

บทที่ 1

บทนำ

ความเป็นมาของปัญหา

การแข่งขันทางความคิดในโลกปัจจุบันส่งผลต่อการเปลี่ยนแปลงของนวัตกรรมหรือทฤษฎีต่างๆ เพื่อเป็นแนวทางสำหรับการแก้ไขปัญหาที่เผชิญอยู่รวมถึงความพยายามแก้วิกฤตการณ์จากความผิดพลาดของการบริหารจัดการโครงการพัฒนาซอฟต์แวร์ในโครงการที่ไม่ประสบความสำเร็จ โดยมุ่งหวังให้ผลิตภัณฑ์ซอฟต์แวร์ตอบสนองความต้องการของผู้ใช้ในด้านคุณภาพการทำงานของระบบ ภายใต้งบประมาณและระยะเวลาที่ผู้ที่เกี่ยวข้องกับโครงการพึงพอใจ ซึ่งผลิตภัณฑ์ซอฟต์แวร์มีความสำคัญต่ออุตสาหกรรมต่างๆ เนื่องจากว่าซอฟต์แวร์เป็นตัวขับเคลื่อนให้เกิดกิจกรรมต่างๆ ภายในองค์กรอย่างมีประสิทธิภาพด้วยเหตุที่ว่าข้อมูลที่มีอยู่อย่างมหาศาลจากกิจกรรมการดำเนินงานในทั้งหน่วยงานของภาครัฐบาลและเอกชนนั้นเมื่อนำข้อมูลเหล่านั้นมาวิเคราะห์หรือผ่านการประมวลผลจะก่อให้เกิดสารสนเทศที่เป็นประโยชน์ต่อการกำหนดยุทธศาสตร์ การคาดการณ์และปรับรูปแบบการดำเนินงานให้ทันต่อสภาพการณ์การเปลี่ยนแปลงของสิ่งแวดล้อมและการแข่งขันทางธุรกิจ ดังนั้นองค์กรต่างๆ จึงต้องการระบบช่วยสำหรับการจัดการข้อมูลจำนวนมากเหล่านั้น ซึ่งความต้องการของแต่ละองค์กรมีรูปแบบที่เฉพาะเจาะจงตามโครงสร้างขององค์กร รูปแบบกิจกรรมภายในองค์กร ลักษณะข้อมูลที่ต้องการประมวลผล เป็นต้น ดังนั้นซอฟต์แวร์ที่พัฒนาขึ้นต้องสามารถทำงานได้อย่างเต็มประสิทธิภาพและตอบสนองความต้องการที่แตกต่างของผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์

ระเบียบวิธีการพัฒนาซอฟต์แวร์ซึ่งถือได้ว่าเป็นหลักสำคัญสำหรับการพัฒนาซอฟต์แวร์ หากพิจารณาถึงสาเหตุของซอฟต์แวร์ที่พัฒนาขึ้นมีประสิทธิภาพต่างจากความต้องการของระบบ จะพบว่าเกิดจากการเปลี่ยนแปลงที่เกิดขึ้นอย่างรวดเร็วและคาดเดาได้ยากผลจากการเปลี่ยนแปลงดังกล่าวอาจเกิดขึ้นจากสภาพการณ์การเปลี่ยนแปลงภายนอกและภายในองค์กร เช่น การเปลี่ยนแปลงของตลาดเทคโนโลยี เป็นต้น จากการสภาวะการณ์แข่งขันทางธุรกิจที่ไม่หยุดนิ่ง รวมทั้งเทคโนโลยีต่างๆ ในปัจจุบันช่วยเพิ่มพูนศักยภาพในการเข้าถึงข้อมูลในยุคของสารสนเทศนี้ ส่งผลองค์กรมีการปรับตัวและปรับปรุงกระบวนการการทำงานขององค์กร ซึ่งการเปลี่ยนแปลงเหล่านี้ส่งผลกระทบต่อการพัฒนาซอฟต์แวร์อย่างหลีกเลี่ยงไม่ได้ โดยอาจก่อให้เกิดปัญหาการส่งมอบงานล่าช้าไม่ตรงตามแผนงานที่วางไว้หรือผลิตภัณฑ์ซอฟต์แวร์ที่ส่งมอบไม่ตรงกับความต้องการของผู้ใช้ภายใต้งบประมาณและระยะเวลาที่ยอมรับได้ของผู้ที่เกี่ยวข้อง รวมทั้งปัญหาจากสิ่งแวดล้อมของโครงการ

