Finzi, S. 1988. Principles of Software Engineering Management. Wokingham: Addison-Wesley.

14. Boehm, B.W. 1988. A Spiral Model of Software Development and Enhancement. Computer 21, No. 5: 61-72.

15. Braude, E. J. 2001. Software Engineering: An Object-Oriented Perspective. New York: Wiley.

16. Putnam, LH. 1978. A General Empirical Solution to the Macrosoftware Sizing and Estimating Problem. IEEE Transactions on Software Engineering 4: 345-361.

17. Nogueira, J. and Luqi, 2000. A Risk Assessment Model for Evolutionary Software Projects. Monterey Workshop on Modelling Software System Structures in a Fastly Moving Scenario.

18. Nogueira, J.C., Luqi and Bhattacharya, S. 2000. A Risk Assessment Model for Software Prototyping Projects. Proceedings of the 11th International Workshop on Rapid System Prototyping.

19. Nogueira, J.C., Luqi, Valdis, B. and Nader, N. 2000. A Formal Risk Assessment Model for Software Evolution. The 2nd International Workshop on Economics-Driven Software Engineering.

20. Foo, Say-Wei and Muruganantham, A. 2000. Software Risk Assessment Model. Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology.

21. Madachy, R. J. 1997. Heuristic Risk Assessment Using Cost Factors. IEEE Software 14, 3: 51-59.

22. Roy, G.G. 2004. A Risk Management Framework for Software Engineering Practice. Proceedings of Australian Software Engineering Conference.

23. McDonald, S., and Wildman. 2003. NOSTROMO: Using Monte Carlo Simulation to Model Uncertainty Risk in COCOMO II. The 18th International Forum on COCOMO and Software Cost Modeling.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

24. Boehm, B. W. 1981. Software Engineering Economics. Englewood Cliffs: Prentice-Hall.

25. Putnam, L. and Myers, W. 1997. Industrial Strength Software. IEEE Computer Society.

26. กวี บูรพสายันห์. 2547. แบบจำลองการประมาณต้นทุนซอฟต์แวร์สำหรับโครงการพัฒนาซอฟต์แวร์. โครงงานมหาบัณฑิต. ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

27. McDonald, P., Giles, S. and Strickland, D. 2004. Extensions of Auto-Generated Code and NOSTROMO Methodologies. The 19th International Forum on COCOMO and Software Cost Modeling.

28. Madachy, R. and Stutzke, R. 2002. Use of Cost Models in Risk Management. The 17th International Forum on COCOMO and Software Cost Modeling.

29. Chulani, S., Boehm, B. and Steece, B. 1999. Bayesian Analysis of Empirical Software Engineering Cost Models. IEEE Transactions on Software Engineering 25, 4: 573-583.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายวิบูลย์ เจียมทับทักษิณ เกิดเมื่อวันที่ 2 เมษายน พ.ศ. 2525 ที่จังหวัดนครปฐม สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ (เกียรตินิยมอันดับ 1) จากภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อปี 2547 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ (แผน ก) ภาควิชาวิศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2547

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย