

บทที่ 5

สรุปผลการศึกษาและข้อเสนอแนะ

5.1 สรุปผลการศึกษา

การศึกษาค้นคว้าครั้งนี้ ได้ใช้หลักการของ CMMI มาช่วยในการวิเคราะห์หารากของปัญหา ซึ่งนำไปสู่วิธีการแก้ไขปัญหาคำสั่ง ซึ่งวิธีการแก้ปัญหาคำสั่ง คือ การแก้ไขคำสั่งโปรแกรมตามหลักการพัฒนาโปรแกรมที่ดี โดยหลังจากการแก้ไขแล้วทำให้โปรแกรมที่เกิดปัญหาใช้เวลาการทำงานมากกว่า 1 วัน สามารถประมวลผลได้สำเร็จภายในเวลา 1 วันตามที่กำหนด และโปรแกรมที่เกิดปัญหาประมวลผลไม่สำเร็จสามารถกลับมาทำงานได้ และใช้เวลาการทำงานตามเวลาที่กำหนดเช่นกัน ซึ่งผลการแก้ไขปัญหาคำสั่งในครั้งนี้ทำให้ผู้ใช้งานเกิดความพึงพอใจมากขึ้น เนื่องจากสามารถปฏิบัติงานในชีวิตประจำวันได้ตามปกติ

จากการวิเคราะห์สาเหตุที่เกิดจากเครื่องมือ คือ ขาดแนวทางการพัฒนาโปรแกรมที่ดี กลายมาเป็นสิ่งที่ได้รับจากการศึกษารวบรวมแนวทางการแก้ปัญหาด้านเทคนิคโปรแกรม คือ หลักการที่ดีในการพัฒนาโปรแกรม ซึ่งผู้ให้คำปรึกษาและทีมพัฒนาโปรแกรมได้จัดทำเป็นบัญชีสำหรับตรวจสอบประสิทธิภาพของโปรแกรกดังตารางที่ 5.1 เพื่อใช้เป็นเอกสารในการตรวจสอบคำสั่งโปรแกรม และได้มอบให้กับแผนกสารสนเทศของบริษัทแห่งนี้ได้ใช้ประโยชน์ต่อไป รวมทั้งทางบริษัทของผู้ให้คำปรึกษาเองจะได้ใช้ประโยชน์จากเอกสารชิ้นนี้ในการพัฒนาโปรแกรมสำหรับโครงการอื่นๆด้วย

วิธีการแก้ปัญหาคำสั่งด้วยการแก้ไขคำสั่งโปรแกรมเป็นการแก้ไขปรับปรุงในส่วนปลายทางของผลิตภัณฑ์ แต่จากการศึกษาหาสาเหตุของปัญหาทำให้พบสาเหตุที่เกิดจากบุคคลากรและกระบวนการ ซึ่งสามารถหาแนวทางการป้องกันการเกิดปัญหาในลักษณะนี้ได้ ผู้ให้คำปรึกษาจะขอเสนอแนวทางการป้องกันการเกิดปัญหาไว้ในส่วนของข้อเสนอแนะ

ตารางที่ 5.1

เอกสารบัญชีสำหรับตรวจประสิทธิภาพ

(Performance Checklist)

Program Name			
Program Version			
Technical Designer			
Programmer			
Performance Reviewer			
Approve By			
คำแนะนำในการจัดทำเอกสาร: - สำหรับการแนบโปรแกรม และเอกสารการวัดเวลาการทำงานของโปรแกรมให้แนบเป็นวัตถุในช่องการแก้ไข (เมนู Insert -> Object) - สำหรับการแก้ไขโปรแกรมให้คัดลอกโปรแกรมก่อนการแก้ไข และหลังการแก้ไข พร้อมทั้งระบุเลขที่บรรทัดของคำสั่งในโปรแกรมด้วย หมายเหตุ: สถานะของรายการมีดังนี้ A: สมบูรณ์ B: ตรวจสอบซ้ำ และรอการเซ็นอนุมัติ N/A: โปรแกรมไม่มีได้ตรงกับรายการการตรวจสอบ			
รายการการตรวจสอบ	สถานะ	การแก้ไข	ลงวันที่
วัดเวลาการทำงานของโปรแกรมด้วยเครื่องมือ ABAP Trace (SE30)			
กำจัดชุดคำสั่งที่ไม่ได้ใช้งานออกจากโปรแกรม			
<u>คำสั่งที่ใช้กับฐานข้อมูล</u>			
ใช้คำสั่ง SELECT แบบระบุ Field แทน SELECT *			

