บทที่ 5
สรุปผลการศึกษา อภิปรายผลการศึกษา ข้อค้นพบและข้อเสนอแนะ

การศึกษาเรื่อง กระบวนการพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงาน ของบริษัท คอนโทรลดาต้า (ประเทศไทย) จำกัด มีวัตถุประสงค์เพื่อศึกษากระบวนการพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงาน โดยรวบรวมข้อมูลจากการสัมภาษณ์ทีมงานพัฒนาโปรแกรมทั้งหมดจำนวน 3 คน สัมภาษณ์ผู้ใช้งานโปรแกรมจากประชากรทั้งหมดจำนวน 3 คน และทำการศึกษาจากเอกสารที่เกี่ยวข้องกับการพัฒนาโปรแกรม สามารถสรุปผลการศึกษาได้ 2 ส่วนดังนี้

ส่วนที่ 1 ข้อมูลที่ได้จากการสัมภาษณ์ทีมงานพัฒนาโปรแกรม

ส่วนที่ 2 ข้อมูลที่ได้จากการสัมภาษณ์ผู้ใช้โปรแกรม

ส่วนที่ 3 สรุปผลการศึกษาเปรียบเทียบ SDLC กับมาตรฐาน CMMI

สรุปผลการศึกษา

ส่วนที่ 1 ข้อมูลที่ได้จากการสัมภาษณ์ทีมงานพัฒนาโปรแกรม

จากการสัมภาษณ์ทีมงานพัฒนาโปรแกรมพบว่า กระบวนการในการพัฒนาโปรแกรมในขั้นตอนต่างๆเป็นไปตามวงจรการพัฒนาระบบงาน (SDLC) ในระยะต่างๆดังนี้

ระยะที่ 1 การวางแผนโครงการ (Project Planning Phase)

1) กำหนดปัญหา (Problem Definition) ผู้จัดการโครงการ (Project Manager) ได้รับทราบปัญหาที่เกิดจากดำเนินงานในปัจุบันของผู้ใช้พร้อมทั้งทำการสรุปประเด็นปัญหาและแนวทางแก้ไขเพื่อนำไปศึกษารายละเอียดการพัฒนาระบบงานต่อไป

2) ศึกษาความเป็นไปได้ (Feasibility Study) นักวิเคราะห์ระบบ (System Analyst) ได้ทำการศึกษาความเป็นไปได้จาก 2 ประเด็นหลักได้แก่ ความเป็นไปได้ทางเทคนิคและความเป็นไปได้ในการปฏิบัติงาน ซึ่งจากการศึกษาความเป็นไปได้นั้น นักวิเคราะห์ระบบพบว่ามีความเป็นไปได้ทั้งทางเทคนิคและความเป็นไปได้ในการปฏิบัติงาน จึงดำเนินการในขั้นตอนต่อไป

3) จัดทำตารางกำหนดเวลาโครงการ (Project Scheduling) ผู้จัดการโครงการและนักวิเคราะห์ระบบได้จัดทำแผนการพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงาน พร้อมทั้งนำแผนดังกล่าวเสนอต่อผู้ที่เกี่ยวข้องเพื่อขอความเห็นชอบ ซึ่งแผนดังกล่าวได้รับความเห็นชอบจากผู้ที่เกี่ยวข้องจึงดำเนินการในขั้นตอนต่อไป

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ระยะที่ 2 การวิเคราะห์ระบบ (Analysis Phase)

1) วิเคราะห์ระบบงานปัจจุบัน นักวิเคราะห์ระบบได้ทำการศึกษาระบบการเบิกค่าใช้จ่ายเบ็ดเตล็ด (Petty Cash) ระบบการเบิกค่าใช้ประจำเดือน (Monthly Expense) และระบบใบคำขอ (Request) เพื่อทำความเข้าใจระบบ ตลอดจนหาจุดเชื่อมโยงข้อมูลที่เหมาะสมจากระบบเหล่านั้นเพื่อใช้เป็นข้อมูลในการดำเนินการในขั้นตอนต่อไป

2) รวบรวมความต้องการในด้านต่างๆ และนำมาวิเคราะห์เพื่อสรุปเป็นข้อกำหนดที่ชัดเจน นักวิเคราะห์ระบบได้ทำการสังเกตการทำงานของเจ้าหน้าที่เพื่อศึกษากระบวนการทำงาน พร้อมทั้งหาแนวทางในการปรับปรุงการทำงานให้มีความกระชับและสะดวกยิ่งขึ้น แล้วจึงทำการสรุปเป็นความต้องการของโปรแกรมการคืนเงินทดรองจ่ายพนักงานเพื่อใช้เป็นกรอบในการพัฒนาโปรแกรมต่อไป

3) สร้างแบบจำลองกระบวนการของระบบใหม่ ด้วยการวาดแผนภาพกระแสข้อมูล (Data Flos Diagram: DFD) นักวิเคราะห์ระบบได้ทำการสร้างกระบวนการทางธุรกิจ (Business Process) เพื่อเป็นเครื่องมือในการทำความเข้าใจร่วมกันระหว่างนักวิเคราะห์และผู้ที่เกี่ยวข้องกับระบบ พร้อมทั้งได้จัดทำแบบจำลองเชิงวัตถุ (Object Model) เพื่อใช้เป็นเครื่องมือหลักในการพัฒนาโปรแกรมในกระบวนการต่อไป

4) สร้างแบบจำลองข้อมูล ด้วยการวาดอีอาร์ ไดอะแกรม (Entity Relationship Diagram: ERD) นักวิเคราะห์ระบบได้จัดทำแบบจำลองข้อมูลเพื่อเป็นเอกสารอ้างอิงสำหรับการออกแบบโครงสร้างของฐานข้อมูล (Database) ต่อไป