ที่มีความจำเพาะเจาะจงอย่างมาก ดังนั้นการกำหนดรายละเอียดของกิจกรรมที่มีความเหมาะสมต่อลักษณะเฉพาะเจาะจงของโครงการจึงถือเป็นสิ่งสำคัญต่อโครงการพัฒนาซอฟต์แวร์สำหรับการเลือกใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ที่เปรียบเสมือนหัวใจหลักของการเพิ่มประสิทธิภาพสำหรับโครงการให้ตรงกับความต้องการที่หลากหลายจึงเป็นสิ่งสำคัญอย่างมาก ซึ่งปัจจุบันมีโมเดลสำหรับสนับสนุนการพัฒนาซอฟต์แวร์หลากหลายรูปแบบดังนี้ แบบจำลองที่มีระเบียบแบบแผน (Disciplined Model) แบบจำลองสมัยใหม่ (Modern Approach Series) รวมทั้งแนวคิดการพัฒนาซอฟต์แวร์แบบจำลองอาไจล์ (Agile Model) ซึ่งเริ่มเข้ามามีบทบาทต่อการพัฒนาซอฟต์แวร์โดยเฉพาะในต่างประเทศ โดยที่แต่ละแนวคิดมีความแตกต่างกัน และมีข้อดีและข้อด้อยแตกต่างกัน งานวิจัยชิ้นนี้ได้กำหนดว่าหากเทคนิคการพัฒนาซอฟต์แวร์แบบใดที่ไม่มีแนวทางปฏิบัติหลักตามที่ได้ประกาศไว้ในหลักการของ อาไจล์ (Agile) จะถือว่าหลักการดังกล่าวไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile)

แนวความคิดของการพัฒนาซอฟต์แวร์แบบไม่ใช่หลักการตามแบบ อาไจล์ (Non-Agile) และการพัฒนาซอฟต์แวร์แบบ อาไจล์ (Agile) ล้วนเป็นวิธีการที่ถูกพัฒนาขึ้นโดยมีวัตถุประสงค์หลักเพื่อให้เพิ่มประสิทธิภาพของการพัฒนาซอฟต์แวร์ และลดความเสี่ยงต่างๆ ที่อาจเกิดขึ้นจากสิ่งแวดล้อมของการพัฒนาซอฟต์แวร์ จึงเป็นสิ่งที่น่าศึกษาว่าแนวคิดการพัฒนาซอฟต์แวร์ตามแนวคิดแบบใด ที่ช่วยสนับสนุนให้ซอฟต์แวร์ที่พัฒนาขึ้นเป็นไปตามความต้องการของทีมพัฒนาสำหรับการประเมินประสิทธิภาพขั้นสุดท้ายของโครงการพัฒนาซอฟต์แวร์ตามหลักการบริหารโครงการพัฒนาซอฟต์แวร์แบ่งเป็น 4 มิติ ดังนี้ 1) มิติด้านการบริหารจัดการแผนการดำเนินงานอย่างมีประสิทธิภาพ (Effective Time Management) 2) มิติด้านการบริหารจัดการงบประมาณอย่างมีประสิทธิภาพ (Effective Cost Management) 3) มิติด้านคุณภาพของกระบวนการ (Quality of Process) และ 4) มิติด้านการทำงานร่วมกันของผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ (Inter-supplier performance and Customer satisfactions) ในการเปรียบเทียบเชิงอนุมาน ดังนั้นผู้วิจัยสนใจที่จะเปรียบเทียบประสิทธิภาพของระเบียบวิธีการพัฒนาซอฟต์แวร์ โดยวิธี อาไจล์ (Agile) และการพัฒนาซอฟต์แวร์แบบไม่ใช่หลักการตามแบบอาไจล์ สำหรับโครงการพัฒนาซอฟต์แวร์ในประเทศไทย โดยผู้วิจัยสร้างเครื่องมือช่วยเก็บรวบรวมข้อมูลด้วยแบบสอบถาม ร่วมกับการสัมภาษณ์เชิงลึก พร้อมกับ การสังเกตพฤติกรรมจากกลุ่มตัวอย่าง และนำข้อมูลที่ได้เข้าสู่กระบวนการวิเคราะห์ด้วยวิธีการทางสถิติ และหาข้อสรุปด้วยสถิติเชิงพรรณนาและสถิติทดสอบ Independent Samples T-Test จากนั้นนำผลที่ได้จากการศึกษามาช่วยในกระบวนการวิเคราะห์ เพื่อหาความเหมาะสมกับการพัฒนาซอฟต์แวร์ในรูปแบบที่มีความแตกต่าง และช่วยใน

