



การประมาณการเวลาในการพัฒนาซอฟต์แวร์ประยุกต์เชิงโครงข่าย โดยวิธีแบบจำลองสมการโครงสร้าง

สมชาย ปรากฏการเจริญ* อุดม จินประดับ** มนต์ชัย เทียนทอง*** และ ราชันย์ บุญธิมา****

บทคัดย่อ

ปัจจุบันในการพัฒนาซอฟต์แวร์จะมีลักษณะเป็นซอฟต์แวร์ประยุกต์เชิงโครงข่าย เทคนิค สูตรที่ใช้ในการประมาณเวลาในการพัฒนาซอฟต์แวร์เดิมๆ เช่น COCOMO, SLIM ไม่สามารถประมาณเวลาดังกล่าวได้ถูกต้อง ในที่นี้ผู้วิจัยได้ทดลองจัดสร้างแบบจำลองในการประมาณการเวลาในการพัฒนาซอฟต์แวร์ที่มีลักษณะเป็นซอฟต์แวร์ประยุกต์เชิงโครงข่าย (Transaction processing web based application) จากกลุ่มตัวอย่างภาคเอกชนและภาครัฐบาล โดยวิธีการแบบจำลองสมการโครงสร้าง กลุ่มตัวอย่างรวบรวมจากโครงการพัฒนาซอฟต์แวร์ประยุกต์เชิงโครงข่ายของหน่วยพัฒนาซอฟต์แวร์จำนวน 208 โครงการ จากแบบจำลองที่พัฒนาขึ้น ผู้วิจัยได้นำไปสอบถามความถูกต้องกับโครงการอื่น ๆ อีกจำนวน 30 โครงการ ผลการสอบทานพบว่า แบบจำลองให้ความผิดพลาดในการประมาณการเวลาในการพัฒนาซอฟต์แวร์ประมาณ 150.77% ในขณะที่ค่าความผิดพลาดในการประมาณเวลา โดยวิธีการอื่น ๆ เช่น COCOMO-II ให้ความผิดพลาดประมาณ 171.21% และแบบจำลองมีความแม่นยำในการพยากรณ์ (R^2) เท่ากับ 96.6%

คำสำคัญ: การประมาณราคา การพัฒนาซอฟต์แวร์เชิงโครงข่าย วิธีแบบจำลองสมการโครงสร้าง

1. บทนำ

วิธีการประมาณการเวลาในการพัฒนาซอฟต์แวร์ที่เป็นที่ยอมรับใช้งานกันอย่างแพร่หลาย เช่น COCOMO, SLIM ได้พัฒนาสมการการประมาณการจากซอฟต์แวร์ที่มีลักษณะเชิงกระบวนการ (Functional oriented) ในปัจจุบันรูปแบบของซอฟต์แวร์มีความแตกต่างจากเดิมกล่าวคือ มีลักษณะเป็น

ซอฟต์แวร์ประยุกต์เชิงโครงข่าย (Web based application) ทำให้สูตรสมการข้างต้น ประมาณการผิดพลาดค่อนข้างสูง [1] จึงมีผู้วิจัยได้เสนอแนวคิดต่าง ๆ ในการพัฒนาแบบจำลองให้แม่นยำยิ่งขึ้นได้แก่ [2], [3], [4] ได้พัฒนาแบบจำลองจากสมการถดถอย ส่วน [5], [6] ได้เสนอการประมาณการขนาดของซอฟต์แวร์ที่มีลักษณะเชิงโครงข่ายด้วย “Web object” แทน “Function point”

การวิจัยนี้มุ่งพัฒนาแบบจำลองในการประมาณการเวลาโดยใช้วิธีการจำลองสมการโครงสร้าง (Structural Equation Modelling : SEM) โดยได้ทำการรวบรวมข้อมูลหาตัวแปรประจักษ์ที่เกี่ยวข้องกับการประมาณการเวลาจากผู้พัฒนาซอฟต์แวร์ทั้งภาคเอกชนและราชการจำนวนรวม 208 โครงการ จากนั้นตัวแปรประจักษ์เหล่านี้จะถูกประมวลผลด้วยการวิเคราะห์ปัจจัย [7] เพื่อจัดกลุ่มตัวแปรประจักษ์เป็นตัวแปรปัจจัย ปัจจัยเหล่านี้จะนำไปวิเคราะห์หาแบบจำลองสมการโครงการสร้างด้วย SEM [8] แบบจำลองที่พัฒนาได้จะทำการทดสอบหาค่าเฉลี่ยความผิดพลาดเชิงสัมพัทธ์ (Mean Magnitude of Relative Error : MMRE) โดยทำการทดสอบกับโครงการพัฒนาซอฟต์แวร์ที่รวบรวมมาอีก จำนวน 30 โครงการ ในขณะเดียวกัน 30 โครงการดังกล่าวจะถูกนำไปคำนวณหา MMRE โดยเทคนิคของ COCOMO-II [9] การเปรียบเทียบค่า MMRE ของทั้งสองวิธี จะใช้ในการแสดงความแม่นยำในการพยากรณ์เวลาของทั้งสองแบบจำลองต่อไป

2. วัตถุประสงค์ของการวิจัย

2.1 เพื่อพัฒนาแบบจำลองการประมาณการเวลาในการพัฒนาซอฟต์แวร์ประยุกต์เชิงโครงข่ายโดยวิธีการแบบจำลองสมการโครงสร้าง

2.2 เพื่อประมาณการความแม่นยำในการพยากรณ์ของ

* คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

*** คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

** สำนักคอมพิวเตอร์และเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**** มหาวิทยาลัยศรีนครินทรวิโรฒ

แบบจำลอง

2.3 เพื่อเปรียบเทียบความแม่นยำในการพยากรณ์ของแบบจำลองที่พัฒนาขึ้นกับวิธีการ COCOMO-II

3. วิธีการดำเนินการวิจัย

ในการวิจัยเพื่อพัฒนาแบบจำลองการประมาณเวลาในการพัฒนาซอฟต์แวร์ประยุกต์เชิงโครงข่ายมีขั้นตอนดังนี้

3.1 ศึกษาวรรณกรรมอ้างอิง งานวิจัยที่เกี่ยวข้อง โดยพิจารณา รวบรวม ค้นหา ตัวชี้วัดที่ได้มีการเผยแพร่ว่าเป็นตัวชี้วัดในการประมาณการเวลาที่ เหมาะสม และจำเป็นต่อการพัฒนาแบบจำลองการประมาณการเวลาซอฟต์แวร์ประยุกต์เชิงโครงข่ายจากแหล่งสำคัญๆ

3.2 ทำการจัดหมวดหมู่ตัวชี้วัดต่างๆและทำการวิเคราะห์วิจารณ์ แก้ไข ปรับปรุง โดยผู้เชี่ยวชาญจากภาคเอกชน และภาคราชการ จำนวนรวม 30 คน โดยใช้วิธีเดลฟายเทคนิค รวมสองหน ผลการปรับปรุงแก้ไขเพิ่มเติมทำให้ได้ตัวชี้วัดจำนวน 8 หมวด 38 ตัวชี้วัด ดังนี้

หมวดที่ 1 ตัวชี้วัดเกี่ยวข้องกับซอฟต์แวร์ที่จะพัฒนา (Product)

- 1) ขนาดของซอฟต์แวร์ (size, web object)
- 2) ความน่าเชื่อถือ (Reliability)
- 3) ความซับซ้อน (Complexity)
- 4) ประสิทธิภาพ – ความเร็ว (Performance)
- 5) ขนาดของฐานข้อมูล (Database Size)
- 6) รูปแบบ โครงสร้างของซอฟต์แวร์และความเป็น

อิสระ (Modularity)

- 7) การป้องกันและตรวจสอบข้อมูล (Data Verification & Validation)

8) การรักษาความปลอดภัยซอฟต์แวร์ (Web Application Security)

- 9) ความสะดวกพึงพอใจต่อการใช้งาน (Web-Usability)
- 10) ความสะดวกในการติดตั้งการใช้งาน (Configurability)

หมวดที่ 2 ตัวชี้วัดเกี่ยวข้องกับบุคลากร (Personnel)

1) ความสามารถของผู้พัฒนาโปรแกรม (Programmer Capability)

2) ประสบการณ์ของผู้พัฒนาโปรแกรม (Programmer Experience)

3) ความสามารถของผู้วิเคราะห์ระบบ (Analyst capability)

- 4) ประสบการณ์ของผู้วิเคราะห์ระบบ (Analyst

Experience)

- 5) ความร่วมแรงร่วมใจในทีมงาน (Team Cohesion)

6) ความมั่นคงในทีมงานต่อการพัฒนาโปรแกรม (การคงอยู่ – การลาออกงาน) (Team Stability)

7) ประสบการณ์ของทีมงานต่อระบบงานที่จะพัฒนา (Team Application Experience)

8) ความสามารถของผู้วางโครงสร้างสถาปัตยกรรม (Web Architecture Capability)

9) ความสามารถของผู้ออกแบบ (Web Designer Capability)

หมวดที่ 3 กรรมวิธีการพัฒนาซอฟต์แวร์ (S/W development method)

1) รูปแบบกระบวนการพัฒนาซอฟต์แวร์ (Water fall, Evolutionary, Reuse-based)

2) ระดับการพัฒนาความสามารถของกระบวนการพัฒนาซอฟต์แวร์ (Capability Maturity Model)

3) การวิเคราะห์ความต้องการของระบบ (Requirement Engineering)

หมวดที่ 4 ฐานระบบงาน (Platform)

1) ระดับความยากของฐานระบบงาน (Platform difficulty เช่น Client/Server, Three tier, Distributed Object)

2) ความแตกต่างของระบบซอฟต์แวร์ที่เป็นปัญหาในการแลกเปลี่ยนข้อมูลระหว่างระบบที่พัฒนากับระบบงานอื่น ๆ (Interoperability /Heterogeneity)

3) ความยากของภาษาที่ใช้ในการพัฒนา (Language difficulty)

4) การปรับเปลี่ยนฐานระบบงาน (Platform Volatility)

หมวดที่ 5 การบริหารจัดการโครงการ (Management/Project)

1) ประสบการณ์ของผู้บริหารโครงการ (Project manager experience)

2) จำนวนโครงการที่ประสบความสำเร็จ (% of success project – complete)

3) สิ่งอำนวยความสะดวกในการทำงาน (Infrastructure /Resource availability & dedicated to project)

หมวดที่ 6 โปรแกรมสนับสนุนการพัฒนาซอฟต์แวร์ (CASE Tool)

1) ประสบการณ์ในการใช้โปรแกรมสนับสนุน (CASE Tool Experience)



- 2) ฟังก์ชันของ CASE tools (CASE tool functions)
- 3) ประเภทของ CASE Tool (CASE tool Category)
- 4) โปรแกรมสนับสนุนการออกแบบเว็บ (Web

Design/Develop tool)

หมวดที่ 7 ข้อจำกัดโครงการ (Constraint)

- 1) ความเร่งรัดของกำหนดการ/ระยะเวลาโครงการในการพัฒนาซอฟต์แวร์ (Schedule)
- 2) ความสะดวกในการเปลี่ยนแปลงความต้องการได้ (Requirement Volatility)
- 3) การจัดหาทรัพยากรทางด้านวิชาการและการสนับสนุนด้านเทคนิค (Resource/technical support)

หมวดที่ 8 ตัวชี้วัดอื่นๆ

- 1) ความสามารถในการพัฒนาจากโครงสร้างเดิม (Reusable Project reusability)
- 2) แผนงานการพัฒนาซอฟต์แวร์ (Web blue Print)

3.3 จัดสร้างแบบสอบถามเพื่อรวบรวมข้อมูลตามตัวชี้วัดข้างต้น ทำการทดสอบความถูกต้อง ความเข้าใจ ความต้องตรงตามวัตถุประสงค์ของแบบสอบถามต่อวัตถุประสงค์ของงานวิจัยโดยการทำการสอบถามจากกลุ่มตัวอย่างทดลองตอบแบบสอบถามจำนวน 10 คน เพื่อให้ข้อเสนอแนะในการปรับปรุง แก้ไข แบบสอบถาม แบบสอบถามที่ได้รับการปรับปรุงแล้วได้ถูกนำไปใช้ในการรวบรวมข้อมูลจากกลุ่มตัวอย่างผู้พัฒนาซอฟต์แวร์ประยุกต์เชิงโครงข่าย ซึ่งรวบรวมจากแหล่งซอฟต์แวร์เฮาส์ และผู้พัฒนาซอฟต์แวร์อิสระรวมจำนวน 186 คน หากนับเป็นโครงการจะได้เท่ากับ 219 โครงการ

3.4 ตรวจสอบความถูกต้องของแบบสอบถาม หลังจากรวบรวมแบบสอบถามแล้วได้มีการตรวจสอบความถูกต้องของแบบสอบถาม และคัดออกแบบสอบถามที่มีลักษณะการตอบที่ผิดปกติ หรือมีความแตกต่างจากกลุ่มตัวอย่างทั่วไป (Outlier) ทำให้คงเหลือแบบสอบถามที่เหมาะสมเข้าสู่การจัดทำแบบจำลองจำนวน 208 โดยที่โครงการทั้งหมดเป็นโครงการขนาดเล็กถึงกลาง (โดยพิจารณาจากเวลาระหว่าง 10,000 – 1,000,000 บาท) รวม 208 โครงการนี้จะถูกนำไปพัฒนาแบบจำลอง และนอกจากนี้ผู้วิจัยได้รวบรวมเพิ่มเติมอีก 30 โครงการสำหรับการสอบทานแบบจำลอง (Model Cross-validation)

3.5 ดำเนินการพัฒนาแบบจำลอง

3.5.1 การวิเคราะห์ปัจจัย (Factor analysis) เป็น

เทคนิคที่ใช้ในการจัดสร้างปัจจัย (Factor หรือ Component) ใหม่ขึ้นโดยการรวบรวมตัวชี้วัดที่มีความสัมพันธ์กันมาอยู่รวมกัน การที่ตัวแปรอิสระมีความสัมพันธ์กัน (Multi-collinearity) จะทำให้การประมาณค่า ตัวแปรตาม (y) ในการวิเคราะห์สมการถดถอยผิดพลาดได้ นั่นคือการวิเคราะห์ปัจจัยจะช่วยให้การจัดกลุ่มตัวแปรอิสระเข้าด้วยกันก่อน (เฉพาะกลุ่มที่มีความสัมพันธ์กันสูงแบบมีนัยสำคัญ) จากนั้นนำปัจจัยเหล่านี้มาวิเคราะห์สมการถดถอย จะทำให้ได้ค่าประมาณการ 'y' ที่ถูกต้องยิ่งขึ้น โดยที่ค่าปัจจัยจะหาได้จาก

$$F_i = \sum_{j=1}^n w_{ij} x_j \quad (1)$$

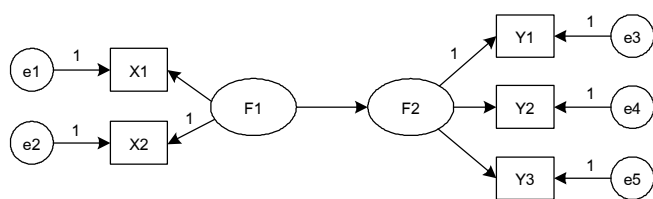
โดยที่ F_i หมายถึง ปัจจัยตัวที่ i (Factor Score)

w_{ij} หมายถึง ค่า weight (หรือ ค่า Factor Coefficients) ของปัจจัยตัวที่ i ตัวแปรอิสระที่ j

x_j หมายถึง ตัวแปรอิสระที่ j

การวิเคราะห์ปัจจัยใช้หลักการรวบรวมตัวแปรอิสระที่มีค่าความสัมพันธ์ต่อกันสูง การวิเคราะห์ภาพรวมของความสัมพันธ์ที่ตัวแปรอิสระทุกตัวมีต่อกันว่าสูงพอต่อการนำไปจัดสร้างเป็นปัจจัยหรือไม่ จะพิจารณาจากค่า KMO (Kaiser – Meyer – Olhin of sampling accuracy) หากค่า KMO มากกว่า 0.6 ขึ้นไปจะถือว่าอยู่ในเกณฑ์ที่สูงพอต่อการวิเคราะห์ปัจจัย [7]

3.5.2 วิธีการแบบจำลองสมการโครงสร้าง (Structural equation modeling) [8] เป็นเทคนิคที่ใช้ในการพิจารณาหาความสัมพันธ์ของกลุ่มตัวแปรแฝง (Latent variable) โดยแสดงภาพออกมาเป็นแบบจำลองสมการโครงสร้าง (Structural Modeling) และใช้ในการแสดงความสัมพันธ์ของตัวแปรแฝงต่อตัวแปรประจักษ์ (Observed variable) โดยแสดงออกมาในแบบจำลองคุณลักษณะแบบจำลอง (Model specification) การแสดงภาพความสัมพันธ์ทั้งหมดจะถูกตรวจสอบหาความถูกต้องของโครงสร้างโดยพิจารณาจากความสามารถอธิบายความผันแปรของทุก ๆ ตัวแปร โดยแบบจำลอง ให้มีค่าใกล้เคียงกับความผันแปรโดยรวมของทุก ๆ ตัวแปร (Residual variance) หรือมีความแตกต่างกันน้อยที่สุด