จากการดำเนินการในระยะที่ 2 การวิเคราะห์ระบบ (Analysis Phase) เมื่อนำไปเปรียบเทียบกับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Requirement Management (REQM) และ Requirement Development (RD) พบว่ามีการดำเนินการตามกระบวนการที่มาตรฐาน CMMI กำหนดไว้ และไม่ได้ดำเนินการตามที่มาตรฐาน CMMI กำหนดไว้ ดังตารางที่ 18 และตารางที่ 19

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

ตารางที่ 18 แสดงผลการเปรียบเทียบการดำเนินงานระยะที่ 2 การวิเคราะห์ระบบ (Analysis Phase) ของวงจรพัฒนาระบบงาน (SDLC) กับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Requirements Management (REQM)

กระบวนการที่ระบุไว้ใน CMMI และตรงกับการดำเนินการของ SDLC	**กระบวนการที่ระบุไว้ใน CMMI แต่ไม่ได้ระบุชัดเจนใน SDLC ให้ดำเนินการ**
1. ทีมงานพัฒนาโปรแกรมได้ทำความเข้าใจความต้องการของโปรแกรม 2. ผู้ที่เกี่ยวข้องกับโครงการได้เห็นชอบร่วมกันในการกำหนดความต้องการของโปรแกรม	1. การจัดการความต้องการเมื่อเกิดการเปลี่ยนแปลงความต้องการของโปรแกรม 2. การนำความต้องการที่ได้มาใช้เป็นเครื่องมือในการตรวจสอบและติดตามการดำเนินงานในกระบวนการต่างๆของโครงการ 3. การระบุถึงความไม่สอดคล้องกันระหว่างความต้องการและโปรแกรมที่จะพัฒนา

ตารางที่ 19 แสดงผลการเปรียบเทียบการดำเนินงานระยะที่ 2 การวิเคราะห์ระบบ (Analysis Phase) ของวงจรพัฒนาระบบงาน (SDLC) กับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Requirements Development (RD)

กระบวนการที่ระบุไว้ใน CMMI และตรงกับการดำเนินการของ SDLC	**กระบวนการที่ระบุไว้ใน CMMI แต่ไม่ได้ระบุชัดเจนใน SDLC ให้ดำเนินการ**
1. การรวบรวมรายละเอียดความต้องการของโปรแกรมเพื่อการกำหนดรายละเอียดในส่วนต่างๆของโปรแกรม 2. สรุปความต้องการ เพื่อใช้เป็นแนวทางในการพัฒนาโปรแกรม 3. จัดทำความต้องการของโปรแกรมและส่วนประกอบต่างๆของโปรแกรม 4. จัดทำข้อกำหนดในการเชื่อมโยงส่วนประกอบต่างๆของโปรแกรมเข้าด้วยกัน 5. การวิเคราะห์ความต้องการของโปรแกรม	1. จับคู่ความต้องการของโปรแกรมกับส่วนประกอบต่างๆของโปรแกรม 2. สร้างสถานการณ์จำลองในการใช้งานโปรแกรม 3. สร้างกระบวนการทำงานของโปรแกรม 4. การตรวจสอบความต้องการของโปรแกรมเพื่อสร้างความมั่นใจว่าโปรแกรมที่จะพัฒนาสามารถตอบสนองความต้องการได้จริง

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ระยะที่ 3 การออกแบบ (Design Phase)

1) ออกแบบสถาปัตยกรรมระบบ (Architecture Design) นักวิเคราะห์ระบบได้ทำการวางแผนด้านการเตรียม Hardware เพื่อเป็นเครื่องแม่ข่าย (Server) สำหรับโปรแกรมพร้อมทั้งเตรียมความพร้อมด้านเครือข่าย (Network) เพื่อรองรับการใช้งานโปรแกรม

2) ออกแบบฐานข้อมูล (Database Design) นักวิเคราะห์ระบบได้ทำการออกแบบฐานข้อมูลโดยใช้แบบจำลองข้อมูลที่ได้จัดทำไว้แล้วเป็นโครงสร้างหลักในการออกแบบ พร้อมทั้งเลือกใช้โปรแกรม Microsoft SQL Server 2000 ในการจัดการฐานข้อมูล

3) ออกแบบเอ้าท์พุต (Output Design) นักวิเคราะห์และนักออกแบบ (Designer) ได้ออกแบบเอ้าท์พุตของโปรแกรมใน 3 รูปแบบได้แก่ เอ้าท์พุตทางจอภาพ (Screen) เอ้าท์พุตทางเครื่องพิมพ์ (Printer) และเอ้าท์พุตในรูปของไฟล์ข้อมูล (Files)

4) ออกแบบอินพุต (Input Design) นักวิเคราะห์ได้ทำการออกแบบการนำเข้าข้อมูลจากระบบการเบิกค่าใช้จ่ายเบ็ดเตล็ด (Petty Cash) ระบบการเบิกค่าใช้ประจำเดือน (Monthly Expense) และระบบใบคำขอ (Request) ในรูปของการประมวลผลแบบออนไลน์ (Online Processing) เพื่อความสะดวกในการทำงานของผู้ใช้ โดยในส่วนของอินพุตอื่นๆ นักวิเคราะห์ได้กำหนดรูปแบบในการอินพุตข้อมูลโดยให้ผู้ใช้กรอกข้อมูลน้อยที่สุดเพื่อป้องกันความผิดพลาดในการกรอกข้อมูลและเพื่อเป็นการอำนวยความสะดวกในการกรอกข้อมูลให้แก่ผู้ใช้

5) ออกแบบยูสเซอร์อินเตอร์เฟซ (User Interface Design) นักวิเคราะห์และนักออกแบบได้ทำการออกแบบยูสเซอร์อินเตอร์เฟซ (User Interface Design) โดยการนำรูปแบบเอ้าท์พุตทางจอภาพ (Screen) มาเป็นกรอบในการออกแบบ