กระบวนการพัฒนาคุณภาพและกระบวนการควบคุมการผลิตซอฟต์แวร์ โดยมุ่งหวังให้ซอฟต์แวร์ที่ผลิตโดยตลาดซอฟต์แวร์ในไทยให้เป็นที่ยอมรับของตลาดโลก

จุดมุ่งหมายของการศึกษา

1. เพื่อเปรียบเทียบด้านการบริหารจัดการแผนการดำเนินงานอย่างมีประสิทธิภาพ (Effective Time Management) จากการประยุกต์ใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile) และการพัฒนาซอฟต์แวร์แบบไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile)
2. เพื่อเปรียบเทียบประสิทธิภาพด้านการบริหารจัดการงบประมาณอย่างมีประสิทธิภาพ (Effective Cost Management) จากการประยุกต์ใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile) และการพัฒนาซอฟต์แวร์แบบไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile)
3. เพื่อเปรียบเทียบประสิทธิภาพด้านคุณภาพของกระบวนการ (Quality of Process) จากการประยุกต์ใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile) และการพัฒนาซอฟต์แวร์แบบไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile)
4. เพื่อเปรียบเทียบประสิทธิภาพด้านการทำงานร่วมกันของผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ (Inter-supplier performance and Customer satisfactions) จากการประยุกต์ใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile) และการพัฒนาซอฟต์แวร์แบบไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile)

ขอบเขตของงานวิจัย

การวิจัยครั้งนี้มีจุดมุ่งหมายเพื่อเปรียบเทียบประสิทธิภาพการพัฒนาซอฟต์แวร์ตามระเบียบวิธีการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile) และการพัฒนาซอฟต์แวร์แบบไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile) โดยเปรียบเทียบปัจจัยที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ และประสิทธิภาพของการประยุกต์ใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ของกรณีศึกษาพร้อมทั้งศึกษาผลกระทบของกระบวนการพัฒนาซอฟต์แวร์ที่มีการเลือกใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ที่แตกต่างกัน เพื่อนำผลที่ได้จากการศึกษามาช่วยในกระบวนการพัฒนาคุณภาพและกระบวนการควบคุมการผลิตซอฟต์แวร์ โดยแบ่งขอบเขตการวิจัยดังนี้

1. ขอบเขตด้านเนื้อหา

ผู้วิจัยเลือกแนวคิดการประเมินโครงการแบบ CIPP MODEL เนื่องจากแนวคิดดังกล่าวเป็นวิธีการประเมินกระบวนการอย่างต่อเนื่องสามารถครอบคลุมกับการบริหารจัดการโครงการซึ่งแบ่งประเด็นการประเมินผลออกเป็น 4 ประเภดังนี้

1.1 ตัวชี้วัดด้านบริบท (Context): ตัวชี้วัดสามารถพิจารณาได้จากสิ่งต่างๆ ดังนี้

1.1.1 สภาวะแวดล้อมของโครงการ

1.1.2 ความจำเป็น หรือความต้องการขณะนั้น และอนาคต

1.1.3 ความเข้าใจร่วมกันของทุกฝ่ายที่เกี่ยวข้องกับโครงการ

1.2 ตัวชี้วัดด้านปัจจัยนำเข้า (Input): ตัวชี้วัดสามารถพิจารณาได้จากสิ่งต่างๆ ดังนี้

1.2.1 ความชัดเจนของวัตถุประสงค์ของโครงการ

1.2.2 ความพร้อมของทรัพยากร เช่น งบประมาณ ผู้ที่เกี่ยวข้องกับโครงการ

พัฒนาซอฟต์แวร์ เวลา และกฎระเบียบ เป็นต้น

1.3 ตัวชี้วัดด้านกระบวนการ (Process): ตัวชี้วัดสามารถพิจารณาได้จากสิ่งต่างๆ

ดังนี้

1.3.1 แนวปฏิบัติหลักที่กรณีศึกษาประยุกต์ใช้กับโครงการพัฒนาซอฟต์แวร์

(Key Practice)

