บทที่ 3

วิธีดำเนินการวิจัย

การวิจัยเป็นการศึกษาเฉพาะกรณี (Case Study) เพื่อศึกษาข้อมูลเชิงลึกของกรณีศึกษา โดยเลือกใช้แบบการวิจัยเปรียบเทียบ โดยพิจารณาดัชนีค่าความคาดเคลื่อนของการวางแผนก่อนพัฒนากับผลของการทำงานที่เกิดขึ้นจริงในมิติด้านประสิทธิภาพการบริหารจัดการแผนการดำเนินงาน ด้านประสิทธิภาพการบริหารจัดการงบประมาณ ด้านประสิทธิภาพการบริหารจัดการคุณภาพ และด้านประสิทธิภาพการทำงานร่วมกันผู้ที่เกี่ยวข้อง โดยมีรายละเอียดการดำเนินการวิจัยดังนี้

กลุ่มตัวอย่าง

การเลือกกลุ่มตัวอย่างใช้วิธีการเลือกโดยวิธีเจาะจง (Purposive Sampling) โดยใช้วิธีการแนะนำต่อๆ กันไป ซึ่งผู้วิจัยไม่ได้ติดต่อกับกลุ่มตัวอย่างเป้าหมายในเบื้องต้น สำหรับการเลือกกลุ่มตัวอย่างที่ประยุกต์ใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile) ผู้วิจัยเลือกทีมพัฒนาที่มีประสบประการณ์ด้านการพัฒนาซอฟต์แวร์ด้วยระเบียบวิธีการพัฒนาตามแนวคิดแบบอาไจล์ (Agile) ซึ่งแนวคิดดังกล่าวเริ่มได้รับความสนใจจากนักพัฒนาซอฟต์แวร์ในวงจำกัด เนื่องจากระเบียบวิธีการพัฒนาตามแนวคิดแบบอาไจล (Agile) นี้ยังคงมีข้อถกเถียงเกี่ยวกับการนำมาประยุกต์ใช้ในสิ่งแวดล้อม และรูปแบบการพัฒนาซอฟต์แวร์ในประเทศไทย ซึ่งมีวิธีสำหรับการเข้าถึงกลุ่มตัวอย่างจากงานสัมมนาองค์วามรู้เกี่ยวกับการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile)

สำหรับกลุ่มตัวอย่างที่เลือกใช้ระเบียบวิธีการพัฒนาซอฟต์แวร์แบบไม่ใช่วิธีการอาไจล์ (Non-Agile) ผู้วิจัยเลือกกลุ่มตัวอย่างโดยอ้างอิงคุณลักษณะของเป้าหมายทางธุรกิจ หรือเป้าหมายสำหรับการทำงานของซอฟต์แวร์ของกรณีศึกษาที่มีลักษณะคล้ายคลึงกับกลุ่มตัวอย่างที่เลือกใช้วิธีการพัฒนาซอฟต์แวร์ตามแนวคิดอาไจล์ (Agile)

เครื่องมือที่ใช้ในการวิจัย

ข้อมูลที่ใช้ในการวิจัยครั้งนี้ได้จากการสร้างเครื่องมือในการเก็บข้อมูล ได้แก่ แบบสอบถาม เชิงสัมภาษณ์ และจากการสังเกต ซึ่งข้อมูลที่เกี่ยวข้องกับการดำเนินการวิจัยสามารถแบ่งได้ดังนี้

1. ข้อมูลปฐมภูมิ (Primary Data) จากการเก็บรวบรวมข้อมูลการสัมภาษณ์ร่วมกับการสังเกต และการส่งแบบสอบถาม เพื่อรวบรวมข้อเท็จจริงเกี่ยวกับคุณลักษณะขององค์กร คุณลักษณะของโครงการพัฒนาซอฟต์แวร์ และกระบวนการปฏิบัติงาน รวมทั้งข้อมูลเกี่ยวกับความคิดเห็น ความสนใจเกี่ยวกับระเบียบวิธีการพัฒนาซอฟต์แวร์ที่ทีมพัฒนาเลือกใช้

2. ข้อมูลทุติยภูมิ (Secondary Data) จากเอกสารแสดงข้อมูลเกี่ยวกับกระบวนการดำเนินงานขององค์กร เป้าหมายความสำเร็จของโครงการพัฒนาซอฟต์แวร์ รวมทั้งข้อมูลเกี่ยวกับนโยบาย วิสัยทัศน์ขององค์กรที่มีผลกระทบต่อโครงการพัฒนาซอฟต์แวร์ และข้อมูลอื่นๆ ที่เป็นประโยชน์ต่อการวิจัย โดยพิจารณาดูจากวัตถุประสงค์หรือปัญหาในการวิจัย ซึ่งมีการรวบรวมหรือเรียบเรียงไว้เรียบร้อยแล้ว ช่วยให้ผู้วิจัยนำข้อมูลนั้นมาประกอบอ้างอิงในการวิจัย

การเก็บรวมรวมข้อมูล

การสร้างเครื่องมือสำหรับรวบรวมข้อมูลเชิงลึกสำหรับใช้ในการวิจัย เพื่อเปรียบเทียบประสิทธิภาพการพัฒนาซอฟต์แวร์ด้วยระเบียบวิธีการพัฒนาซอฟต์แวร์ตามวิธีการ อาไจล์ (Agile) และดังกล่าวไม่ใช่หลักการตามแบบอาไจล์ (Non-Agile)

การสร้างเครื่องมือที่ใช้สำหรับเก็บรวบรวมข้อมูล มีขั้นตอนดังนี้

1. ศึกษารูปแบบสำหรับสร้างเครื่องวัด แนวคิด ทฤษฎี จากเอกสารและงานวิจัยที่เกี่ยวข้องกับ และประเมินผลแบบต่างๆ

2. วิเคราะห์รายละเอียดการดำเนินการพัฒนา และสิ่งแวดล้อมของโครงการพัฒนาซอฟต์แวร์

3. ศึกษาความสัมพันธ์ระหว่างเครื่องมือสำหรับเก็บรวบรวมข้อมูล และเครื่องมือที่ใช้สำหรับการวิเคราะห์ข้อมูล เพื่อสรุปผลตามวัตถุประสงค์ของการวิจัย

4. สร้างเครื่องมือสำหรับเก็บรวบรวมข้อมูล ได้แก่ แบบสอบถาม โดยมีเนื้อหาครอบคลุมตามตารางวิเคราะห์

5. ตรวจสอบคุณภาพของเครื่องมือสำหรับการวิจัยโดยการสร้างเครื่องมือสำหรับรวบรวมข้อมูลอ้างอิงจากเกณฑ์สำหรับตัวชี้วัด และลักษณะข้อมูลที่ต้องการ กระบวนการพัฒนาซอฟต์แวร์ ดังนี้

การสัมภาษณ์ เพื่อให้ได้ข้อมูลเกี่ยวกับการปฏิบัติงาน และความคิดเห็น ทัศนคติ ความเชื่อ โดยวิธีการซักถามและโต้ตอบแบบเผชิญหน้ากัน (Face-to-Face Contacts) ด้วยการสัมภาษณ์โดยใช้แบบสัมภาษณ์เชิงโครงสร้าง (Structured or Standardized Interview) ที่สร้างขึ้น

ไว้เรียบร้อยแล้ว โดยผู้ถูกสัมภาษณ์ทุกคนจะถูกถามคำถามเดียวกัน ทั้งนี้การสัมภาษณ์อาจจะมีความแตกต่างกันขึ้นอยู่กับความรู้ และประสบการณ์ของผู้ถูกการสัมภาษณ์ ดังนั้นผู้วิจัยต้องดึงคำตอบจากผู้ถูกสัมภาษณ์เพื่อให้ได้ข้อมูลตามวัตถุประสงค์ หรือประเด็นของการวิจัยที่ตั้งไว้ เข้ามาร่วมด้วย

การสังเกตโดยไม่เข้าไปร่วมด้วย (Non-Participant Observation) ผู้วิจัยสังเกตการปฏิบัติงาน โดยไม่เข้าไปมีส่วนร่วมในกิจกรรมที่กลุ่มกำลังทำอยู่เครื่องมือที่ใช้ประกอบการสังเกต ได้แก่ ตารางจดบันทึกข้อมูล (Data Gathering Schedule) แบบฟอร์มที่สร้างขึ้นเพื่อคอยตรวจสอบว่าเหตุการณ์ หรือกิจกรรมใดเกิดขึ้นจำนวนกี่ครั้งในช่วงเวลาหนึ่ง และระเบียนพฤติการณ์ (Anecdotal Record) แบบฟอร์มที่สร้างขึ้นเพื่อการจดบันทึกเหตุการณ์หรือกิจกรรมที่เกิดขึ้นจริง

การส่งแบบสอบถาม สำหรับรวบรวมข้อมูลจากผู้ให้ข้อมูลโดยตรง ทั้งนี้ผู้ให้ข้อมูลจะเป็นผู้กรอกแบบสอบถามนั้นเอง เพื่อรวบรวมข้อมูลความคิดเห็น ทัศนคติ ความเชื่อ จากผู้ให้ข้อมูล โดยผู้วิจัยออกแบบสอบถามแบบผสม ทั้งแบบสอบถามปลายเปิด ผู้ให้ข้อมูลสามารถตอบได้อย่างเสรี โดยจะเว้นช่องว่างมาให้ และแบบสอบถามปลายปิด ผู้ให้ข้อมูลสามารถเลือกตอบหรือเติมคำสั้นๆ เช่น เลือกตอบคำตอบเดียว เลือกหลายคำตอบ แบบมาตราส่วนประมาณค่า และแบบจัดอันดับความสำคัญ เพื่อให้ได้ข้อมูลครอบคลุมตามวัตถุประสงค์ของการวิจัย

1. ข้อมูลเกี่ยวกับแบบสอบถามปลายซึ่งเก็บข้อมูลเกี่ยวกับ หมายเลขแบบสอบถาม สถานที่เก็บข้อมูล และเงื่อนไขของแบบสอบถามปลาย

2. หมวดหมู่คำถาม จากแนวคิดพื้นฐานเกี่ยวกับทฤษฎี Ecosystem ช่วยให้สามารถกำหนดข้อมูลต่างๆ ในรูปแบบของวัตถุ และกำหนดความสัมพันธ์ของวัตถุเหล่านั้น แนวคิดดังกล่าวช่วยให้ผู้วิจัยได้เข้าถึงแก่นแท้ของความสัมพันธ์ระหว่างกระบวนการทางธุรกิจ และกระบวนการวิศวกรซอฟต์แวร์โดยการเก็บข้อมูลเชิงลึกจากการเก็บข้อมูลเชิงลึกด้วยแบบสอบถาม โดยจากศึกษาแนวคิดดังกล่าว และความสัมพันธ์ของสิ่งแวดล้อม ซึ่งส่งผลกระทบต่อกระบวนการพัฒนาซอฟต์แวร์ เครื่องมือช่วยเก็บข้อมูลในงานวิจัยสามารถจัดหมวดหมู่แบบสอบถามได้ 4 ตอนดังนี้

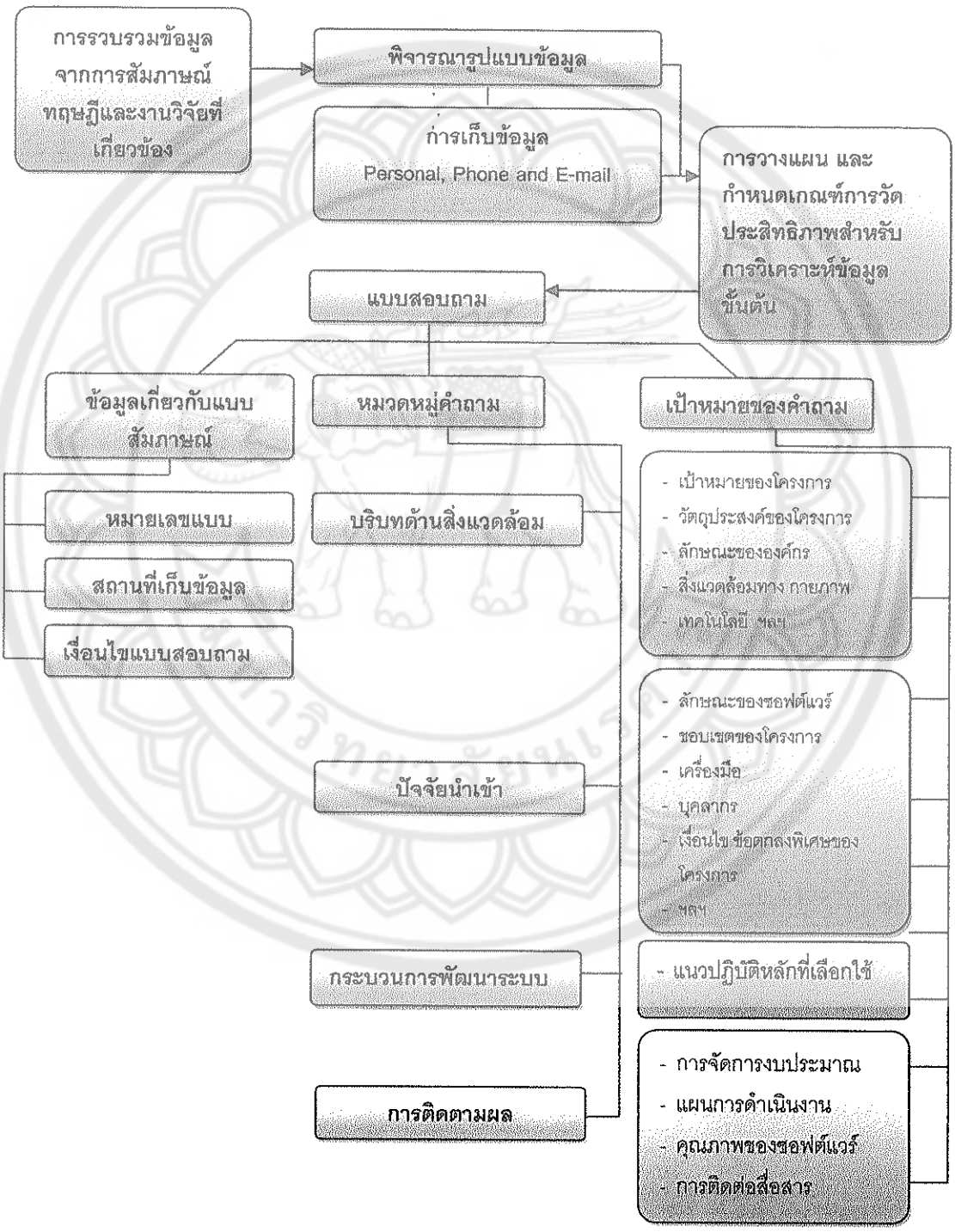
ตอนที่ 1 ข้อมูลเกี่ยวกับตัวชี้วัดด้านบริบท