6) จัดทำต้นแบบ (Prototype) นักออกแบบได้นำส่วนต่างๆที่ได้ทำการออกแบบไว้ได้แก่ การออกแบบเอ้าท์พุต (Output Design) การออกแบบอินพุต (Input Design) และการออกแบบยูสเซอร์อินเตอร์เฟซ (User Interface Design) มารวบรวมให้เป็นต้นแบบของโปรแกรม

7) ออกแบบโปรแกรม (Program Design) นักวิเคราะห์ระบบได้จัดทำข้อกำหนดของโปรแกรม เพื่อใช้เป็นเอกสารในการส่งมอบงานให้กับโปรแกรมเมอร์ในการเขียนโปรแกรมต่อไป

จากการดำเนินการในระยะที่ 3 การออกแบบ (Design Phase) เมื่อนำไปเปรียบเทียบกับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Technical Solution (TS) และ Product Integration (PI) พบว่ามีการดำเนินการตามกระบวนการที่มาตรฐาน CMMI กำหนดไว้ และไม่ได้ดำเนินการตามที่มาตรฐาน CMMI กำหนดไว้ดังตารางที่ 20 และตารางที่ 21

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ตารางที่ 20 แสดงผลการเปรียบเทียบการดำเนินงานระยะที่ 3 การออกแบบ (Design Phase) ของวงจรพัฒนาระบบงาน (SDLC) กับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Technical Solution (TS)

กระบวนการที่ระบุไว้ใน CMMI และตรงกับการดำเนินการของ SDLC	**กระบวนการที่ระบุไว้ใน CMMI แต่ไม่ได้ระบุชัดเจนใน SDLC ให้ดำเนินการ**
1. การเลือกวิธีการในการแก้ปัญหา 2. ออกแบบโปรแกรมและส่วนประกอบต่างๆในโปรแกรม 3. ออกแบบการเชื่อมโยงส่วนประกอบต่างๆในโปรแกรม	1. การสร้างทางเลือกในการแก้ปัญหา 2. ออกแบบข้อกำหนดของส่วนประกอบต่างๆในโปรแกรม 3. การสร้างเอกสารเพื่อการบำรุงรักษาระบบ 4. พิจารณาว่าจะพัฒนาโปรแกรมเองหรือซื้อโปรแกรมมาใช้

ตารางที่ 21 แสดงผลการเปรียบเทียบการดำเนินงานระยะที่ 3 การออกแบบ (Design Phase) ของวงจรพัฒนาระบบงาน (SDLC) กับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Product Integration (PI)

กระบวนการที่ระบุไว้ใน CMMI และตรงกับการดำเนินการของ SDLC	**กระบวนการที่ระบุไว้ใน CMMI แต่ไม่ได้ระบุชัดเจนใน SDLC ให้ดำเนินการ**
1. กำหนดลำดับในการเชื่อมต่อส่วนประกอบต่างๆของโปรแกรม 2. การสร้างการเชื่อมโยงส่วนประกอบต่างๆในโปรแกรม 3. การรวบรวมส่วนประกอบต่างๆของโปรแกรมเข้าด้วยกัน	1. การสร้างข้อกำหนดของส่วนประกอบต่างๆในโปรแกรม 2. ตรวจสอบรายละเอียดในการเชื่อมโยงส่วนประกอบต่างๆของโปรแกรม 3. การจัดการการเชื่อมโยงส่วนประกอบต่างๆตั้งแต่การออกแบบ การสร้าง ตลอดจนการเชื่อมโยงส่วนประกอบเหล่านั้นเข้าด้วยกัน 4. การยืนยันว่าส่วนประกอบต่างๆภายในระบบสามารถเชื่อมโยงกันได้ 5. การประเมินความสามารถในการเชื่อมโยงส่วนประกอบต่างๆของโปรแกรม 6. การรวบรวมโปรแกรมให้พร้อมสำหรับการจัดส่งโปรแกรมไปยังผู้ใช้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ระยะที่ 4 การนำไปใช้ (Implementation Phase)

1) การเขียนโปรแกรม (Coding) โครงการพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงานมีโปรแกรมเมอร์ทำหน้าที่เขียนโปรแกรมจำนวน 1 คน โดยโปรแกรมเมอร์จะเขียนโปรแกรมตามเอกสารข้อกำหนดของโปรแกรมที่ได้รับจากนักวิเคราะห์ระบบ พร้อมทั้งทำการทดสอบโปรแกรมเบื้องต้นก่อนส่งมอบโปรแกรมให้นักวิเคราะห์ระบบไปทำการทดสอบอีกครั้ง ทั้งนี้โปรแกรมเมอร์ใช้เวลา 20 วันทำการ ในการเขียนโปรแกรมต่างๆของโครงการนี้

2) การทดสอบ (Testing) นักวิเคราะห์ระบบเป็นผู้ดำเนินการทดสอบโปรแกรมหลังจากได้รับมอบโปรแกรมที่พัฒนาแล้วเสร็จมาจากโปรแกรมเมอร์ โดยหลังจากโปรแกรมผ่านการทดสอบแล้ว ผู้จัดการโครงการได้ทำการสาธิตการใช้งานโปรแกรม (Demonstartion) ให้แก่ผู้ที่เกี่ยวข้อง พร้อมรับฟังข้อเสนอเพื่อการปรับปรุงโปรแกรมให้สมบูรณ์ยิ่งขึ้นต่อไป

3) ติดตั้งระบบ (System Installation) นักวิเคราะห์ระบบนำโปรแกรมที่ได้รับการปรับปรุงหลังจากการสาธิตโปรแกรม ติดตั้งบนเครื่องแม่ข่ายที่เตรียมไว้ พร้อมทั้งเปิดให้ผู้ใช้เข้าทดสอบการทำงานของโปรแกรมก่อนการเริ่มใช้งานจริง