ภาพที่ 1 ตัวอย่างแบบจำลองสมการโครงสร้าง

จากภาพที่ 1 ตัวแปร x_1 , x_2 เป็นตัวแปรประจักษ์ที่มีความเกี่ยวข้องกับตัวแปรแฝง F_1 ความสัมพันธ์ต่อกันเรียก F_1 Model Specification ในขณะที่ตัวแปร y_1 , y_2 , y_3 เป็นตัวแปรประจักษ์ที่มีความเกี่ยวข้องกับตัวแปรแฝง F_2 ความสัมพันธ์ต่อกันเรียก F_2 Model Specification ส่วน F_1 มีความสัมพันธ์โดยตรง (อิทธิพล) ต่อ F_2 ความสัมพันธ์ F_1 กับ F_2 เรียกเป็น F_1 - F_2 Structural equation การตรวจสอบความเหมาะสมของแบบจำลองที่ตั้งสมมติฐานขึ้น จะตรวจสอบจากค่าดัชนีต่าง ๆ ได้แก่ Chi-square: χ^2 ($p > 0.05$), Goodness of fit index: GFI (> 0.9), Adjusted Goodness of fit index: AGFI (> 0.9), Root mean square error of approximation: RMSEA (< 0.05) [8] ในการพิจารณาว่าตัวแปรประจักษ์ใดสามารถเป็นตัวชี้วัดค่าตัวแปรแฝงได้อย่างเหมาะสม จะพิจารณาจากค่า Cronbach's alpha (> 0.7)

3.6 การประมาณความแม่นยำ (Evaluation Criterion)

แบบจำลองที่สร้างขึ้นต้องมีความแม่นยำเข้ากันได้ (Model best fit) กับข้อมูลที่ใช้ในการสร้างแบบจำลองนั้นสูงสุด เช่น ค่า R^2 ในการวิเคราะห์สมการถดถอย หลังจากนั้นแบบจำลองนี้จะถูกนำไปทดสอบกับกลุ่มข้อมูลอีกชุดหนึ่งที่ทราบเวลาจริง (Actual time) ผลจากการพยากรณ์ข้อมูลชุดใหม่ (Predicted time) จะถูกนำมาคำนวณค่าความคลาดเคลื่อนสัมพัทธ์ (Magnitude of relative error – MRE) ดังสมการ(2)

$$MRE_i = \frac{|Actual\ cost_i - Predicted\ cost_i|}{Actual\ cost_i} \quad (2)$$

หากมีหลายโครงการจะนำค่าความคลาดเคลื่อนสัมพัทธ์ของแต่ละโครงการมารวมกันและหาค่าเฉลี่ยดังสมการ

$$MMRE = \frac{1}{n} \sum_{i=1}^n \left[\frac{|Actual\ cost_i - Predicted\ cost_i|}{Actual\ cost_i} \right] \times 100 \quad (3)$$

หากค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (Mean – MRE : MMRE) มีค่าสูงหมายถึงเปอร์เซ็นต์ของความคลาดเคลื่อนสูง

หากค่า $MMRE = 0$ หมายถึงค่าของการพยากรณ์เท่ากับค่าจริงทุกๆ ค่า นั่นคือ $MMRE$ ยิ่งน้อย หมายถึงมีความแม่นยำสูง

3.7 วิเคราะห์เวลาในการพัฒนาโปรแกรมโดยวิธี COCOMO II การประมาณการเวลาโดยสูตร 'COCOMO – II' Boehm [9] ได้พัฒนาสูตรที่ใช้ในการประมาณการเวลาในการพัฒนาซอฟต์แวร์ไว้ มีสูตรดังนี้

$$PM = A \times Size^B \times M$$

และ

(4)

$$DT = 3 \times (PM)^{(0.33+0.2 \times (B-1.01))}$$

โดยที่ PM หมายถึง person month (man month)

DT หมายถึง development time (month)

A หมายถึง ค่าคงที่เท่ากับ 2.5

$Size$ หมายถึง ขนาดของ software นับหน่วยหลักพัน

B หมายถึง ค่าความผกผันขึ้นกับตัวแปร 5 ตัว

(Precedentedness, Development flexibility, Architecture / risk resolution, Team cohesion และ Process maturity)

M หมายถึง ค่าวิฤตุน ค่าวนจากตัวแปร 7 ตัว

(Product reliability and Complexity, Reused required, Platform difficulty, Personal capability, Personal experience, Schedule, Support)

การพัฒนาสมการดังกล่าว ได้พัฒนาให้ใช้งานบนภาษาคอมพิวเตอร์ที่มีลักษณะการ 'Reuse' ในยุคต้นๆ ของการโปรแกรมเชิงวัตถุ หากนำสูตรดังกล่าวมาใช้ในการประมาณการเวลาในการพัฒนาซอฟต์แวร์ประยุกต์เชิงโครงข่าย อาจก่อให้เกิดความผิดพลาดสูง Reifer [6] จึงได้นำเสนอการประมาณการ 'Size' โดย 'Web – object' ซึ่งจะสามารถประมาณการขนาดซอฟต์แวร์ที่มีลักษณะเชิงโครงข่ายได้ดียิ่งขึ้น

3.8 สรุปผลเปรียบเทียบความแม่นยำของการพยากรณ์เพื่อหาแนวทางที่เหมาะสมที่สุดในการประมาณการเวลา

4. ผลการทดลอง

ผลของการดำเนินการวิจัยของแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

4.1 สถิติพื้นฐานของกลุ่มตัวอย่าง

จากกลุ่มตัวอย่าง 208 โครงการ ค่าสถิติพื้นฐานของตัวแปรเวลาในการพัฒนาซอฟต์แวร์ และขนาดของซอฟต์แวร์



(Web object) มีดังนี้

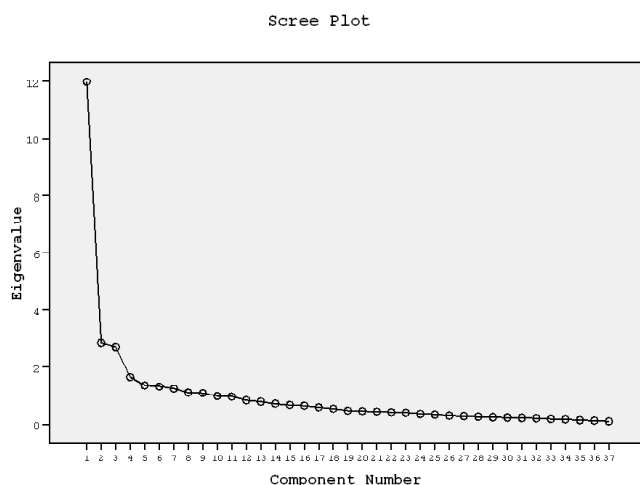
ตารางที่ 1 ค่าสถิติพื้นฐานของตัวแปรเวลา และขนาดของซอฟต์แวร์