ตอนที่ 2 ข้อมูลเกี่ยวกับตัวชี้วัดด้านปัจจัยนำเข้า

ตอนที่ 3 ข้อมูลเกี่ยวกับตัวชี้วัดด้านผลผลิต และทัศนคติที่มีต่อประสิทธิภาพ การประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาซอฟต์แวร์สำหรับโครงการพัฒนาซอฟต์แวร์

ตอนที่ 4 ข้อมูลเกี่ยวกับตัวชี้วัดด้านกระบวนการ และผลจากการประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาซอฟต์แวร์สำหรับโครงการพัฒนาซอฟต์แวร์

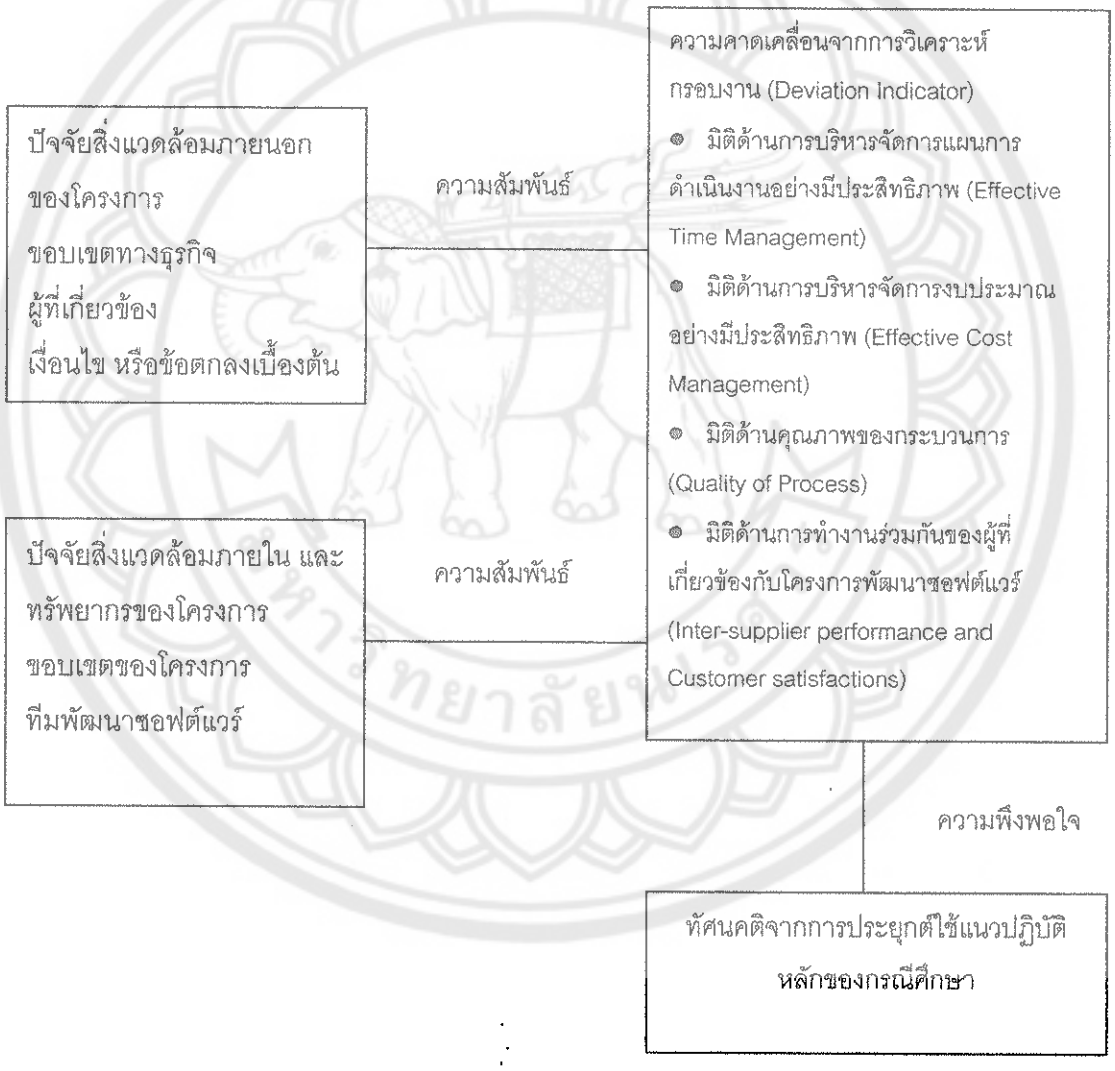
ตอนที่ 5 ข้อมูลเกี่ยวกับตัวชี้วัดด้านผลกระทบ และทัศนคติที่มีต่อข้อบกพร่อง การประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาซอฟต์แวร์สำหรับโครงการพัฒนาซอฟต์แวร์