ตารางที่ 5.1 (ต่อ)
เอกสารบัญชีสำหรับตรวจประสิทธิภาพ
(Performance Checklist)

รายการการตรวจสอบ	สถานะ	การแก้ไข	ลงวันที่
เงื่อนไขใน Where Clause เป็นไปตาม Key หรือ Index ของตารางที่อ่าน			
การเปรียบเทียบเงื่อนไขใน Where Clause ให้ใช้ AND และ EQ			
ใช้คำสั่ง SELECT...INTO TABLE แทน SELECT...ENDSELECT			
ใช้คำสั่ง SELECT...FOR ALL ENTRIES แทน Nested Select			
ใช้คำสั่ง JOIN แทน Nested Select			
ใช้คำสั่ง SELECT...UP TO 1 ROWS แทน SELECT SINGLE หากเงื่อนไขในการอ่านข้อมูลไม่ครบตาม Key			
หลีกเลี่ยงการใช้คำสั่ง SELECT ร่วมกับ Aggregated Function			
อ่านข้อมูลจาก Database View แทนการอ่านข้อมูล 2 ตารางด้วยคำสั่ง JOIN			
ใช้คำสั่ง INSERT, UPDATE, หรือ DELETE FROM Internal Table			
<u>คำสั่งที่ใช้กับ Internal Table</u>			
กำหนดค่าเริ่มต้นด้วยคำสั่ง OCCURS 0			
ใช้คำสั่ง APPEND แทน INSERT			
ใช้คำสั่ง REFRESH และ FREE ทั้งก่อนและหลังการใช้งานตัวแปร			

ตารางที่ 5.1 (ต่อ)
เอกสารบัญชีสำหรับตรวจประสิทธิภาพ
(Performance Checklist)

รายการการตรวจสอบ	สถานะ	การแก้ไข	ลงวันที่
ใช้คำสั่ง READ TABLE...WITH KEY...BINARY SEARCH			
ใช้คำสั่ง LOOP แบบระบุ Index			
ลดการใช้ตัวแปร Global แล้วปรับมาใช้ ตัวแปร Local			
ใช้คำสั่ง ITAB2[] = ITAB1[] เพื่อคัดลอก ข้อมูล			
ใช้คำสั่ง DELETE ITAB...WHERE			
ใช้ตัวแปร Field Symbol แทน Header Line			
ใช้คำสั่ง SORT... ASCENDING หรือ DESCENDING			
ใช้คำสั่ง CASE...WHEN...WHEN...ENDCASE แทน IF...ELSEIF...ELSEIF...ENDIF			

5.2 ข้อจำกัดในการศึกษา

การศึกษาผลกระทบของวิธีการแก้ไขด้วยการสร้างดรรชนีฐานข้อมูล ผู้ให้คำปรึกษา
และทีมพัฒนาโปรแกรมไม่สามารถทำการทดลองได้เอง ต้องขอความร่วมมือจากทีมผู้ดูแลระบบ
มาช่วยในการศึกษาเปรียบเทียบ และพิจารณาผลกระทบที่อาจเกิดขึ้นกับระบบ เพราะการสร้าง

ดรชนี้ฐานข้อมูลทำให้ทีมผู้ดูแลระบบต้องทำการปรับเปลี่ยนค่าพารามิเตอร์บางอย่างของฐานข้อมูล ดังนั้นการแก้ไขด้วยวิธีนี้จึงไม่ได้นำมาใช้