	Mean	SD	MIN	MAX
Cost	194,807.88	16,581.91	10,000	1,000,000
Schedule	6.87 (เดือน)	0.51	1	36
Web object	601.66	48.81	25	3,089

4.2 แบบจำลองการประมาณเวลาและค่าสถิติที่เกี่ยวข้อง
คณะผู้วิจัยได้ทดลองประมวลผลโดยโปรแกรมสำเร็จรูปทางสถิติ (SPSS V.15) โดยใช้เทคนิคต่างๆ ได้ผลดังนี้

4.2.1 แบบจำลองการวิเคราะห์ปัจจัย (Factor analysis)

ผลการวิเคราะห์ปัจจัยโดยเทคนิค “Principle Component Analysis -PCA” และหมุนแกนแบบ “Quartimax” จำนวนรอบในการคำนวณซ้ำทั้งหมด 29 รอบ ได้ค่าปัจจัยจำนวน 8 ตัว ดังตารางที่ 2 ค่าสถิติ Kaiser Meyer – Olkin เท่ากับ 0.879 แสดงว่าจำนวนกลุ่มตัวอย่างมีจำนวนมากพอต่อการสกัดปัจจัย และค่า Barlett's test of Sphericity มีค่า $X^2 = 4,530.828$ Sig = 0.000 นั่นคือ ค่าเมทริกซ์สหสัมพันธ์ของตัวแปรประจักษ์มีลักษณะไม่สมมาตร (Identical) สามารถนำมาสกัดปัจจัยได้ดีกราฟแสดงความเหมาะสมของจำนวนปัจจัยระบุจำนวนของปัจจัยเท่ากับ 8 ค่า แสดงดังภาพที่ 2



ภาพที่ 2 Screenplot แสดงจำนวนปัจจัยที่ได้จากการสกัดปัจจัย

องค์ประกอบของตัวแปรประจักษ์ต่าง ๆ ต่อแต่ละปัจจัย แสดงในตารางที่ 2 ค่า Cronbach's alpha แสดงถึงความ

เหมาะสมในการจัดกลุ่มปัจจัย โดยปัจจัยที่เหมาะสมคือ F1 เนื่องจากมีค่า Cronbach's alpha สูงสุด (0.883) เว้นแต่ F7 (0.61)

ตารางที่ 2 ผลการสกัดปัจจัยและค่าเปอร์เซ็นต์สะสมการอธิบายความผันแปร โดยปัจจัย Cronbach's alpha

ปัจจัย	ตัวแปรประจักษ์	การอธิบายความผันแปร	Cronbach's alpha
F1	V2_1, V2_2, V2_3, V2_4, V5_1	9.98%	0.883
F2	V2_5, V2_6, V2_7, V2_8, V2_9, V5_3	19.37%	0.850
F3	V1_2, V1_4, V1_7, V1_8, V1_9, V1_10	28.65%	0.850
F4	V4_1, V4_2, V4_3, V4_4, V1_3, V1_5	37.85%	0.801
F5	V1_6, V3_1, V3_2, V3_3, V5_2	44.91%	0.817
F6	V6_1, V6_2	51.38%	0.875
F7	V6_3, V6_4	56.68%	0.610
F8	V8_1, V8_2, V1_1	61.88%	0.742
F9	V7_1, V7_2, V7_3	67.02%	0.721

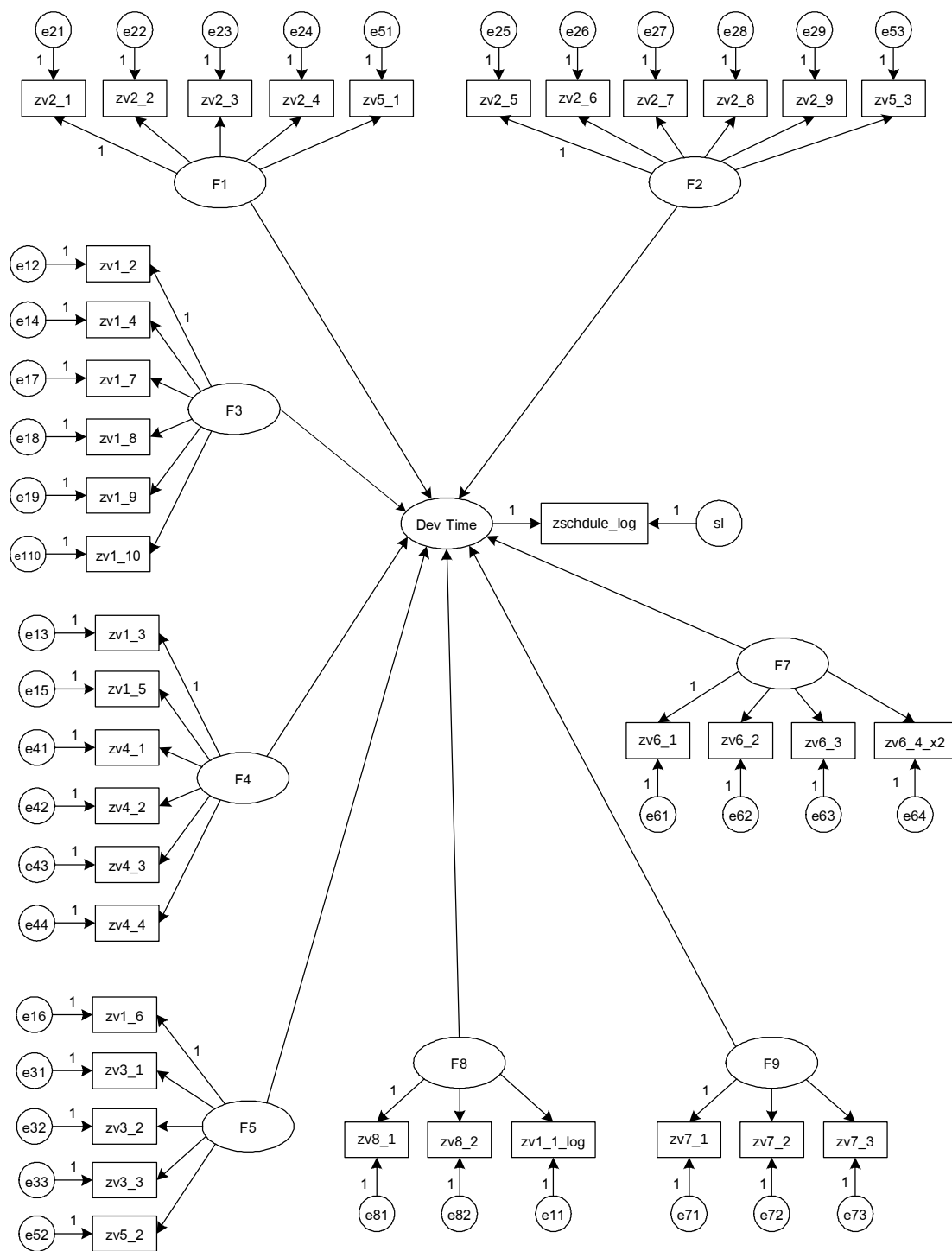
4.2.2 แบบจำลองสมการโครงสร้าง

ค่าปัจจัยจากการวิเคราะห์แบบจำลองได้ถูกนำไปพัฒนาจำลองสมการโครงสร้างโดย Statistical package AMOS-6 ได้ แบบจำลองสมการโครงสร้างเพื่อประมาณการเวลาในการพัฒนาซอฟต์แวร์ ดังภาพที่ 3

ค่า Regression weights ของปัจจัยต่างๆ ที่ได้จากการวิเคราะห์แบบจำลองสมการโครงสร้างต่อตัวแปรแฝง Devtime (Schdule) แสดงในตารางที่ 3

ตารางที่ 3 ค่า Regression Weight ระหว่างตัวแปรแฝง (Devtime) และปัจจัย

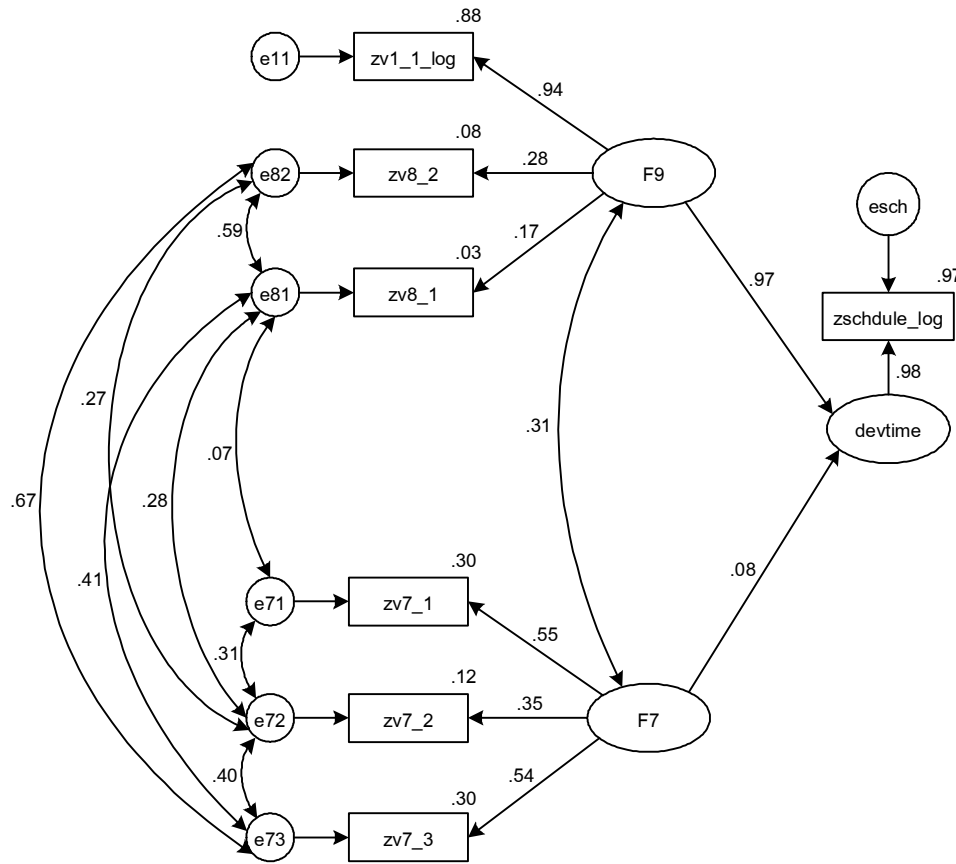
Regression Weights: (Group number 1 - Default model)					
		Estimate	S.E.	C.R.	P
DevTime	<--- F1	-.053	.037	-1.421	.155
DevTime	<--- F3	-.068	.043	-1.591	.112
DevTime	<--- F4	.125	.050	2.531	.011
DevTime	<--- F5	.113	.055	2.068	.039
DevTime	<--- F8	5.548	2.357	2.354	.019
DevTime	<--- F9	.121	.047	2.578	.010
DevTime	<--- F7	.010	.031	.309	.757
DevTime	<--- F2	-.041	.041	-.986	.324



ภาพที่ 3 แบบจำลองสมการโครงสร้างเต็มรูป (Model 1)

ปัจจัย F7 มีระดับนัยสำคัญต่อตัวแปรแฝง Devtime ที่ $p=0.973$ (non significance) นั่นคือปัจจัย F7 มีความน่าจะเป็นในการถูกตัด (Trim) ออกจากแบบจำลองมากที่สุด ผู้วิจัยได้ดำเนินการตัดปัจจัยและ/หรือตัวแปรประจักษ์โดยพิจารณาจากค่า Regression Weights ข้างต้น ท้ายสุดได้แบบจำลองสมการโครงสร้าง ดังภาพที่ 4

ผลการตรวจสอบความเหมาะสมของแบบจำลองของทั้งสองแบบจำลองภาพที่ 3 และภาพที่ 4 แสดงในตารางที่ 4 จากผลการเปรียบเทียบแบบจำลองโครงสร้างเต็มรูป (Model 1) พบว่าไม่มีความเข้าได้กับข้อมูล (χ^2 มีค่า Significance) ในขณะที่แบบจำลองที่ 2 ให้ความเหมาะสมเข้ากันได้ดีกับข้อมูลที่ใช้ในการสร้างแบบจำลอง เนื่องจากค่า χ^2 มีค่า



ภาพที่ 4 แบบจำลองสมการโครงสร้างประมาณการเวลา(Model 2)

ตารางที่ 4 ค่าการตรวจสอบความเหมาะสมของแบบจำลองโครงสร้าง (Goodness of fit Indices)

Model	Chisquare	Df	P	GFI	AGFI	RMSEA
Model 1	2376.01	695	0.00	0.57	0.52	0.11
Model 2	8.73	6	0.19	0.99	0.95	0.04

non – significance, GFI > 0.9, AGFI > 0.9 และ RMSEA < 0.05

จากภาพที่ 4 การประมาณค่า Devtime (Schdule) สามารถดำเนินการพัฒนาจากตัวแปรประจักษ์ ZV1_1_log, ZV8_1, ZV8_2, ZV7_1, ZV7_2, ZV7_3 และ ZSchdule_log ค่า ZV1_1_log เป็นค่ามาตรฐานของ V1_1 ทั้งนี้เพื่อการแปลงค่า (Transform) โดยการแปลงค่า $\log_{10}$ ของ V1_1 ทั้งนี้เพื่อให้ค่า V1_1 มีค่ามาตรฐาน (Standardized) และเป็นค่าปกติ (Normalized) ทำนองเดียวกันกับค่า Schdule_log

การนำผลการพัฒนาแบบจำลองไปใช้งานได้แก่การกำหนดสมการประมาณค่า Devtime (Schdule) โดยมีสมการดังสมการที่ (5)

$$\text{Devtime} = 0.973 \text{ F9} + 0.0800 \text{ F7} \quad (5)$$

$$R^2 = 0.966$$

ส่วนค่าของตัวแปรแฝง F7, F9 จาก measurement model กำหนดค่าได้ดังนี้

$$\text{F7} = 0.55 \text{ ZV7_1} + 0.35 \text{ ZV7_2} + 0.54 \text{ ZV7_3} \quad (6)$$

$$\text{F9} = 0.94 \text{ ZV1_1_log} + 0.17 \text{ ZV8_1} + 0.28 \text{ ZV8_2} \quad (7)$$

4.3 การประมาณการเวลาของโครงการพัฒนา

ซอฟต์แวร์ที่พัฒนาแล้วเสร็จโดยวิธี COCOMO – II นอกจากนั้น 30 โครงการพัฒนาซอฟต์แวร์ซึ่งแล้วเสร็จไปทำการประมาณการเวลาในสมการที่ (4) แล้วโครงการดังกล่าวได้ถูกนำไปคำนวณเวลาในการพัฒนาซอฟต์แวร์โดยสูตร COCOMO – II ซึ่งมีค่าที่ต้องกำหนดขึ้น ตามความต้องการของแบบจำลอง COCOMO – II ในสองสิ่งคือ

4.3.1 ค่าความผกผัน (B) ในงานวิจัยนี้ได้เทียบเคียงตัวแปรที่ใช้ประกอบในการคำนวณหาค่า 'B' จากแบบสอบถามดังนี้

ตารางที่ 5 การเทียบเคียงตัวแปรค่าความผกผันกับค่าตัวชี้วัดในแบบสอบถาม

ตัวแปรใน COCOMO (B)	ตัวชี้วัดจากแบบสอบถาม
1. Precedentedness	V2_6 (2.6)
2. Development flexibility	V3_2 (3.2)
3. Architecture/risk resolution	V3_3 (3.3)
4. Team cohesion	V5_2 (5.2)
5. Process Maturity	V7_2 (7.2)

4.3.2 ค่าทวีคูณ (Multiplier – M)

การเทียบเคียงตัวแปรที่ใช้ประกอบการคำนวณค่า 'M' กับตัวชี้วัดในแบบสอบถามได้ผลดังตารางที่ 6

ตัวแปร (B, M) จะใช้ตัวชี้วัดจากแบบสอบถาม ซึ่งรวบรวมจากโครงการ 30 โครงการไปคำนวณค่า B, M เพื่อแทนลงในสมการ (4) ต่อไป

4.4 การทดสอบ – เปรียบเทียบ ความแม่นยำในการ

สืบค้นเทียบเคียง เวลาของโครงการที่นำมาประมาณการโดยวิธี 4.2 สมการ 4 และจากการประมาณการเวลาโดยวิธี COCOMO-II

ตารางที่ 6 การเทียบเคียงตัวแปรค่าทวีคูณกับตัวชี้วัดในแบบสอบถาม

ตัวแปรใน COCOMO (M)	ตัวชี้วัดจากแบบสอบถาม
1. Product reliability and Complexity	(V1_2+V1_3)/2
2. Reused required	V8_1
3. Platform difficulty	V4_1
4. Personal capability	(V2_1+V2_3+V2_8+V2_9)/4
5. Personal experience	(V2_2+V2_4+V2_7+V5_1)/4
6. Schedule	V7_1
7. Support	N/A

ใน (4.3) ผลที่ได้ (เวลา) จะนำไปเปรียบเทียบกับค่าแท้จริงของโครงการนั้น ๆ เพื่อคำนวณหา MMRE ผลของการวิเคราะห์สรุปผลได้ดังนี้

ตารางที่ 7 ผลการเปรียบเทียบ MMRE ของวิธีการแบบจำลองสมการโครงสร้าง(4.2.2) และวิธี COCOMO-II(4.3)

	4.2 (สมการ 4)	4.3
MMRE	150.77%	171.21%
PRED(L)	23.33%	0%

$$\text{Pred(L)} = K/N \quad (8)$$

โดยที่ L หมายถึง ระดับร้อยละของการพยากรณ์ผิดพลาดที่ไม่ต้องการให้เกิน เช่น 50%

K หมายถึง จำนวนตัวอย่างที่มีค่าพยากรณ์ภายใต้เกณฑ์ 'L'

N หมายถึง จำนวนตัวอย่างทั้งหมดที่นำมาคำนวณ Cross Validation

5. สรุปผลการวิจัย

จากผลการวิจัย ผู้วิจัยได้รวบรวมตัวชี้วัด จากงานวิจัยต่างๆ และผ่านการพิจารณาจากผู้เชี่ยวชาญ ทำให้ได้ตัวชี้วัดจำนวนทั้งสิ้น 38 ตัว ใน 8 หมวด หลังจากนั้นทำการจัดหมวดหมู่ตัวแปรใหม่เป็น เก้าปัจจัย (F1-F9) ปัจจัยเหล่านี้ได้ถูกนำไปพัฒนาแบบจำลองสมการโครงสร้าง ดังปรากฏในภาพที่ 3 แต่เมื่อผ่านการพิจารณาคัดออก (Trim) ตัวชี้วัดหรือปัจจัยที่ไม่มีนัยสำคัญทางสถิติ ($\alpha = 0.05$) แล้ว แบบจำลองสมการโครงสร้างที่เหมาะสมและประหยัด (parsimony) มากที่สุด และมีความเข้ากันได้กับข้อมูล (Model Goodness of fit) สูงสุดแสดงได้ดังภาพที่ 4

จากแบบจำลองนี้ได้นำไปสร้างเป็นสมการในการประมาณการเวลา (Development time) ดังปรากฏในสมการที่ (4), (5) และ(6) ซึ่งสมการดังกล่าว เป็นการประมาณค่าต่างๆ ในรูปแบบของค่ามาตรฐาน (Standardize equation) และตัวแปรประจักษ์ V1_1 ถูกแปลงค่า (Transform) ให้เป็นค่าปกติ (Normality) สมการดังกล่าวได้นำไปใช้ในการพยากรณ์เวลา กับกลุ่มตัวอย่างซึ่งทราบเวลาจริงในการพัฒนาซอฟต์แวร์แล้วจำนวน 30 โครงการ เวลาจากการพยากรณ์และเวลาจริงถูกนำมาคำนวณหาความผิดพลาดเชิงสัมพัทธ์เฉลี่ย (MMRE) พบว่าความผิดพลาดในการประมาณการมีค่าเพียง 150.77% และเมื่อเปรียบเทียบความผิดพลาดในการประมาณการเวลา



ของวิธี COCOMO-II พบว่า วิธีการ COCOMO-II มีความผิดพลาดในการประมาณการสูงมากถึง 171.21% นอกจากนั้นร้อยละของกลุ่มตัวอย่างที่ใช้ในการตรวจสอบโดยสมการแบบจำลองโครงสร้าง ให้ความผิดพลาดจากการประมาณการไม่เกิน 50% มีจำนวนถึง 23.33% ของกลุ่มตัวอย่าง ในขณะที่การประมาณการแบบ COCOMO-II ไม่มีตัวอย่างใดเลยที่ทำให้การประมาณการผิดพลาดไม่เกิน 50% จากเวลาจริง

6. ปัญหาและข้อเสนอแนะ

จากผลการวิจัย คณะผู้วิจัยได้สรุปผลไว้ดังนี้

6.1 ตัวชี้วัดจำนวน 38 ตัว ที่ใช้ในการพยากรณ์เวลาตาม ที่รวบรวมจากงานวิจัยต่างๆ ยังครอบคลุมไม่เพียงพอ โดยพบว่าตัวชี้วัดที่มีผล – อิทธิพล มีเพียง 5 ตัวชี้วัด อาจจะมีตัวชี้วัดอื่นๆ อีก

6.2 การพัฒนาแบบจำลองสมการโครงสร้าง สร้างจากกลุ่มตัวอย่าง 208 โครงการ และสอบทานการประมาณ เพื่อหาความผิดพลาดในการประมาณการจากกลุ่มตัวอย่าง 30 โครงการ จำนวนตัวอย่างรวบรวมได้ไม่มากนัก เนื่องจากความลับทางธุรกิจ ถึงแม้จำนวนกลุ่มตัวอย่างจะมีจำนวนเกินเกณฑ์ขั้นต่ำของการพัฒนาแบบจำลอง แต่การเพิ่มกลุ่มตัวอย่างมากขึ้นจะก่อให้เกิดความมั่นใจยิ่งขึ้นในผลการทดลอง

7. เอกสารอ้างอิง

- [1] Zhihao chen, “Finding the right data for software cost modeling”, University of southern california, USA, 2005.
- [2] Martin Sheppard, “Effort estimating using analogy”, Bournemouth university, USA, 1996.
- [3] Sarah Jane deleny, “The application of CASE base reasoning to early software project cost estimation”, Trinity college, UK, 2000.
- [4] Emillia Mendes, “Web metrics estimating design and authoring effort”, University of Auckland, Newzealand, 2001.
- [5] Donald Riefer, “System level metrics for software development estimation”, University of southern califonia, USA, 2002.
- [6] Donald Riefer, “Web object counting conventions”, Reifer consultants, Inc., CA , USA, 2001.
- [7] Classes M., “Factor analysis tutorial”, Sam Houston university, USA, 2005.
- [8] Arbuckle, James L. “AMOS6.0 user guide”, Chicago Illinois, SPSS Inc., USA, 2006.
- [9] Barry Boehm, “COCOMO tutorial”, University of southern califonia, 2004.