ภาพ 16 ขั้นตอนการสร้างเครื่องมือช่วยเก็บรวบรวมข้อมูล

กรอบแนวคิดการวิจัย

ผู้วิจัยเลือกใช้แนวคิดการประเมินของสตัฟเฟิลบีม (Stufflebeam's CIPP Model) มาช่วยในการประเมินโครงการพัฒนาซอฟต์แวร์ เนื่องจากจากแนวคิดและวิธีการทางการวัดและประเมินผลดังกล่าวเป็นวิธีการประเมินกระบวนการอย่างต่อเนื่อง ซึ่งสามารถใช้ควบคู่กับการบริหารโครงการตลอดทั้งโครงการ โดยแนวคิดดังกล่าวแบ่งประเด็นการประเมินผลออกเป็น 4 ประเภท ดังภาพ 19



ภาพ 17 กรอบแนวคิดการวิจัย

1. การประเมินด้านบริบทหรือสภาวะแวดล้อม (Context Evaluation: C)

ผู้วิจัยเลือกการประเมินสภาวะแวดล้อมช่วยสนับสนุนการตัดสินเกี่ยวกับ ปัจจัยทางสภาพแวดล้อมของโครงการที่จะช่วยให้โครงการบรรลุเป้าหมายและตรวจสอบความเหมาะสมของโครงการว่าสอดคล้องกับความต้องการที่แท้จริงหรือไม่ รวมทั้งตรวจสอบความชัดเจนของวัตถุประสงค์ของโครงการว่ามีความสอดคล้องกับนโยบายขององค์กรหรือไม่

จากแนวคิดของ M. Hadzic, E. Chang, T. Dillon เกี่ยวกับความสัมพันธ์ระหว่างสิ่งแวดล้อมทางดิจิตอล (Digital) ซึ่งสัมพันธ์กับการปรับเปลี่ยน และการทำงานร่วมกันของชุมชนดิจิตอล (Digital) โดยมีผลกระทบต่อหน่วยการทำงานและเชื่อมโยงกันตลอดการไหลของสารสนเทศและกิจกรรมทางธุรกิจ ซึ่งมีความสอดคล้องกับแนวคิดของ Model-Based (System) Architecting and Software Engineering โดยพยายามเชื่อมโยงความสัมพันธ์ของโมเดลความสำเร็จ (Success Model) เข้ากับโมเดลอื่นๆ

ตาราง 1 แสดงขอบเขตการประเมินบริบทหรือสภาพแวดล้อมสำหรับการวัดกระบวนการพัฒนาซอฟต์แวร์

สิ่งแวดล้อมของการพัฒนาซอฟต์แวร์ (Environment of software development)		
ปัจจัย (Factor)	**ปัจจัยย่อย** (Sub Factor)	**ตัวชี้วัด**
บริบท (Context)	สภาวะแวดล้อมของก่อนมีโครงการ (Business area)	1. ประเภทของธุรกิจ/ องค์กร/ หน่วยงาน 2. ระยะเวลาการประกอบการขององค์กร/ หน่วยงาน 3. จำนวนผู้เชี่ยวชาญ หรือผู้ปฏิบัติงานประจำด้านการพัฒนาซอฟต์แวร์ภายในองค์กร/ หน่วยงานที่คุณปฏิบัติงาน 4. สิ่งแวดล้อมทางกายภาพ (Physical environment) ของทีมพัฒนากับสถานที่ติดตั้งซอฟต์แวร์ที่พัฒนาขึ้น โดยมีความเกี่ยวข้อง เช่น ต้องมีการแลกเปลี่ยนข้อมูลร่วมกัน

ตาราง 1 (ต่อ)

สิ่งแวดล้อมของการพัฒนาซอฟต์แวร์ (Environment of software development)		
ปัจจัย (Factor)	ปัจจัยย่อย (Sub Factor)	ตัวชี้วัด
		5. ลักษณะของโครงสร้างทีมพัฒนา หรือลักษณะการทำงาน โดยแบ่งเป็น In house/ outsource/ partnership Development/Commercial Product 6. รูปแบบวัฒนธรรมองค์กร (Business culture) ในการยอมรับเทคโนโลยี (Technology) หรือวิธีการ (Method) ใหม่เกิดขึ้น 7. จำนวนผู้ปฏิบัติงานร่วมกับซอฟต์แวร์เมื่อได้ผลิตภัณฑ์ขั้นสุดท้าย (End User) จากโครงการพัฒนาซอฟต์แวร์ 8. มาตรฐานขององค์กร โดยส่งผลต่อการพัฒนาซอฟต์แวร์ ซึ่งช่วยให้การพัฒนาซอฟต์แวร์มีประสิทธิภาพขึ้น เช่น CMMI, ISO, Software Quality Assurance, Verification and Validation
	ความพร้อมของทรัพยากร เช่น งบประมาณ เวลา และ กฎระเบียบ เป็นต้น (Constrain)	1. เงื่อนไขด้านงบประมาณที่ได้รับสำหรับโครงการพัฒนาซอฟต์แวร์ 2. เงื่อนไขด้านความกดดันในเรื่องของระยะเวลา 3. เงื่อนไขด้านความต้องการให้ซอฟต์แวร์มีความแน่นอน 4. การกำหนดแนวทางสำหรับโครงการพัฒนาซอฟต์แวร์ที่ประสบความสำเร็จ สำหรับองค์กร การกำหนดแนวทางสำหรับการบริหาร

2. การประเมินปัจจัยเบื้องต้นหรือปัจจัยป้อน (Input Evaluation: I)

การประเมินเพื่อพิจารณา ความเป็นไปได้ของโครงการอีกทั้งยังประเมินทรัพยากรที่จะใช้ในการดำเนินโครงการ เช่น บุคลากร วัสดุอุปกรณ์ งบประมาณ เวลา รวมทั้งเทคโนโลยีและแผนการดำเนินงาน ซึ่งมีความสอดคล้องกับแนวคิดของ Model-Based (System) Architecting and Software Engineering โดยพยายามเชื่อมโยงความสัมพันธ์ของโมเดลความผลิตภัณฑ์ (Product Model) เข้ากับโมเดลอื่นๆ