4) จัดทำเอกสารคู่มือการใช้งาน (Documentation/Manuals) นักวิเคราะห์ระบบไม่ได้จัดทำเอกสารคู่มือการใช้งานโปรแกรม เนื่องจากเห็นว่าโปรแกรมสามารถใช้งานง่ายและมีผู้ใช้งานโปรแกรมเพียง 3 คน

5) การประเมินผลระบบ (System Evaluation) โครงการพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงานไม่ได้มีการประเมินผลอย่างจริงจัง โดยผู้จัดการโครงการเพียงแค่ใช้การสอบถามผู้ใช้ถึงปัญหาและอุปสรรคที่พบในการใช้งานโปรแกรม โดยเมื่อพบปัญหาและอุปสรรคในการใช้งาน ผู้จัดการโครงการจะแจ้งไปยังนักวิเคราะห์ระบบเพื่อรับทราบและหาทางแก้ไขปัญหาที่พบต่อไป

จากการดำเนินการในระยะที่ 4 การนำไปใช้ (Implementation Phase) เมื่อนำไปเปรียบเทียบกับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Verification (VER) และ Product Integration (PI) พบว่ามีการดำเนินการตามกระบวนการที่มาตรฐาน CMMI กำหนดไว้และไม่ได้ดำเนินการตามที่มาตรฐาน CMMI กำหนดไว้ดังตารางที่ 22 และตารางที่ 23

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

ตารางที่ 22 แสดงผลการเปรียบเทียบการดำเนินงานระยะที่ 4 การนำไปใช้ (Implementation Phase) ของวงจรพัฒนาระบบงาน (SDLC) กับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Verification (VER)

กระบวนการที่ระบุไว้ใน CMMI และตรงกับการดำเนินการของ SDLC	**กระบวนการที่ระบุไว้ใน CMMI แต่ไม่ได้ระบุชัดเจนใน SDLC ให้ดำเนินการ**
1. การเลือกโปรแกรมเพื่อทำการทดสอบการทำงาน 2. การสร้างสภาพแวดล้อมเพื่อการทดสอบการทำงานของโปรแกรม 3. การทดสอบการทำงานของโปรแกรม 4. การวิเคราะห์ผลที่ได้จากการทดสอบการทำงานของโปรแกรม	1. การสร้างขั้นตอนและเงื่อนไขในการทดสอบการทำงานของโปรแกรม 2. การเตรียมความพร้อมสำหรับการตรวจสอบโปรแกรม 3. การรวบรวมข้อมูลต่างๆที่ได้จากการตรวจสอบโปรแกรม 4. การวิเคราะห์ข้อมูลต่างๆที่ได้จากการตรวจสอบโปรแกรม

ตารางที่ 23 แสดงผลการเปรียบเทียบการดำเนินงานระยะที่ 4 การนำไปใช้ (Implementation Phase) ของวงจรพัฒนาระบบงาน (SDLC) กับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area Validation (VAL)

กระบวนการที่ระบุไว้ใน CMMI และตรงกับการดำเนินการของ SDLC	**กระบวนการที่ระบุไว้ใน CMMI แต่ไม่ได้ระบุชัดเจนใน SDLC ให้ดำเนินการ**
1. การเลือกโปรแกรมเพื่อทำการทดสอบความถูกต้องในการทำงาน 2. การสร้างสภาพแวดล้อมเพื่อการทดสอบความถูกต้องในการทำงานของโปรแกรม 3. การทดสอบความถูกต้องในการทำงานของโปรแกรม 4. การวิเคราะห์ผลที่ได้จากการทดสอบความถูกต้องในการทำงานของโปรแกรม	1. การสร้างขั้นตอนและเงื่อนไขในการทดสอบความถูกต้องในการทำงานของโปรแกรม

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ระยะที่ 5 การบำรุงรักษา (Maintenance Phase)

1) การบำรุงรักษาระบบ (System Maintenance) หลังจากเริ่มการใช้งานโปรแกรมนักวิเคราะห์ระบบได้เข้าไปร่วมสังเกตการทำงานของผู้ใช้เพื่อรับทราบปัญหาที่เกิดขึ้นในการใช้งานโปรแกรม พร้อมทั้งมอบหมายให้โปรแกรมเมอร์เตรียมพร้อมเพื่อแก้ปัญหาเหล่านั้นในทันที

2) การบำรุงรักษาด้วยการปรับปรุงให้ระบบมีประสิทธิภาพดียิ่งขึ้น (Perfective Maintenance) นักวิเคราะห์ระบบร่วมกับโปรแกรมเมอร์ได้มีการปรับปรุงการเขียนโปรแกรมในบางส่วนเพื่อให้โปรแกรมสามารถทำงานได้อย่างรวดเร็วมากยิ่งขึ้น พร้อมกันนี้ได้มีการเพิ่มหน้าจอสำหรับเลขาแผนกเพื่อใช้ในการตรวจสอบรายการเบิกเงินทดรองจ่ายคืนพนักงานอีกด้วย

ส่วนที่ 2 ข้อมูลที่ได้จากการสัมภาษณ์ผู้ใช้โปรแกรม

จากการสัมภาษณ์ผู้ใช้งานโปรแกรมจากประชากรทั้งหมดจำนวน 3 คน พบว่าผู้ใช้มีความพึงพอใจต่อโปรแกรมมาก เนื่องจากโปรแกรมสามารถดำเนินการได้ตามวัตถุประสงค์ที่ได้ตั้งไว้ สามารถแก้ไขปัญหาที่พบในการปฏิบัติงานได้เป็นอย่างดี สามารถลดค่าใช้จ่ายที่เกิดจากการคืนเงินให้พนักงานได้ นอกจากนี้ยังสามารถสร้างรายงานเพื่อใช้เป็นเอกสารประกอบการดำเนินงานในด้านต่างๆได้อย่างครบถ้วน การใช้งานโปรแกรมก็ไม่ยุ่งยาก ซับซ้อน และสามารถประมวลผลได้อย่างรวดเร็ว