1.4 ตัวชี้วัดด้านผลผลิต (Product) และตัวชี้วัดด้านผลกระทบ (Impact): ตัวชี้วัด

สามารถพิจารณาได้จากสิ่งต่างๆ ดังนี้

1.4.1 การตรวจสอบกิจกรรมด้านวงจรเวลาในการพัฒนา (Cycle Time) การตรวจ

สอบกิจกรรมด้านการบริหารงบประมาณ (Cost and Benefit Effective)

1.4.2 การตรวจสอบกิจกรรมด้านคุณภาพของกระบวนการพัฒนาซอฟต์แวร์

(Quality of Process)

1.4.3 การตรวจสอบกิจกรรมด้านการทำงานร่วมกันของผู้ที่เกี่ยวข้องกับโครงการ

พัฒนาซอฟต์แวร์ (Inter-Supplier Performance)

1.4.4 ผลกระทบทางบวก / เป็นผลที่คาดหวังจากการมีโครงการ

1.4.5 ผลกระทบทางลบ / เป็นผลที่ไม่คาดหวังจากโครงการ

2. ขอบเขตด้านเครื่องมือ

เครื่องมือที่ใช้ในการวิจัยครั้งนี้ได้แบบสอบถาม ซึ่งข้อมูลที่เกี่ยวข้องกับการดำเนินการวิจัยแบ่งออกเป็น 5 ตอน

ตอนที่ 1 ข้อมูลเกี่ยวกับตัวชี้วัดด้านบริบท

ตอนที่ 2 ข้อมูลเกี่ยวกับตัวชี้วัดด้านปัจจัยนำเข้า

ตอนที่ 3 ข้อมูลเกี่ยวกับตัวชี้วัดด้านกระบวนการ และผลจากการประยุกต์ใช้เทคนิค

หรือวิธีการพัฒนาซอฟต์แวร์สำหรับโครงการพัฒนาซอฟต์แวร์

ตอนที่ 4 ข้อมูลเกี่ยวกับตัวชี้วัดด้านผลผลิต และทัศนคติที่มีต่อประสิทธิภาพการประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาซอฟต์แวร์สำหรับโครงการพัฒนาซอฟต์แวร์

ตอนที่ 5 ข้อมูลเกี่ยวกับตัวชี้วัดด้านผลกระทบ และทัศนคติที่มีต่อข้อบกพร่องการประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาซอฟต์แวร์สำหรับโครงการพัฒนาซอฟต์แวร์

คำถามของการวิจัย

1. การพัฒนาซอฟต์แวร์ด้วยวิธีการ อาไจล์ (Agile) ส่งผลให้ผู้บริหารจัดการโครงการสามารถบริหารจัดการแผนการดำเนินงานอย่างมีประสิทธิภาพ (Effective Time Management) ตามระยะเวลาการพัฒนาซอฟต์แวร์ในแต่ละขั้นตอนของการพัฒนาตามหลักการของวงจรการพัฒนาซอฟต์แวร์แตกต่างจากการพัฒนาซอฟต์แวร์ด้วยวิธีการที่ไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile) หรือไม่
2. การพัฒนาซอฟต์แวร์ด้วยวิธีการ อาไจล์ (Agile) ส่งผลให้ผู้บริหารจัดการโครงการสามารถบริหารจัดการงบประมาณอย่างมีประสิทธิภาพ (Effective Cost Management) ภายใต้งบของงบประมาณแตกต่างจากการพัฒนาซอฟต์แวร์ด้วยวิธีการที่ไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile) หรือไม่
3. การพัฒนาซอฟต์แวร์ด้วยวิธีการ อาไจล์ (Agile) ส่งผลให้ประสิทธิภาพการบริหารจัดการคุณภาพของกระบวนการ (Quality of Process) แตกต่างจากการพัฒนาซอฟต์แวร์ด้วยวิธีการ ไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile) หรือไม่
4. การพัฒนาซอฟต์แวร์ด้วยวิธีการ อาไจล์ (Agile) ด้านการทำงานร่วมกันของผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ (Inter-supplier performance and Customer satisfactions) แตกต่างจากการพัฒนาซอฟต์แวร์ด้วยวิธีการที่ไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile) หรือไม่