ตาราง 2 แสดงขอบเขตการประเมินปัจจัยเบื้องต้นสำหรับการวัดกระบวนการพัฒนาซอฟต์แวร์

สิ่งแวดล้อมของการพัฒนาซอฟต์แวร์ (Environment of software development)		
ปัจจัย (Factor)	ปัจจัยย่อย (Sub Factor)	ตัวชี้วัด
ปัจจัยนำเข้า (Input)	ขอบเขต และเทคโนโลยี (Domain and Technology)	1. รูปแบบการจัดทำเอกสารสัญญา (Contract) หรือเอกสารนำเสนอโครงการ (Proposal) สำหรับการดำเนินการพัฒนาซอฟต์แวร์ 2. ระดับความคล้ายคลึงกับโครงการเดิม หรือมีแบบอย่างมาก่อน 3. ตัวบ่งชี้ความซับซ้อนโครงการพัฒนาซอฟต์แวร์ 4. เป้าหมายของการพัฒนาซอฟต์แวร์ เช่น พัฒนาระบบใหม่ (new system) พัฒนาเพื่อปรับปรุงระบบเดิมที่มีอยู่ (enhancement to existing system) พัฒนาเพื่อแทนที่ระบบเดิมที่มีอยู่ (Replacement of existing system) เป็นต้น 5. เป้าหมายทางธุรกิจ เช่น การบัญชี/การธนาคารด้านความบันเทิง/การจัดการสารสนเทศ เป็นต้น

ตาราง 2 (ต่อ)

สิ่งแวดล้อมของการพัฒนาซอฟต์แวร์ (Environment of software development)		
ปัจจัย (Factor)	ปัจจัยย่อย (Sub Factor)	ตัวชี้วัด
		6. เป้าหมายสำหรับการทำงาน เช่น ระบบการสนับสนุนและตัดสินใจ (decision support system)/ระบบสารสนเทศเพื่อการจัดการ (management information system)/ระบบสำนักงานอัตโนมัติ (office automation system) เป็นต้น 7. รูปแบบการใช้งานโปรแกรมที่เชื่อมต่อกับผู้ใช้ เช่น Interface (console)/ Interface (windows)/ Middleware/ Web development (dynamic) เป็นต้น 8. หน้าที่รับผิดชอบภายในโครงการ เช่น พัฒนาโปรแกรม (Programmer / Developer)/ วิเคราะห์ระบบ (Analyst)/นักออกแบบซอฟต์แวร์ (Software Architect) เป็นต้น 9. จำนวนสมาชิกในทีมของโครงการพัฒนาซอฟต์แวร์ 10. ประสบการณ์การทำงานโดยเฉลี่ยของทีมที่เกี่ยวข้องกับขอบเขตทางธุรกิจ (Business Domain) ในรูปแบบเดียวกับโครงการพัฒนาซอฟต์แวร์ 11. ประสบการณ์การทำงานโดยเฉลี่ยของทีมที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ (Software development) 12. ประสบการณ์โดยเฉลี่ยของทีมหรือความคุ้นเคยกับภาษา

ตาราง 2 (ต่อ)

สิ่งแวดล้อมของการพัฒนาซอฟต์แวร์ (Environment of software development)		
ปัจจัย (Factor)	ปัจจัยย่อย (Sub Factor)	ตัวชี้วัด
		13. ประสบการณ์โดยเฉลี่ยของทีมในการทำงานกับระบบงาน (Platform) 14. ประสบการณ์โดยเฉลี่ยของทีมในการทำงานกับระเบียบวิธีการพัฒนาซอฟต์แวร์ 15. รูปแบบการติดต่อสื่อสารภายในทีมพัฒนาซอฟต์แวร์ 16. รูปแบบการติดต่อสื่อสารระหว่างทีม และผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ 17. ช่วงเวลาการทำงานแบบวงรอบ (Iterative) เพื่อจัดส่งซอฟต์แวร์ที่ให้กับลูกค้าเพื่อแสดงความก้าวหน้าของโครงการ

3. **การประเมินกระบวนการ** (Process Evaluation: P)

การประเมินระหว่างการดำเนินงานโครงการ ผู้วิจัยเลือกใช้ปัจจัยตามกรอบงานของกระบวนการซึ่งเป็นพื้นฐานของกระบวนการทางซอฟต์แวร์ ประกอบด้วยกิจกรรมต่างๆ ครอบคุลมถึงชุดกิจกรรมร่ม (Umbrella Activities) ซึ่งเป็นกิจกรรมที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ทั้งกระบวนการ ซึ่งสอดคล้องกับแนวคิดของ Model-Based (System) Architecting and Software Engineering ที่พยายามเชื่อมโยงโมเดลกระบวนการ (Process Model) กับโมเดลอื่น ๆ

ผู้วิจัยนำแนวคิดการประเมินระดับความเป็นอาไจล์ (Agile) ภายในวิธีการอาไจล์ (Agile) ทั้ง 6 วิธีของ A. Qumer, B. Henderson-Sellersd เข้ามาช่วยกำหนดขอบข่ายการประเมินค่าระดับความเป็นอาไจล์ (Agile) ทั้ง 4 มิติตามแถลงการณ์อาไจล์ (Agile) เพื่อประเมินกรณีศึกษาว่ามีการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile) หรือไม่ หากกรณีศึกษาใดมีแนวปฏิบัติหลักที่สอดคล้องกับค่าประเมินระดับความเป็นอาไจล์ (Agile) ทั้ง 4 มิติ ตามแถลงการณ์อาไจล์ (Agile) ถือว่าโครงการนั้นเป็นโครงการพัฒนาซอฟต์แวร์ตามหลักการอาไจล์ (Agile)

โดยการนำเสนอผู้วิจัยเพิ่มเติมมิติการสนับสนุนกระบวนการอาไจล์ (Keeping Process Agile) เพื่อศึกษาว่าแนวปฏิบัติหลักใดบ้างที่กรณีศึกษาเลือกใช้เอื้อให้เกิดกระบวนการอาไจล์ (Keeping Process Agile)

ตาราง 3 แสดงขอบเขตการประเมินกระบวนการสำหรับการวัดกระบวนการพัฒนาซอฟต์แวร์

สิ่งแวดล้อมของการพัฒนาซอฟต์แวร์ (Environment of software development)		
ปัจจัย (Factor)	ปัจจัยย่อย (Sub Factor)	ตัวชี้วัด
กระบวนการ (Process)	การตรวจสอบกิจกรรม เวลา และทรัพยากรของโครงการการมีส่วนร่วมของผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ ความพึงพอใจของประชาชนที่เข้าร่วมโครงการ	1. ขั้นตอนการดำเนินการพัฒนาโครงการ (Phase) และแนวปฏิบัติหลักที่เลือกใช้ (Practice) 2. กระบวนการบริหารจัดการโครงการ (Project management process) 3. แผนการจัดการโครงแบบของระบบ/กระบวนการสนับสนุน (Software configuration control process/Support process) 4. การบริหารจัดการกระบวนการ (Process management)