แม้ในช่วงแรกของการใช้งานโปรแกรม ผู้ใช้จะประสบปัญหาในการใช้งานอยู่บ้าง โดยเฉพาะเรื่องความต่อเนื่องในการใช้งานระบบ แต่ทีมงานพัฒนาก็สามารถดำเนินการแก้ไขปัญหาเหล่านั้นได้อย่างทันท่วงที ทำให้ผู้ใช้โปรแกรมสามารถปฏิบัติงานต่อได้จนเสร็จสิ้นกระบวนการ นอกจากนี้ทีมงานยังมีการเพิ่มเติมความสามารถใหม่ๆเข้าไปในโปรแกรม เพื่อให้โปรแกรมสามารถดำเนินงานได้อย่างรวดเร็ว ถูกต้องมากยิ่งขึ้น และสามารถใช้เป็นเครื่องมือในการติดตามความคืบหน้าเกี่ยวกับการเบิกเงินคืนของพนักงานได้เป็นอย่างดี

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ส่วนที่ 3 สรุปผลการศึกษาเปรียบเทียบ SDLC กับมาตรฐาน CMMI

จากการศึกษาสามารถสรุปภารกิจเฉพาะ (Specic Practice) ของแต่ละ Process Area ที่ทีมงานได้ดำเนินการและไม่ได้ดำเนินการ เพื่อประเมินระดับ Capability Level ของทีมงานดังตารางที่ 24

ตารางที่ 24 แสดงภารกิจเฉพาะ (Specic Practice) ของแต่ละ Process Area ที่ทีมงานได้ดำเนินการและไม่ได้ดำเนินการ

กระบวนการที่ระบุไว้ใน CMMI	ดำเนินการ	ไม่ได้ดำเนินการ
Process Area Requirements Management (REQM)		
ทีมงานพัฒนาโปรแกรมได้ทำความเข้าใจความต้องการของโปรแกรม	✓	
ผู้ที่เกี่ยวข้องกับโครงการได้เห็นชอบร่วมกันในการกำหนดความต้องการของโปรแกรม	✓	
การจัดการความต้องการเมื่อเกิดการเปลี่ยนแปลงความต้องการของโปรแกรม		✓
การนำความต้องการที่ได้มาใช้เป็นเครื่องมือในการตรวจสอบและติดตามการดำเนินงานในกระบวนการต่างๆของโครงการ		✓
การระบุถึงความไม่สอดคล้องกันระหว่างความต้องการและโปรแกรมที่จะพัฒนา		✓
Process Area Requirements Development		
การรวบรวมรายละเอียดความต้องการของโปรแกรมเพื่อการกำหนดรายละเอียดในส่วนต่างๆของโปรแกรม	✓	
สรุปความต้องการ เพื่อใช้เป็นแนวทางในการพัฒนาโปรแกรม	✓	
จัดทำความต้องการของโปรแกรมและส่วนประกอบต่างๆของโปรแกรม	✓	
จัดทำข้อกำหนดในการเชื่อมโยงส่วนประกอบต่างๆของโปรแกรมเข้าด้วยกัน	✓	

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การวิเคราะห์ความต้องการของโปรแกรม	✓	
จับคู่ความต้องการของโปรแกรมกับส่วนประกอบต่างๆ ของโปรแกรม		✓
สร้างสถานการณ์จำลองในการใช้งานโปรแก		✓
สร้างกระบวนการทำงานของโปรแกรม		✓
การตรวจสอบความต้องการของโปรแกรมเพื่อสร้างความมั่นใจว่าโปรแกรมที่จะพัฒนาสามารถตอบสนองความต้องการได้จริง		✓
Process Area Technical Solution (TS)		
การเลือกวิธีการในการแก้ปัญหา	✓	
ออกแบบโปรแกรมและส่วนประกอบต่างๆ ในโปรแกรม	✓	
ออกแบบการเชื่อมโยงส่วนประกอบต่างๆในโปรแกรม	✓	
การสร้างทางเลือกในการแก้ปัญหา		✓
ออกแบบข้อกำหนดของส่วนประกอบต่างๆ ในโปรแกรม		✓
การสร้างเอกสารเพื่อการบำรุงรักษาระบบ		✓
พิจารณาว่าจะพัฒนาโปรแกรมเองหรือซื้อโปรแกรมมาใช้		✓
Process Area Product Integration (PI)		
กำหนดลำดับในการเชื่อมต่อส่วนประกอบต่างๆของโปรแกรม	✓	
การสร้างการเชื่อมโยงส่วนประกอบต่างๆในโปรแกรม	✓	
การรวบรวมส่วนประกอบต่างๆของโปรแกรมเข้าด้วยกัน	✓	
การสร้างข้อกำหนดของส่วนประกอบต่างๆ ในโปรแกรม		✓
ตรวจสอบรายละเอียดในการเชื่อมโยงส่วนประกอบต่างๆของโปรแกรม		✓