5.3 ข้อเสนอแนะสำหรับการให้คำปรึกษาในอนาคต

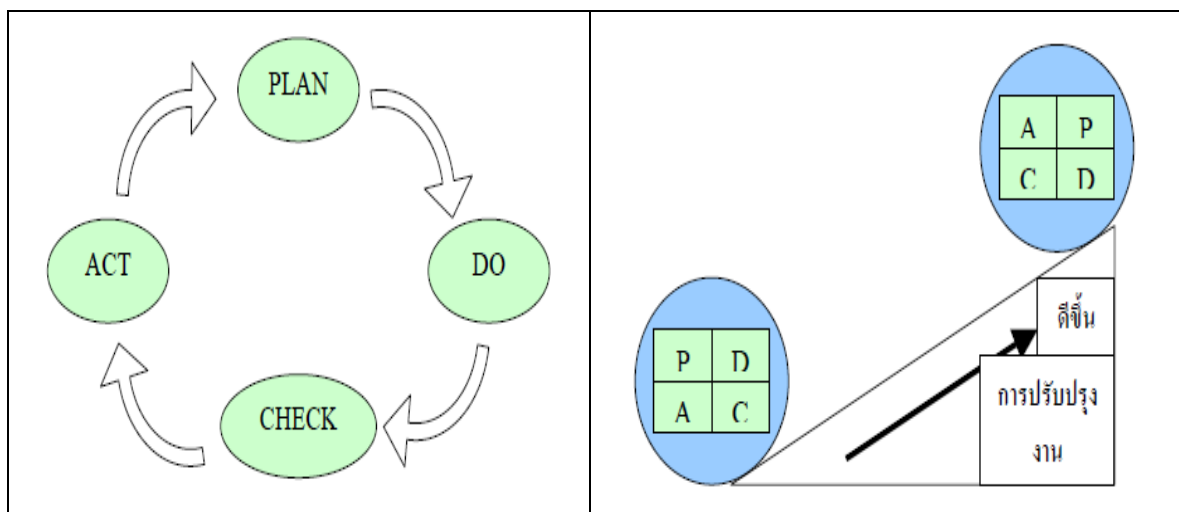
ในการให้คำปรึกษาครั้งนี้ยังคงมีประเด็นที่ต้องนำไปศึกษาและทดลองต่อไป ดังที่ได้กล่าวไว้ในข้อจำกัดในการศึกษา สำหรับข้อเสนอแนะจะขอเสนอแนวทางการป้องกันการเกิดปัญหาที่ทางทีมพัฒนาโปรแกรมได้ระดมสมองร่วมกัน ในส่วนของการปรับปรุงกระบวนการพัฒนาโปรแกรม ซึ่งแนวทางนี้จะนำไปใช้ในการพัฒนาโปรแกรมของโครงการถัดไป โดยปรับปรุงจากกระบวนการพัฒนาโปรแกรมแบบเดิมไปเล็กน้อย แสดงการปรับปรุงกระบวนการทำงานได้ดังภาพที่ 5.1 ขั้นตอนการทำงานที่เสนอให้ปรับปรุงมีดังนี้

1. ขั้นตอนการทบทวนเอกสารออกแบบการทำงานของโปรแกรม แนะนำให้เพิ่มความเข้มงวดการทบทวนเอกสาร เนื่องจากในการทำงานจริงบางครั้งไม่ได้พิจารณาอย่างถี่ถ้วน แล้วจึงทำการอนุมัติไป นอกจากนี้ให้เพิ่มเติมส่วนของผู้รับผิดชอบในการทบทวนเอกสาร โดยให้หัวหน้าทีมพัฒนาโปรแกรม หรือผู้พัฒนาโปรแกรมอาวุโสช่วยตรวจสอบเรื่องทางเทคนิคที่อาจมีผลกระทบต่อประสิทธิภาพการทำงานของโปรแกรม
2. ขั้นตอนการสร้างเอกสารออกแบบทางเทคนิคโปรแกรม จะต้องทำก่อนการพัฒนาโปรแกรม ส่วนใหญ่ในการทำงานจริงจะเป็นเอกสารที่ทำตามหลังจากการพัฒนาโปรแกรมเสร็จไปแล้ว ซึ่งขัดกับขั้นตอนที่ควรจะเป็น
3. ขั้นตอนการทบทวนเอกสารออกแบบทางเทคนิคโปรแกรม ควรเน้นเช่นเดียวกับการทบทวนเอกสารออกแบบโปรแกรม ไม่ควรละเลยเหมือนที่ผ่านมา
4. ขั้นตอนการพัฒนาโปรแกรม ให้หัวหน้าทีมพัฒนาโปรแกรมชี้แจงหลักการพัฒนาโปรแกรมตามบัญชีตรวจสอบประสิทธิภาพการทำงานของโปรแกรม และให้ผู้พัฒนาโปรแกรมปฏิบัติตามนั้น
5. ขั้นตอนการทบทวนโปรแกรม จากเดิมที่ไม่มีขั้นตอนนี้ ให้หัวหน้าทีมพัฒนาโปรแกรมหรือผู้พัฒนาโปรแกรมอาวุโสช่วยทบทวนโปรแกรมก่อนส่งไปทดสอบ
6. ขั้นตอนการทดสอบโปรแกรมโดยผู้ออกแบบการทำงานของโปรแกรม โดยปกติจะทดสอบเพียงความถูกต้องของการทำงานตามที่ได้ออกแบบไว้ ให้เพิ่มการทดสอบเรื่องประสิทธิภาพการทำงานของโปรแกรมโดยให้ผู้พัฒนาโปรแกรมเป็นผู้รับผิดชอบร่วมกัน