4. **การประเมินผลผลิต** (Product Evaluation: P)

การประเมินเพื่อเปรียบเทียบผลลัพธ์ที่เกิดขึ้นผู้วิจัยเลือกการประเมินประสิทธิภาพของโครงการ (Efficiency of Program) ตามหลักการ QCD (Quality, Cost, Delivery) ประกอบการด้วยมิติด้านการบริหารจัดการแผนการดำเนินงานอย่างมีประสิทธิภาพ (Effective Time Management), มิติด้านบริหารจัดการงบประมาณอย่างมีประสิทธิภาพ (Effective Cost Management) และมิติด้านมิติด้านคุณภาพของกระบวนการ (Quality of Process) ซึ่งผู้วิจัยเพิ่มเติมมิติด้านการทำงานร่วมกันของผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ (Inter-supplier performance and

Customer satisfactions) เนื่องจากสมดุล QCD มีการกล่าวถึงความพึงพอใจของลูกค้า ซึ่งสอดคล้องกับแนวคิดของ Model-Based (System) Architecting and Software Engineering กล่าวถึงความสำเร็จ หรือล้มเหลวของโครงการนั้นเกิดจากความต้องการของผู้ที่เกี่ยวข้องในระบบมีความสอดคล้อง หรือขัดแย้งกัน

ตาราง 4 แสดงขอบเขตการประเมินผลผลิตสำหรับการวัดกระบวนการพัฒนาซอฟต์แวร์

การกำหนดตัวชี้วัด (Measurements)		
	คุณลักษณะ (Features)	**คำอธิบาย** (Description)
1	มิติด้านการบริหารจัดการแผนการดำเนินงานอย่างมีประสิทธิภาพ (Effective Time Management)	มิติด้านวงจรในการพัฒนา ซึ่งช่วยลดระยะเวลาในการพัฒนาสำหรับการเปลี่ยนแปลงคุณภาพการทำงานให้เหมาะสมกับธุรกิจ, เพิ่มลักษณะพิเศษสำหรับการทำงาน, ปรับให้เข้ากับเทคโนโลยีใหม่, ลดระยะเวลาการส่งมอบ (Reduce delivery time)
2	มิติด้านบริหารจัดการงบประมาณอย่างมีประสิทธิภาพ (Effective Cost Management)	มิติด้านการบริหารจัดการงบประมาณ ซึ่งช่วยให้สามารถใช้งบประมาณได้อย่างมีประสิทธิภาพ สำหรับกระบวนการพัฒนาซอฟต์แวร์
3	มิติด้านคุณภาพของกระบวนการ (Quality of Process)	มิติด้านคุณภาพของกระบวนการ ซึ่งช่วยให้สามารถปรับปรุงระดับคุณภาพของกระบวนการทั้งด้าน 1. การบรรลุเป้าหมายของโครงการ (Effectiveness) ความพึงพอใจของโครงการ และผู้มีส่วนได้ส่วนเสียของโครงการ 2. ความอยู่รอดของโครงการด้านระยะเวลา และค่าใช้จ่าย (Sustainability) 3. การกระจายงานในแต่ละขั้นตอน (Equity) 4. ผลผลิตเปรียบเทียบกับเป้าหมายรวมกิจกรรมที่ทำแล้วเสร็จ ทรัพยากร และเวลาที่ใช้ไปได้ตรงตามเป้าหมาย (Progress) 5. ความพอเพียงของทรัพยากร (Adequacy)

ตาราง 4 (ต่อ)

การกำหนดตัวชี้วัด (Measurements)		
	คุณลักษณะ (Features)	**คำอธิบาย** (Description)
		6. ลดความสูญเสียที่เกิดขึ้นสำหรับโครงการพัฒนาซอฟต์แวร์
4	มิติด้านการทำงานร่วมกันของผู้ที่เกี่ยวข้องกับโครงการพัฒนาซอฟต์แวร์ (Inter-supplier performance and Customer satisfactions)	มิติด้านความร่วมมือกัน และช่วยสนับสนุนความสามารถใน 1. การบรรลุข้อตกลงของโครงการ (Collaborative) 2. การทำหน้าที่เป็นตัวแทน สร้างองค์กรเสมือนจริง ด้วยการสร้างแม่แบบ ประกอบด้วยมุมมองที่สามารถแลกเปลี่ยนแนวคิดกันระหว่างกลุ่ม (Representative) 3. การมีสิทธิตัดสินใจเฉพาะหน้าในเรื่องการเปลี่ยนแปลงความต้องการ (Authorized) 4. การให้ความสำคัญกับสมาชิกในทีมแต่ละคนเท่าๆ กัน โดยการมอบหมายความดูแล หรือความรับผิดชอบสำหรับการพัฒนา (Committed) 5. สร้างความรู้ความเข้าใจในระบบงาน และช่วยสนับสนุนให้เกิดการแลกเปลี่ยนความรู้เกี่ยวกับระบบงานที่จะพัฒนาของสมาชิกในทีม (Knowledgeable)

การวิเคราะห์ข้อมูลและกระบวนการหาความเที่ยงตรง

งานวิจัยมีการตรวจสอบความเที่ยงตรง และความเชื่อมั่นของแบบสอบถามเพื่อบ่งชี้คุณภาพของแบบทดสอบที่สร้างขึ้น โดยมีกระบวนการหาความเที่ยงตรงและความเชื่อมั่นดังนี้

1. **ความเที่ยงตรงตามเนื้อหา** (Content Validity)

เพื่อพิจารณาว่าแบบสอบถามมีความครอบคลุมขอบเขตด้านเนื้อหาหรือไม่จากการหาค่าดัชนีความสอดคล้อง (Index of Consistency: IOC) IOC ซึ่งประเมินแบบสอบถามแต่ละข้อกับตารางวิเคราะห์รายละเอียดดังนี้

ถ้าแน่ใจว่า "ตรง" จะกาเครื่องหมายในช่อง "+1"

ถ้าแน่ใจว่า "ไม่ตรง" จะกาเครื่องหมายในช่อง "-1" และ

ถ้าไม่แน่ใจว่า "ตรงหรือไม่" จะกาเครื่องหมายในช่อง "0"

จากนั้นหาดัชนีความสอดคล้อง (Index of Consistency: IOC) IOC = $\Sigma R/N$

เมื่อ N คือ จำนวนผู้เชี่ยวชาญ และ

R คือ ค่าคะแนนความสอดคล้อง

หากค่า IOC มีค่าตั้งแต่ .50 ขึ้นไปถือว่าแบบสอบถามข้อนั้นสร้างตรงตามตารางวิเคราะห์รายละเอียด ด้วยวิธีการหาความเที่ยงตรงเชิงเหตุผล (Logical Validity) (บุญชม ศรีสะอาด, 2535)