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การจัดการการเชื่อมโยงส่วนประกอบต่างๆ ตั้งแต่การออกแบบ การสร้าง ตลอดจนการเชื่อมโยงส่วนประกอบเหล่านั้นเข้าด้วยกัน		✓
การยืนยันว่าส่วนประกอบต่างๆภายในระบบสามารถเชื่อมโยงกันได้		✓
การประเมินความสามารถในการเชื่อมโยงส่วนประกอบต่างๆของโปรแกรม		✓
การรวบรวมโปรแกรมให้พร้อมสำหรับการจัดส่งโปรแกรมไปยังผู้ใช้		✓
Process Area Verification (VER)		
การเลือกโปรแกรมเพื่อทำการทดสอบการทำงาน	✓	
การสร้างสภาพแวดล้อมเพื่อการทดสอบการทำงานของโปรแกรม	✓	
การทดสอบการทำงานของโปรแกรม	✓	
การวิเคราะห์ผลที่ได้จากการทดสอบการทำงานของโปรแกรม	✓	
การสร้างขั้นตอนและเงื่อนไขในการทดสอบการทำงานของโปรแกรม		✓
การเตรียมความพร้อมสำหรับการตรวจสอบโปรแกรม		✓
การรวบรวมข้อมูลต่างๆที่ได้จากการตรวจสอบโปรแกรม		✓
การวิเคราะห์ข้อมูลต่างๆที่ได้จากการตรวจสอบโปรแกรม		✓
Process Area Validation (VAL)		
การเลือกโปรแกรมเพื่อทำการทดสอบความถูกต้องในการทำงาน	✓	
การสร้างสภาพแวดล้อมเพื่อการทดสอบความถูกต้องในการทำงานของโปรแกรม	✓	
การทดสอบความถูกต้องในการทำงานของโปรแกรม	✓	

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การวิเคราะห์ผลที่ได้จากการทดสอบความถูกต้องในการทำงานของโปรแกรม	✓	
การสร้างขั้นตอนและเงื่อนไขในการทดสอบความถูกต้องในการทำงานของโปรแกรม		✓

อภิปรายผลการศึกษา

เมื่อนำผลการศึกษามาทำการตรวจสอบกระบวนการต่างๆ ที่ทีมงานพัฒนาโปรแกรมได้ดำเนินการไป แล้วนำมาเปรียบเทียบกับมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ใน Process Area ที่สอดคล้องกันจะพบความแตกต่างระหว่างระดับ Capability Level ที่คาดหวัง กับระดับ Capability Level ของทีมงานที่เป็นอยู่ ดังตารางที่ 25

ตารางที่ 25 แสดงผลการเปรียบเทียบระดับ Capability Level ที่คาดหวัง กับระดับ Capability Level ที่เป็นอยู่ จำแนกตาม Process Area

Process Area	**Capability Level ที่คาดหวัง**	**Capability Level ที่เป็นอยู่**
Requirement Management (REQM)	Level 2	Level 0
Product Integration (PI) Requirement Development (RD) Technical Solution (TS) Validation (VAL) Verification (VER)	Level 3	

จากตารางที่ 25 พบว่าระดับ Capability Level ที่เป็นอยู่ของทีมงานพัฒนาโปรแกรมใน Process Area Requirement Management (REQM) Product Integration (PI) Requirement Development (RD) Technical Solution (TS) Validation (VAL) และ Verification (VER) อยู่ใน Level 0 ซึ่งหมายถึง ทีมงานยังไม่มีรูปแบบในการพัฒนาโปรแกรมที่ชัดเจน และไม่สามารถดำเนินการตามกระบวนการพื้นฐานของแต่ละ Process Area ได้

จากผลการศึกษาพบว่า การพัฒนาโปรแกรมตามทฤษฎีวงจรการพัฒนาระบบงาน (SDLC) หากทีมงานพัฒนาดำเนินการกระบวนการพัฒนาโปรแกรมในระยะต่างๆ ก็สามารถบรรลุวัตถุประสงค์ในการสร้างโปรแกรมได้ตามต้องการ แต่ทั้งนี้การพัฒนาโปรแกรมตามวงจรการ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

พัฒนาระบบงาน (SDLC) นั้นไม่มีข้อกำหนดที่ชัดเจนในการดำเนินการระยะต่างๆ ซึ่งหมายถึงทีมงานพัฒนาโปรแกรมแต่ละทีม แม้ว่าจะดำเนินการตามแนวทางของวงจรการพัฒนาระบบงาน (SDLC) ก็อาจได้ผลลัพธ์ที่แตกต่างกัน เนื่องจากในรายละเอียดของการดำเนินการนั้นแตกต่างกันไปตามความคิดของผู้ที่เกี่ยวข้องในทีมงานนั้นๆ

การพัฒนาซอฟท์แวร์โดยอาศัยวิธีการพัฒนาระบบร่วมกัน (Joint Application Development: JAD) ยังเป็นการช่วยให้การสื่อสารระหว่างทีมงานพัฒนาระบบกับผู้ใช้และผู้ที่เกี่ยวข้องเป็นไปอย่างมีประสิทธิภาพ และยังช่วยให้สามารถสร้างโปรแกรมที่ตอบสนองความต้องการของผู้ใช้ได้เป็นอย่างดี

การพัฒนาซอฟท์แวร์โดยอาศัยวิธีการพัฒนาเชิงวัตถุ (Object Oriented) เป็นการพัฒนาที่ช่วยให้การจัดรูปแบบของโปรแกรมมีความเป็นระเบียบ ทำให้ง่ายต่อการบำรุงรักษาโปรแกรมในอนาคต นอกจากนี้การพัฒนาเชิงวัตถุ (Object Oriented) นี้ยังช่วยให้เกิดการใช้โปรแกรมซ้ำ (Reuse) ซึ่งช่วยลดเวลาในการพัฒนาโปรแกรมที่มีลักษณะการทำงานที่เหมือนกัน ซึ่งช่วยให้โปรแกรมเมอร์ไม่ต้องเขียนโปรแกรมซ้ำๆอีกต่อไป