นอกจากในส่วนของกระบวนการที่ควรปรับปรุงแล้ว ยังขอเสนอแนะให้ปรับปรุงในเรื่องของงานเอกสาร โดยใช้หลักการการบริหารงานคุณภาพเข้ามาช่วยในการจัดการเรื่องข้อผิดพลาดต่างๆ ของโปรแกรม โดยจัดทำเป็นเอกสารในลักษณะของ Non-conformity Report (NCR) เป็นรายงานหรือเอกสารที่แจ้งถึงข้อบกพร่องของโปรแกรมซึ่งทำให้โปรแกรมไม่สามารถทำงานได้ตามความต้องการของผู้ใช้ และ Corrective Action Request (CAR) เป็นรายงานหรือเอกสารที่ร้องขอให้แก้ไขข้อบกพร่องของโปรแกรม โดยมีการวิเคราะห์หาสาเหตุของข้อบกพร่องเหล่านั้น เพื่อขจัดที่สาเหตุ และป้องกันการเกิดข้อบกพร่องเดิมๆ ซ้ำอีก ซึ่งทางผู้ให้คำปรึกษาคิดว่าเอกสารการบันทึกข้อบกพร่องเหล่านี้จะช่วยให้ผู้ดูแลบำรุงรักษาระบบ สามารถนำมาใช้ในอนาคตหากโปรแกรมเกิดปัญหาในลักษณะเดิมขึ้น และช่วยให้การบำรุงรักษาระบบได้มีการพัฒนาอย่างต่อเนื่องตามหลักการ PDCA Cycle (Deming, 1950) โดย “P” ย่อมาจาก Plan “D” ย่อมาจาก Do “C” ย่อมาจาก Check และ “A” ย่อมาจาก Act ดังภาพที่ 5.1 และคำอธิบายในตารางที่ 5.2 ในส่วนของเอกสารบัญชีสำหรับตรวจประสิทธิภาพของโปรแกรมยังคงต้องมีการปรับปรุงต่อไปสำหรับเรื่องที่จะใช้ในการเป็นแนวทางในการพัฒนาโปรแกรมและรูปแบบของเอกสาร

ภาพที่ 5.1

PDCA Cycle



ที่มา: “เดมมิง (Dr. W. Edwards Deming)” โดย กิตติวัฒน์ สิริเกษมสุข, ธเนศ กิตติธาดาชัยกุล, และอนุศาสตร์ แพบทาม