รายชื่อผู้เชี่ยวชาญตรวจสอบหาความเที่ยงเชิงประจักษ์ (Face Validity) ผู้เชี่ยวชาญประกอบด้วย

1. ผู้ช่วยศาสตราจารย์ ดร.อรสา เตติวัฒน์
   ตำแหน่งทางวิชาการ/คุณวุฒิ อาจารย์
   คณะต้นสังกัดคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร
2. ผู้ช่วยศาสตราจารย์ ดร.ภูพงษ์ พงษ์เจริญ
   ตำแหน่งทางวิชาการ/คุณวุฒิ อาจารย์
   คณะต้นสังกัดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
3. อาจารย์ภานุวัฒน์ เรือนมูล
   ผู้เชี่ยวชาญด้านระบบสารสนเทศ โครงการวิจัยสุขภาพจิตคนไทย กระทรวงสาธารณสุข ประธานคณะกรรมการสารสนเทศ และหัวหน้าแผนกคอมพิวเตอร์ โรงพยาบาลโอเวอร์บรุ๊ค จังหวัดเชียงราย
4. คุณวุฒิชัย ทวีชื่นสกุล
   ผู้บริหารจัดการโครงการพัฒนาซอฟต์แวร์ (Project Manager) บริษัท Norton Rose
5. คุณมานิตย์ แก้วมานอม
   ผู้บริหารจัดการโครงการพัฒนาซอฟต์แวร์ (Project Manager) บริษัท Innovation Group.

ผลของการตรวจเครื่องมือที่ใช้ในงานวิจัยเชิงวัตถุประสงค์ ค่า IOC ไม่ต่ำกว่า 0.80 เกือบทุกข้อมีบางข้อที่ต้องทำการปรับปรุงแก้ไข โดยภาพรวมแบบสอบถามมีความสอดคล้อง ภาษาที่ใช้เหมาะสม เข้าใจง่าย ข้อคำถามตรงกับวัตถุประสงค์ของงานวิจัยดี

2. วิธีการ Triangulation

การสร้างเครื่องมือสำหรับรวบรวมข้อมูลจากแบบสอบถาม ผลการสัมภาษณ์การสังเกต ผู้วิจัยคำนึงถึงคุณภาพของเครื่องมือ และการวิเคราะห์ผลดังนั้นจึงตรวจสอบคุณภาพของข้อมูล ตามวิธีการ Triangulation ใน 7 มิติ ดังนี้

1. มิติด้านข้อมูล (Data Triangulation) การเปรียบเทียบและตรวจสอบความเที่ยงตรงของข้อมูลโดยนำข้อมูลที่รวบรวมจากแหล่งต่างๆ ซึ่งมีความแตกต่างด้านเวลา และบุคคลมาเปรียบเทียบ

2. มิติด้านวิธีการ (Methodological Triangulation) โดยเปรียบเทียบข้อมูลที่ได้จากการรวบรวมข้อมูลหลายวิธีการเพื่อยืนยันสภาพปรากฏการณ์เดียวกัน

3. มิติด้านทฤษฎี (Theory Triangulation) โดยใช้มุมมองของทฤษฎีมาพิจารณาข้อมูล เพื่อช่วยอธิบายในปรากฏการณ์นั้นๆ เพื่อให้เกิดมุมมองที่หลากหลาย

4. มิติด้านผู้วิจัย (Multiple Investigator Triangulation) การเก็บข้อมูลในประเด็นวิจัยเดียวกันในสภาวะเดียวกัน เพื่อช่วยให้ข้อมูลนั้นๆ มีความรอบด้านมากยิ่งขึ้น

นอกจากวิธีการ Triangulation ช่วยให้ข้อมูลที่รวบรวมได้มีความเที่ยงตรงแล้ว ยังช่วยให้ผู้วิจัยมองเห็นปัญหา และสามารถให้อภิปรายผล หรือปรากฏการณ์ต่างๆ ได้ดียิ่งขึ้น เนื่องจากเป็นวิธีการนำข้อมูลที่ได้รับมาเปรียบเทียบ ซึ่งข้อมูลที่รวบรวมได้นั้นอาจมีคล้ายคลึงกัน ข้อมูลไม่เสมอต้นเสมอปลาย หรือขัดแย้งกันขึ้นอยู่กับมุมมองพื้นฐานเกี่ยวกับแนวคิดของผู้ให้ข้อมูล

สถิติที่ใช้สำหรับการวิจัย

ข้อมูลที่รวบรวมได้ ผู้วิจัยจะนำไปวิเคราะห์และคำนวณหาค่าสถิติ โดยใช้โปรแกรมสำเร็จรูป SPSS for Windows version 11.5 เป็นเครื่องมือในการวิเคราะห์ ค่าสถิติต่างๆ ดังนี้

1. ข้อมูลที่มีลักษณะไม่ต่อเนื่อง (Categorical Data หรือ Discrete Data) จากการเก็บรวบรวมข้อมูลในระดับนามมาตราและอันดับมาตรา การจัดรูปแบบข้อมูลแบบสัดส่วน เช่น อัตราและร้อยละ แล้วนำเสนอในรูปตาราง

2. ข้อมูลที่มีลักษณะต่อเนื่อง (Continuous Data) จากการเก็บรวบรวมข้อมูลในระดับขั้นมาตราและอัตราส่วนมาตรา การจัดรูปแบบข้อมูลในลักษณะของการวัดแนวโน้มเข้าสู่ศูนย์กลาง เช่น ค่าเฉลี่ย มัธยฐาน ฐานนิยม หรือพิสัย (อัจฉรีย์ จันทลักขณา, 2544)

2.1 การหาค่าร้อยละ (Percentage)

เพื่อหาค่าร้อยละความเป็นอาไจล์ (Agile) ของกรณีศึกษาแต่ละโครงการโดยใช้สูตรดังนี้

$$P = \frac{f}{n} \times 100$$

P หมายถึง ค่าร้อยละ(percentage)

f หมายถึง ความถี่หรือจำนวนข้อมูล

X หมายถึง ค่าของข้อมูลหรือคะแนน

n หมายถึง จำนวนตัวอย่างหรือผู้ตอบแบบสอบถาม

2.2 **ค่าเฉลี่ยตัวอย่าง** (Sample mean = $\bar{x}$)

เพื่อหาค่าเฉลี่ยกับทัศนคติที่มีต่อมิติด้านประสิทธิภาพการบริหารจัดการคุณภาพ (Quality of process) และมิติด้านประสิทธิภาพการทำงานร่วมกันผู้ที่เกี่ยวข้อง (inter-supplier performance) หลังจากประยุกต์ระเบียบวิธีการพัฒนาสำหรับโครงการพัฒนาซอฟต์แวร์ และความพึงพอใจของกรณีศึกษาที่มีต่อประสิทธิภาพการประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาระบบสำหรับโครงการพัฒนาซอฟต์แวร์โดยใช้สูตรดังนี้โดยใช้สูตรดังนี้

$$\bar{X} = \frac{\sum fx}{n}$$

$\bar{X}$ หมายถึง ค่าเฉลี่ยของตัวอย่าง

$\sum fx$ หมายถึง ผลรวมของข้อมูลทั้งหมด

n หมายถึง จำนวนตัวอย่างที่ตอบแบบสอบถาม

ตาราง 5 แสดงค่าคะแนนเฉลี่ยระดับความพึงพอใจที่มีต่อแนวปฏิบัติที่ประยุกต์ใช้สำหรับโครงการพัฒนาซอฟต์แวร์

คะแนน ระดับความพึงพอใจ ช่วงคะแนน
5 พึงพอใจมากที่สุด 4.51-5.00
4 พึงพอใจมาก 3.51-4.50
3 พึงพอใจปานกลาง 2.51-3.50
2 พึงพอใจน้อย 1.51-2.50
1 พึงพอใจน้อยที่สุด 1.00-1.50

2.3 การวัดความกระจายด้วยส่วนเบี่ยงเบนมาตรฐาน (S.D)

เพื่อวัดความกระจายด้วยส่วนเบี่ยงเบนมาตรฐานกับทัศนคติที่มีต่อมิติด้านประสิทธิภาพการบริหารจัดการคุณภาพ (Quality of process) และมิติด้านประสิทธิภาพการทำงานร่วมกันผู้ที่เกี่ยวข้อง (inter-supplier performance) หลังจากประยุกต์ระเบียบวิธีการพัฒนาสำหรับโครงการพัฒนาซอฟต์แวร์ และความพึงพอใจของกรณีศึกษาที่มีต่อประสิทธิภาพการประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาระบบสำหรับโครงการพัฒนาซอฟต์แวร์โดยใช้สูตรดังนี้โดยใช้สูตรดังนี้

$$S.D. = \sqrt{\frac{n(\sum fx^2) - (\sum x)^2}{n(n-1)}}$$

S.D. หมายถึง ส่วนเบี่ยงเบนมาตรฐาน

$\sum fx^2$ หมายถึง ผลรวมกำลังสอง

$(\sum fx)^2$ หมายถึง ผลรวมของข้อมูล

N หมายถึง ขนาดของตัวอย่าง

หลังจากรวบรวมข้อมูลและแจกแจงความถี่แล้ว จากนั้นนำคะแนนเฉลี่ยของกลุ่มตัวอย่างมาพิจารณาระดับความพึงพอใจ ดังนี้

$$\text{ระดับ} = \frac{\text{คะแนนสูงสุด - คะแนนต่ำสุด}}{\text{จำนวนชั้น}}$$

2.4 ดัชนีวัดค่าความคาดเคลื่อนของแผนการดำเนินงาน (% of Schedule Deviation Indicator)

$$\%SDI = \frac{\% \text{ of (Actual Schedule} - \text{Planned Schedule)}}{\text{Planned Schedule}}$$

ค่า SDI ที่ได้จากการคำนวณจะเป็นตัวบอกว่า แผนการดำเนินงาน (Schedule) ในแผนงานมีความคลาดเคลื่อนไปจากที่ได้วางแผนไว้เท่าไร

2.5 **ดัชนีวัดค่าความคาดเคลื่อนของ** Schedule **จากแผนงานโดยรวม** (% of Schedule Deviation Compared with the total project Indicator)

$$\%SDSI = \frac{\%\text{ of (Actual Schedule} - \text{Planned Schedule)}}{\text{Planned Schedule}}$$

ค่า SDSI สามารถบอกได้ว่าแผนการดำเนินงาน (Schedule) ที่คาดไว้ในแผนงานมีค่าเบี่ยงเบนของงานในแต่ละงาน ที่อาจทำส่งผลกระทบต่อโครงการโดยรวมได้อย่างไรบ้าง

2.6 **ดัชนีวัดค่าความคาดเคลื่อนของการใช้งบประมาณ** (% of Budget Deviation Indicator)

$$\%SDI = \frac{\%\text{ of (Actual Budget} - \text{Planned Budget)}}{\text{Planned Budget}}$$

ค่า SDI ที่ได้จากการคำนวณจะเป็นตัวบอกว่าการใช้งบประมาณ (Budget) ในแผนงานมีความคลาดเคลื่อนไปจากที่ได้วางแผนไว้เท่าไร

2.7 **ดัชนีวัดค่าความคาดเคลื่อนของ** Schedule **จากแผนงานโดยรวม** (% of Schedule Deviation Compared with the total project Indicator)

$$\%SDSI = \frac{\%\text{ of (Actual Budget} - \text{Planned Budget)}}{\text{Planned Budget}}$$

ค่า SDSI สามารถบอกได้ว่าการใช้งบประมาณ (Budget) ที่คาดไว้ในแผนงานมีค่าเบี่ยงเบนของงานในแต่ละงาน ที่อาจทำส่งผลกระทบต่อโครงการโดยรวมได้อย่างไรบ้าง

เมื่อได้ค่าความคาดเคลื่อนจากการคำนวณค่าแผนการดำเนินงาน (Schedule) **และการใช้งบประมาณ** (Budget) **แล้วสามารถนำค่าที่ได้มาเปรียบเทียบในตาราง 6 เพื่อให้สามารถวิเคราะห์การดำเนินงานได้อย่างมีประสิทธิภาพมาก** (เมสินี นาคมณี, 2547)

ตาราง 6 แสดงตารางวิเคราะห์ความคลาดเคลื่อนแผนการดำเนินงาน (Schedule) และการใช้งบประมาณ (Budget)

Type/Risk	Very high	High	Nominal	Low	Very Low
Schedule	> 15%	10% - 14%	6% - 9%	0% - 5%	< or = 0%
Budget	> 30%	15% - 29%	4% - 14%	0% - 3%	< or = 0%

2.8 **มัธยฐาน (Median)**

เพื่อหาค่าคะแนนที่อยู่ตรงกลางจากการให้คะแนนความพึงพอใจของกรณีศึกษาที่มีต่อประสิทธิภาพการประยุกต์ใช้เทคนิค หรือวิธีการพัฒนาระบบสำหรับโครงการพัฒนาซอฟต์แวร์