ในขณะที่มาตรฐานการพัฒนาซอฟท์แวร์ CMMI เป็นมาตรฐานที่ช่วยทำให้ทีมงานมีการดำเนินกระบวนการในการพัฒนาโปรแกรมที่ชัดเจน เป็นรูปแบบมากขึ้น ซึ่งช่วยให้การทำหน้าที่ต่างๆของทีมงานในโครงการเป็นระบบมากขึ้น มีมาตรฐานการดำเนินงานที่ชัดเจน โดยเฉพาะหากโครงการพัฒนาโปรแกรมนั้นมีขนาดใหญ่ ประกอบด้วยทีมงานจำนวนมาก การมีกระบวนการที่ต้องดำเนินงานอย่างชัดเจน จะช่วยให้ทีมงานสามารถแบ่งงานไปทำเพื่อลดระยะเวลาในการพัฒนาให้สั้นลงและเมื่อนำโปรแกรมที่ทีมงานแต่ละส่วนพัฒนาขึ้นมารวมกัน โปรแกรมเหล่านั้นจะสามารถทำงานร่วมกันได้เป็นอย่างดี เนื่องจากมาตรฐานการพัฒนาซอฟท์แวร์ CMMI มีการกำหนดให้ต้องกำหนดรูปแบบในการเชื่อมโยงการทำงานของส่วนประกอบต่างๆไว้แล้ว

นอกจากนี้มาตรฐานการพัฒนาซอฟท์แวร์ CMMI ยังเป็นเครื่องยืนยันการทำงานของทีมงานว่ามีรูปแบบการทำงานที่ชัดเจน เชื่อถือได้ โดยโปรแกรมที่พัฒนาขึ้นจะสามารถตอบสนองความต้องการของผู้ที่เกี่ยวข้องเป็นอย่างดี

จากการศึกษาพบว่าการพัฒนาโปรแกรมโดยใช้เว็บเทคโนโลยี (Web Technology) สามารถสร้างระบบงานที่สามารถปฏิบัติงานได้อย่างถูกต้อง และยังสามารถสร้างความพึงพอใจให้กับผู้ใช้งานได้เป็นอย่างดี ซึ่งสอดคล้องกับผลการศึกษาของ ทรงยศ แก้ววิจิตร (2547) ที่พบว่าระบบที่พัฒนาขึ้นโดยใช้เว็บเทคโนโลยี (Web Technology) เป็นระบบที่สามารถใช้งานได้ดี ทำงานได้อย่างมีประสิทธิภาพและสามารถปฏิบัติงานได้อย่างถูกต้อง และยังพบอีกว่าทีมงานพัฒนาโปรแกรมจะดำเนินกระบวนการในการพัฒนาโปรแกรม โดยอาศัยประสบการณ์และความคิดเห็น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ของทีมงานในการพัฒนาโปรแกรมเป็นหลัก ซึ่งสอดคล้องกับผลการศึกษาของ ศิริลักษณ์ โรจน์วงษ์วิริยะ (2549) ที่พบว่า บุคลากรมักมองเพียงความสะดวกและความรวดเร็ว เพื่อให้ทันต่อความต้องการในการนำข้อมูลไปใช้งานเป็นหลัก จนละเลยมุมมองสำคัญด้านอื่นๆไป

จากการศึกษาครั้งนี้ทำให้ทราบว่า การพัฒนาโปรแกรมโดยที่ทีมงานมีมาตรฐานในการพัฒนาที่ชัดเจน จะช่วยเพิ่มมาตรฐานของซอฟท์แวร์ไทยให้สามารถก้าวขึ้นสู่การพัฒนาซอฟท์แวร์ในระดับสากลได้ แม้ว่ามาตรฐานที่นำมาใช้จะทำให้ทีมงานต้องเพิ่มรายละเอียดในการพัฒนาโปรแกรมแต่ละขั้นตอนให้ชัดเจนมากยิ่งขึ้นก็ตาม นอกจากนี้อาจกล่าวได้ว่า มาตรฐานการพัฒนาซอฟท์แวร์ CMMI มีกระบวนการที่ยุ่งยาก ซับซ้อน และอาจเหมาะสมกับการพัฒนาซอฟท์แวร์ขนาดใหญ่มากกว่าการพัฒนาซอฟท์แวร์ขนาดเล็ก

ข้อค้นพบ

จากการศึกษา กระบวนการพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงาน ของบริษัท คอนโทรลดาต้า (ประเทศไทย) จำกัด มีข้อค้นพบที่น่าสนใจดังนี้

1. การที่ทีมงานพัฒนาโปรแกรมเชิญผู้มีส่วนเกี่ยวข้องกับโปรแกรมที่จะพัฒนาให้เข้ามามีส่วนร่วมในการกำหนดปัญหา และร่วมให้ความคิดเห็นกับแนวทางแก้ไขปัญหาตั้งแต่เริ่มต้นโครงการนั้น จะส่งผลให้การกำหนดปัญหาเป็นไปอย่างตรงประเด็น รวมไปถึงแนวทางการแก้ไขปัญหาอยู่บนพื้นฐานที่เป็นไปได้ และผู้เกี่ยวข้องพร้อมที่จะปรับเปลี่ยนการดำเนินงานเพื่อให้สอดคล้องกับวิธีการแก้ไขปัญหาเป็นไปด้วยความเต็มใจ

2. การที่ทีมงานพัฒนาโปรแกรมดำเนินการสร้างต้นแบบ (Prototype) เพื่อใช้เป็นเครื่องมือในการทำความเข้าใจระหว่างทีมงานพัฒนากับผู้ที่เกี่ยวข้อง จะช่วยให้โปรแกรมที่พัฒนาขึ้นได้รับการยอมรับจากผู้ที่เกี่ยวข้องเป็นอย่างดี เนื่องจากผู้ที่เกี่ยวข้องสามารถเห็นภาพจำลองการปฏิบัติงาน อีกทั้งยังสามารถให้ข้อเสนอแนะเพื่อปรับปรุงการทำงานของโปรแกรมให้ดีและตรงตามความต้องการของผู้ที่เกี่ยวข้องยิ่งขึ้น