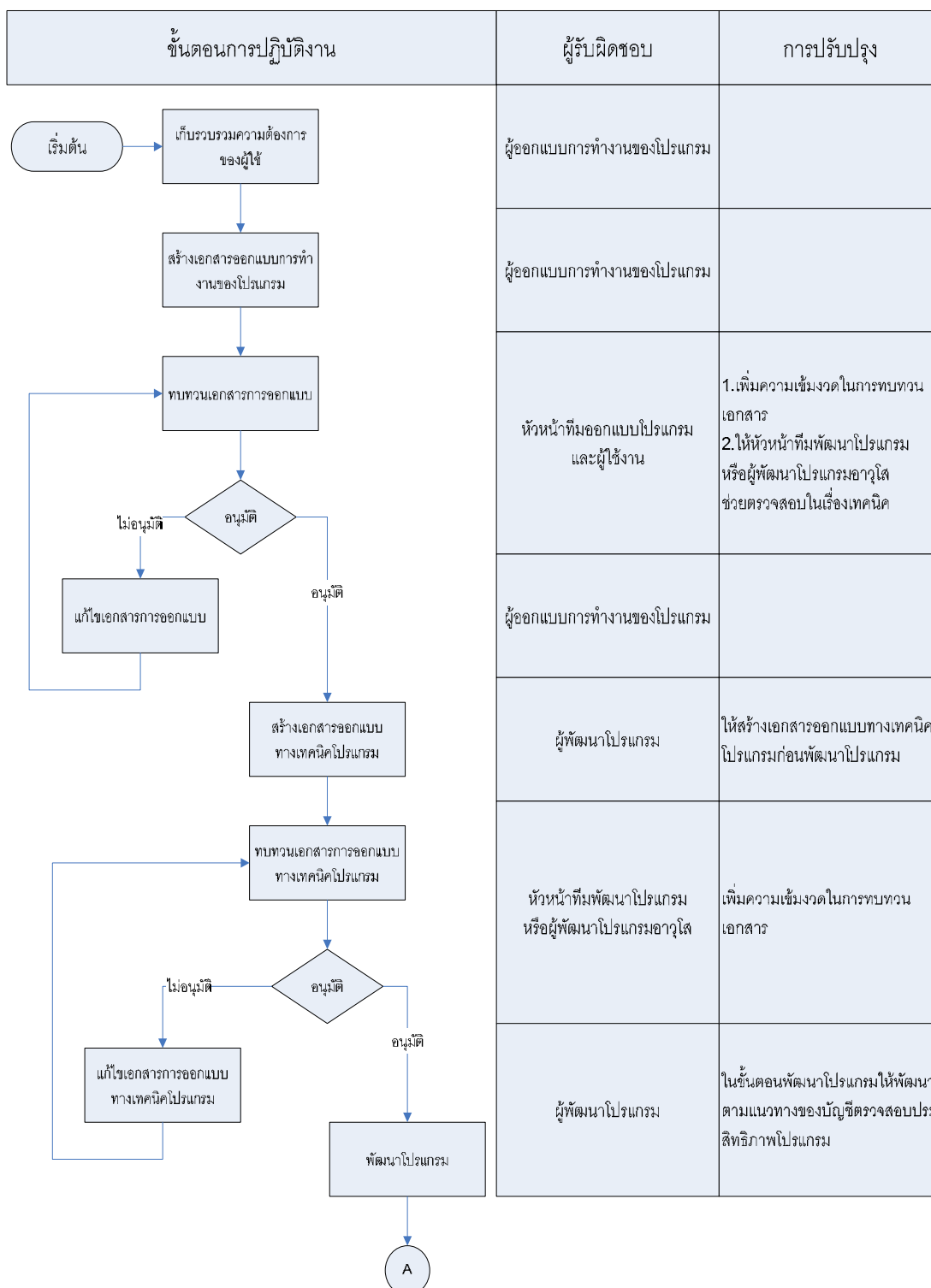
ตารางที่ 5.2
คำอธิบายของ PDCA Cycle

กิจกรรม	คำอธิบาย
“P” PLAN การวางแผน ,การกำหนดแผน	ออกแบบและกำหนดสิ่งที่จะต้องทำ ผลิตต้นแบบและทดสอบ (หลังจากที่ทราบความต้องการของลูกค้าแล้ว)
“D” DO ลงมือทำตามแผนที่ได้วางไว้	ผลิตตามแบบ
“C” CHECK ตรวจสอบผลลัพธ์กับแผน	ตรวจสอบดูว่าสินค้าที่ผลิตได้มีคุณภาพตามความต้องการหรือ ข้อกำหนดของลูกค้าหรือไม่
“A” ACT - หากไม่บรรลุแผนให้สาเหตุ และวางแผนแก้ไขใหม่ (เริ่มทำ วงจร PDCA ใหม่) - หากบรรลุแผนให้กำหนดเป็น มาตรฐานเพื่อยึดถือปฏิบัติ ต่อไป	ตรวจสอบดูว่าลูกค้าพอใจหรือไม่ - ถ้ามีข้อบกพร่องก็แก้ไขซ่อมแซมโดยส่งข้อมูลป้อนกลับไปยัง กระบวนการเดิม - ถ้าเป็นที่พอใจ ก็บันทึกไว้เป็นมาตรฐานวิธีทำงาน

ที่มา: “เดมมิง (Dr. W. Edwards Deming)” โดย กิตติวัฒน์ สิริเกษมสุข, ธเนศ กิตติธาดาชัยกุล,
และอนุศาสตร์ แพบทาม

ภาพที่ 5.2

การปรับปรุงกระบวนการพัฒนาโปรแกรม



ภาพที่ 5.2 (ต่อ)

การปรับปรุงกระบวนการพัฒนาโปรแกรม