3. ผู้ใช้งานโปรแกรมจะมีความพึงพอใจในตัวโปรแกรม เมื่อโปรแกรมดังกล่าวมีความสะดวกในการใช้งาน มีขั้นตอนในการใช้งานน้อย และสามารถถ่ายทอดการใช้งานให้กับผู้ใช้โปรแกรมท่านอื่นได้อย่างง่าย

4. การพัฒนาซอฟท์แวร์ตามวงจรการพัฒนาระบบงาน (SDLC) อาจเป็นการดำเนินการที่ไม่มีมาตรฐานที่ชัดเจนเพียงพอ เนื่องจาก SDLC ไม่มีข้อกำหนดในการดำเนินงานที่ชัดเจน โดยในแต่ละระยะนั้นบอกเพียงกรอบในการปฏิบัติเท่านั้น ไม่ได้เขียนเป็นข้อบังคับให้ต้องปฏิบัติ ในขณะที่การพัฒนาซอฟท์แวร์ตามมาตราฐาน CMMI อาจไม่เหมาะกับการพัฒนาซอฟท์แวร์ขนาดเล็ก

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

เนื่องจากการพัฒนาซอฟท์แวร์ขนาดเล็กไม่มีความซับซ้อนมากนัก การดำเนินการตามที่มาตรฐาน CMMI กำหนดทั้งหมดจึงอาจเป็นการสร้างความยุ่งยากให้กับทีมงานพัฒนาได้ แต่ถ้านำมาตรฐาน CMMI เพียงบางส่วนหรือบาง Process Area มาประยุกต์ใช้น่าจะช่วยให้การพัฒนาซอฟท์แวร์มีรูปแบบที่ชัดเจนขึ้นได้ ตัวอย่างเช่น การไม่ได้ดำเนินการตามมาตรฐาน CMMI ใน Process Area Requirements Management (REQM) ในเรื่องการนำความต้องการมาเป็นเครื่องมือในการติดตามความคืบหน้าในการพัฒนาโปรแกรม ซึ่งทำให้ไม่ทราบว่าความต้องการข้อใดบ้างที่ได้รับการพัฒนาแล้ว และความต้องการข้อใดบ้างที่อยู่ระหว่างการพัฒนา เป็นต้น

5. ถึงแม้ว่ากระบวนการพัฒนาโปรแกรมที่ได้ทำการศึกษาในครั้งนี้จะยังไม่มีมาตรฐานที่ชัดเจน แต่ก็สามารถพัฒนาโปรแกรมที่ตอบสนองต่อความต้องการของผู้ใช้โปรแกรมได้เป็นอย่างดี และผู้ใช้ก็มีความพึงพอใจต่อโปรแกรมที่พัฒนาขึ้น อาจเป็นเพราะการให้ผู้ใช้เข้ามามีส่วนร่วมตั้งแต่แรกและต่อเนื่องจนจบโครงการ

ข้อเสนอแนะ

จากการศึกษา กระบวนการพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงาน ของบริษัท คอนโทรลดาต้า (ประเทศไทย) จำกัด ผู้ศึกษามีข้อเสนอแนะจากการศึกษาดังนี้

1. จากการศึกษาพบว่า ในการพัฒนาโปรแกรมควรนำหลายๆวิธีการมาประยุกต์ใช้ ซึ่งขึ้นอยู่กับลักษณะของงานและขนาดของโครงการที่จะพัฒนา โดยในแต่ละงานอาจเลือกใช้วิธีการที่เหมาะสมในการพัฒนาเช่น หากเป็นโครงการขนาดเล็กและไม่มีความซับซ้อน อาจพัฒนาตามแนวทาง SDLC โดยนำเทคนิค JAD และ Prototype มาใช้ แต่หากเป็นโครงการขนาดกลางขึ้นไปควรพัฒนาตามมาตรฐาน CMMI ซึ่งจะช่วยให้สามารถติดตาม ควบคุม การพัฒนาให้เป็นไปในแนวทางที่ชัดเจนและเหมาะสมกับขนาดของโครงการ

2. จากการศึกษาพบว่า การให้ผู้มีส่วนเกี่ยวข้องกับการพัฒนาโปรแกรมเข้ามามีส่วนร่วมกับโครงการตั้งแต่ ระยะที่ 1 : การวางแผนโครงการ (Project Planning Phase) ซึ่งเป็นระยะเริ่มต้นโครงการ เพื่อร่วมกำหนดปัญหาตลอดจนร่วมพิจารณาแนวทางแก้ไขที่เหมาะสม สามารถสร้างความร่วมมือและสร้างความยอมรับในโปรแกรมที่จะพัฒนาได้เป็นอย่างดี จึงควรให้ผู้มีส่วนเกี่ยวข้องได้เข้าร่วมกับโครงการนั้นๆตั้งแต่เริ่มต้นโครงการ

3. จากการศึกษาพบว่า ทีมงานพัฒนาโปรแกรมการคืนเงินทดรองจ่ายพนักงานยังไม่มีมาตรฐานในการดำเนินงานอย่างชัดเจน จึงควรนำมาตรฐานการพัฒนาซอฟท์แวร์ CMMI ในบางกระบวนการมาประยุกต์ใช้เป็นมาตรฐานสำหรับองค์กรต่อไป ทั้งนี้เนื่องจากมาตรฐานการพัฒนาซอฟท์แวร์ CMMI มีหลาย Process Area จึงควรเริ่มจาก Requirement Management (REQM) ใน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

กระบวนการนำความต้องการที่ได้มาใช้เป็นเครื่องมือในการตรวจสอบ และติดตามการดำเนินงานในกระบวนการต่างๆของโครงการ เพื่อช่วยให้สามารถชี้แจงและตรวจสอบความก้าวหน้าของความต้องการเรื่องต่างๆได้อย่างชัดเจนมากขึ้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved