บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 แนวคิดในการทำวิทยานิพนธ์

จะทำการพัฒนาแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ โดยมีขั้นตอนการทำงาน 3 ขั้นตอน ดังนี้

2.1.1 แนวคิดในการสร้างขั้นตอนการทำงานในการจัดทำเอกสารข้อกำหนดความต้องการซอฟต์แวร์

การสร้างขั้นตอนการทำงานในการจัดทำเอกสารข้อกำหนดความต้องการซอฟต์แวร์ จะนำแนวทางจากระเบียบวิธีการวิเคราะห์และออกแบบโครงสร้างของระบบมาใช้ในการสร้าง

2.1.2 แนวคิดในการจัดระเบียบของเอกสารข้อกำหนดความต้องการซอฟต์แวร์

การจัดระเบียบของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ จะใช้การจัดระเบียบของมาตรฐาน IEEE Std 830-1993 เป็นแนวทางในการจัดทำ

2.1.3 แนวคิดในการพัฒนาเครื่องมือซอฟต์แวร์ที่ใช้ในการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์

เครื่องมือที่จะพัฒนา สามารถเพิ่ม ลบ บันทึก เปลี่ยนแปลงแก้ไข แสดงรายละเอียดบนหน้าจอ และพิมพ์ออกทางเครื่องพิมพ์ พร้อมทั้งมีวิซาร์ด (wizard) ซึ่งประกอบด้วยคำอธิบายและตัวอย่างของแต่ละเนื้อหาในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เพื่อช่วยให้ผู้พัฒนาซอฟต์แวร์จัดทำเอกสารข้อกำหนดความต้องการซอฟต์แวร์ได้ง่าย โดยจะจำแนกความสำคัญของผู้พัฒนาซอฟต์แวร์ออกเป็น 3 ระดับ คือ ระดับเริ่มต้น (beginner level) ระดับปานกลาง (intermediate level) และระดับผู้เชี่ยวชาญ (expert level)

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 วัฏจักรของการพัฒนาซอฟต์แวร์

ในการพัฒนาซอฟต์แวร์ให้กับผู้ใช้หรือลูกค้า ผู้พัฒนาซอฟต์แวร์ทำตามวัฏจักรของการพัฒนาซอฟต์แวร์ ซึ่งแบ่งออกเป็น 4 วิธี ได้แก่

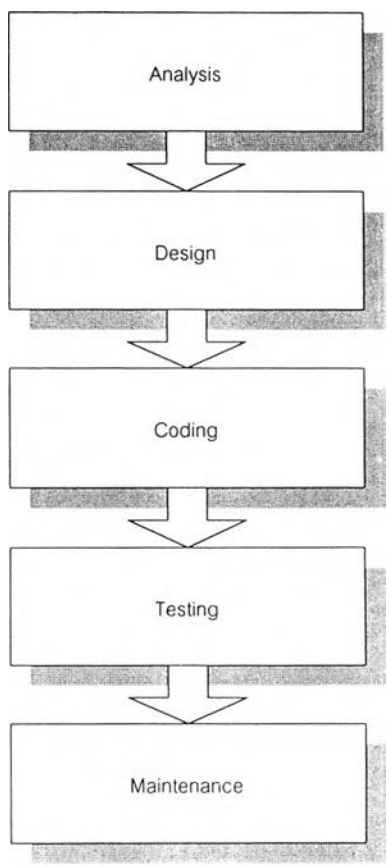
หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2.1.1 แบบจำลองน้ำตก (Waterfall model)

แบบจำลองน้ำตก (Pressman, 1992) เป็นรูปแบบหนึ่งในวัฏจักรของการพัฒนาซอฟต์แวร์ เรียกอีกอย่างหนึ่งว่า แบบจำลองเชิงเส้นโดยลำดับ (linear sequential model) แบบจำลองนี้มีแนวความคิดแบบเรียงลำดับ ซึ่งมีการทำงานตั้งแต่ขั้นตอนแรกไปยังขั้นตอนสุดท้าย โดยไม่มีการย้อนกลับ แบ่งการทำงานออกเป็น 5 ขั้นตอน คือ

1. การวิเคราะห์ระบบ (Analysis)
2. การออกแบบระบบ (Design)
3. การเขียนโปรแกรม (Coding)
4. การทดสอบระบบ (Testing)
5. การบำรุงรักษาระบบ (Maintenance)

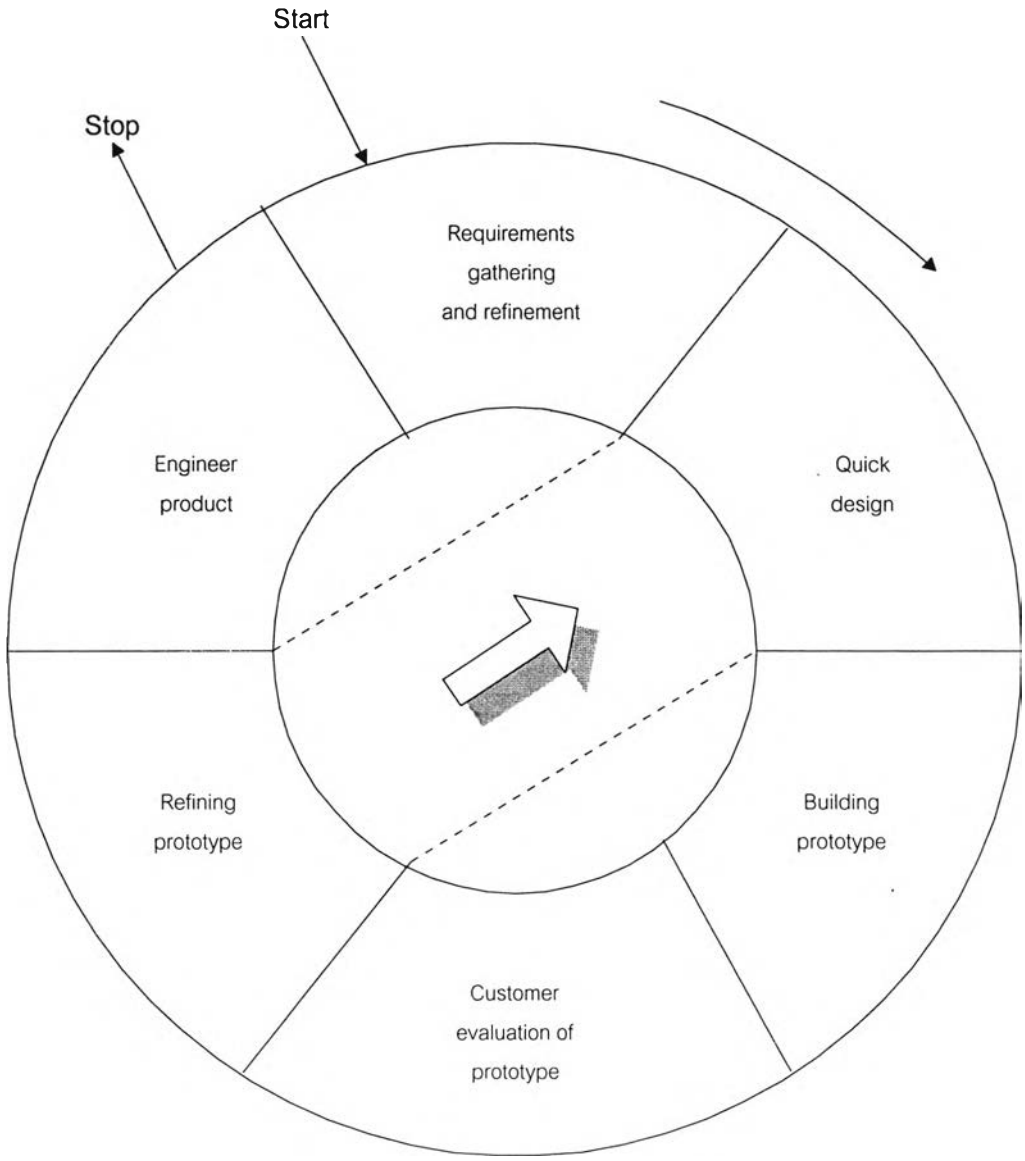
ในขั้นตอนของการพัฒนาซอฟต์แวร์ นักวิเคราะห์ระบบทำการสอบถามความต้องการจากผู้ใช้หรือลูกค้า เพื่อระบุเป็นข้อกำหนด ซึ่งข้อกำหนดนี้ใช้เป็นหลักฐานในการส่งมอบซอฟต์แวร์ให้กับลูกค้า เพื่อตรวจสอบว่าซอฟต์แวร์ทำหน้าที่ได้ตรงตามความต้องการของลูกค้าหรือไม่ โดยนักวิเคราะห์ระบบจะทำการส่งมอบงานให้กับนักออกแบบระบบ เพื่อทำการออกแบบระบบ จากนั้นโปรแกรมเมอร์จะทำการเขียนโปรแกรมตามที่ได้ออกแบบไว้ พร้อมทั้งทำการทดสอบ และบำรุงรักษาระบบ เนื่องจากผู้ใช้หรือลูกค้าอาจมีการเปลี่ยนแปลงความต้องการในภายหลังหรือระบบอาจมีความผิดพลาดเกิดขึ้นขณะใช้งาน ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แบบจำลองน้ำตก

2.2.1.2 **แบบจำลองเชิงวิวัฒน์** (Evolutionary model)

แบบจำลองเชิงวิวัฒน์ (Pressman, 1992) เป็นรูปแบบหนึ่งในวัฏจักรของการพัฒนาซอฟต์แวร์ เริ่มจากนักวิเคราะห์ระบบทำการสอบถามความต้องการจากผู้ใช้หรือลูกค้า พร้อมทั้งเก็บรวบรวมข้อมูลที่ได้ และส่งมอบงานให้กับนักออกแบบระบบ เพื่อทำการออกแบบระบบโดยเร็ว และทำการสร้างต้นแบบของซอฟต์แวร์ เพื่อให้ลูกค้าทำการประเมินต้นแบบว่าพอใจหรือไม่ โดยนักออกแบบระบบทำการปรับปรุงต้นแบบให้ดีขึ้น และทำขั้นตอนการออกแบบระบบโดยเร็วซ้ำอีกครั้งจนกระทั่งลูกค้าพอใจต้นแบบ จากนั้นส่งมอบงานให้กับโปรแกรมเมอร์เพื่อทำการพัฒนาซอฟต์แวร์จนกระทั่งส่งมอบซอฟต์แวร์ให้กับลูกค้า ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แบบจำลองเชิงวิวัฒน์

2.2.1.3 แบบจำลองขดหอย (Spiral model)

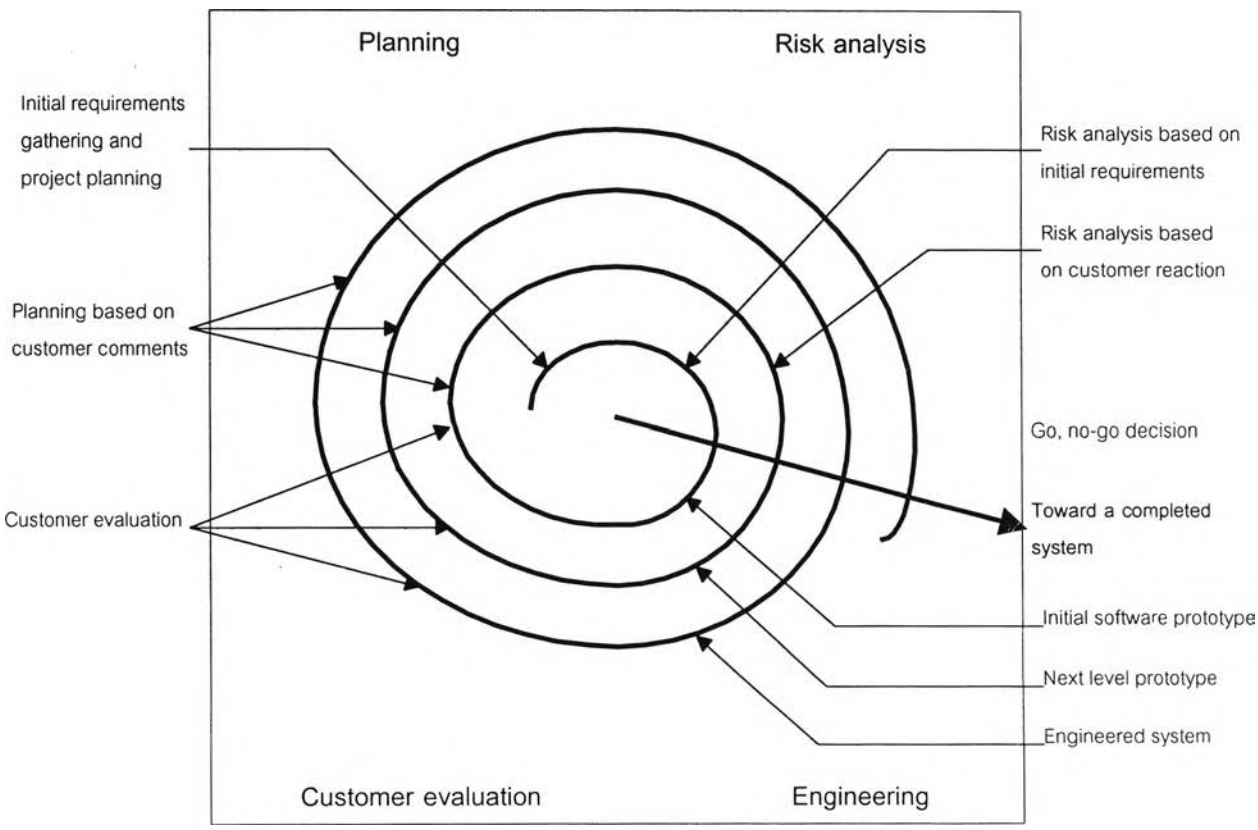
แบบจำลองขดหอย (Pressman, 1992) เป็นรูปแบบหนึ่งในวัฏจักรของการพัฒนาซอฟต์แวร์ ซึ่งเป็นการนำแบบจำลองน้ำตก และแบบจำลองเชิงวิวัฒน์ผสมเข้าด้วยกัน ลักษณะการทำงานเป็นแบบไม่เรียงตามลำดับ (non-sequential) ทำงานตั้งแต่ขั้นตอนแรกวนไปเรื่อยๆ ไม่หยุดจนกระทั่งได้ซอฟต์แวร์ตามต้องการ โดยแบ่งการทำงานออกเป็น 4 ขั้นตอน คือ

1. *การวางแผน (Planning)* เป็นการกำหนดวัตถุประสงค์ ทางเลือก และข้อจำกัดต่างๆ

2. *การวิเคราะห์ความเสี่ยง (Risk analysis)* เป็นการวิเคราะห์ทางเลือกต่างๆ และระบุถึงความเสี่ยงที่อาจเกิดขึ้น

3. *วิศวกรรม (Engineering)* ทำการพัฒนาซอฟต์แวร์ โดยการสร้างต้นแบบของซอฟต์แวร์

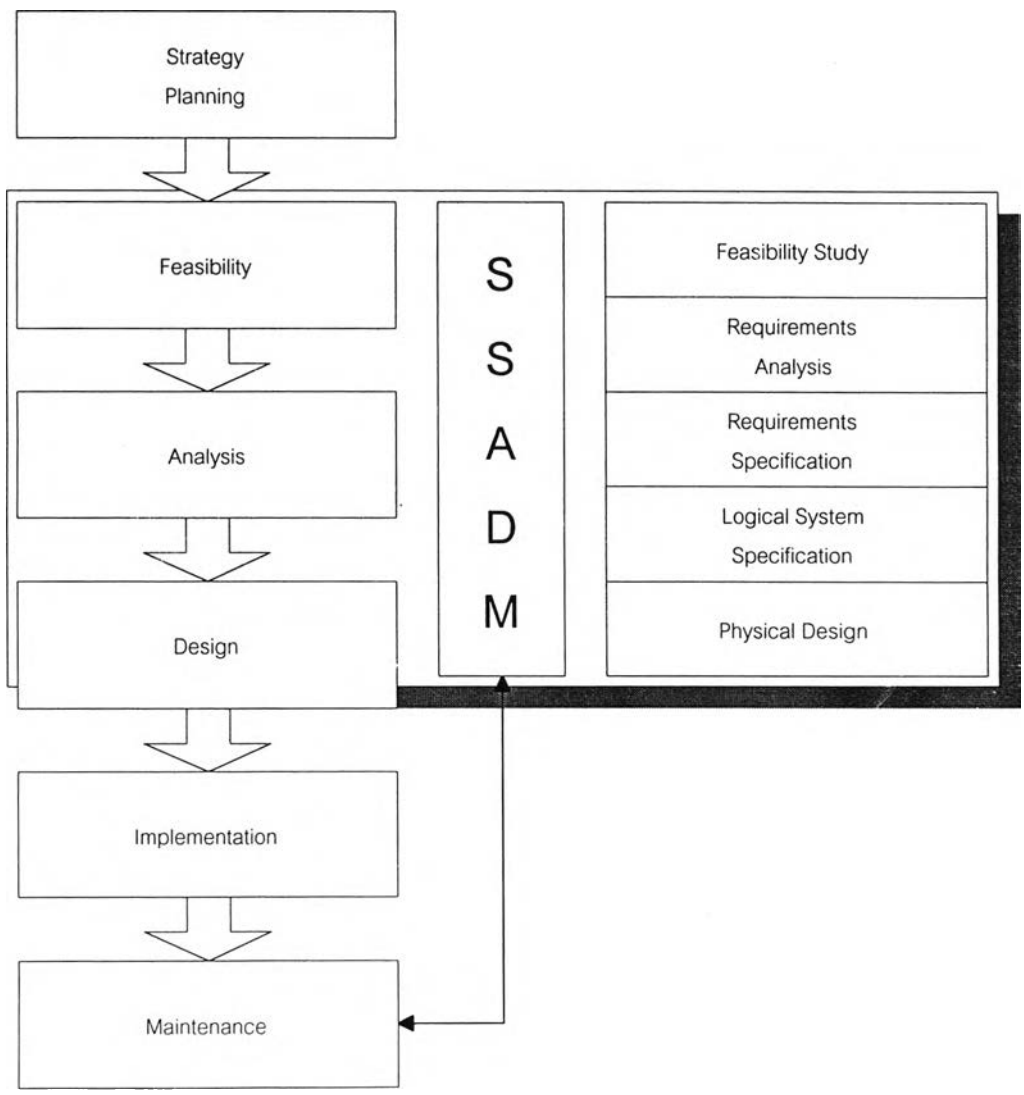
4. *การประเมินผลของลูกค้า (Customer evaluation)* ลูกค้าทำการประเมินผลลัพธ์ที่ได้จากขั้นตอนวิศวกรรม โดยประเมินต้นแบบว่าใช้ได้หรือไม่



รูปที่ 2.3 แบบจำลองขดหอย

จากรูปที่ 2.3 แสดงถึงแบบจำลองขดหอย เริ่มจากนักวิเคราะห์ระบบทำการรวบรวมความต้องการจากผู้ใช้หรือลูกค้า และทำการวางแผนโครงการที่จะพัฒนา ทำการวิเคราะห์ความเสี่ยงโดยพิจารณาจากความต้องการเบื้องต้นของลูกค้า ถ้ามีความเสี่ยงน้อยและสามารถพัฒนาซอฟต์แวร์ต่อไปได้ ผู้พัฒนาซอฟต์แวร์จะทำการสร้างต้นแบบของซอฟต์แวร์เป็นครั้งแรก และนำต้นแบบนี้ให้ลูกค้าทำการประเมินว่าใช้ได้หรือไม่ ผู้พัฒนาซอฟต์แวร์ทำการวางแผนโดยพิจารณาจากคำวิจารณ์ของลูกค้า และนำมาวิเคราะห์ความเสี่ยงโดยพิจารณาจากปฏิกิริยาของลูกค้าหลังจากใช้ต้นแบบแล้ว ถ้ามีความเสี่ยงน้อย ผู้พัฒนาซอฟต์แวร์จะทำการสร้างต้นแบบในระดับถัดไปและนำต้นแบบให้ลูกค้าทำการประเมิน ผู้พัฒนาซอฟต์แวร์ทำการวางแผนโดยพิจารณาจากคำวิจารณ์ของลูกค้า จากนั้นโปรแกรมเมอร์ทำการเขียนโปรแกรมตามที่ได้ออกแบบไว้ ทดสอบการใช้งานของโปรแกรม และนำโปรแกรมส่งให้ลูกค้าเพื่อทำการประเมินซอฟต์แวร์ ผู้พัฒนาซอฟต์แวร์นำคำวิจารณ์ของลูกค้ามาปรับปรุงซอฟต์แวร์ให้ดียิ่งขึ้น จนกระทั่งส่งมอบซอฟต์แวร์ให้กับลูกค้า

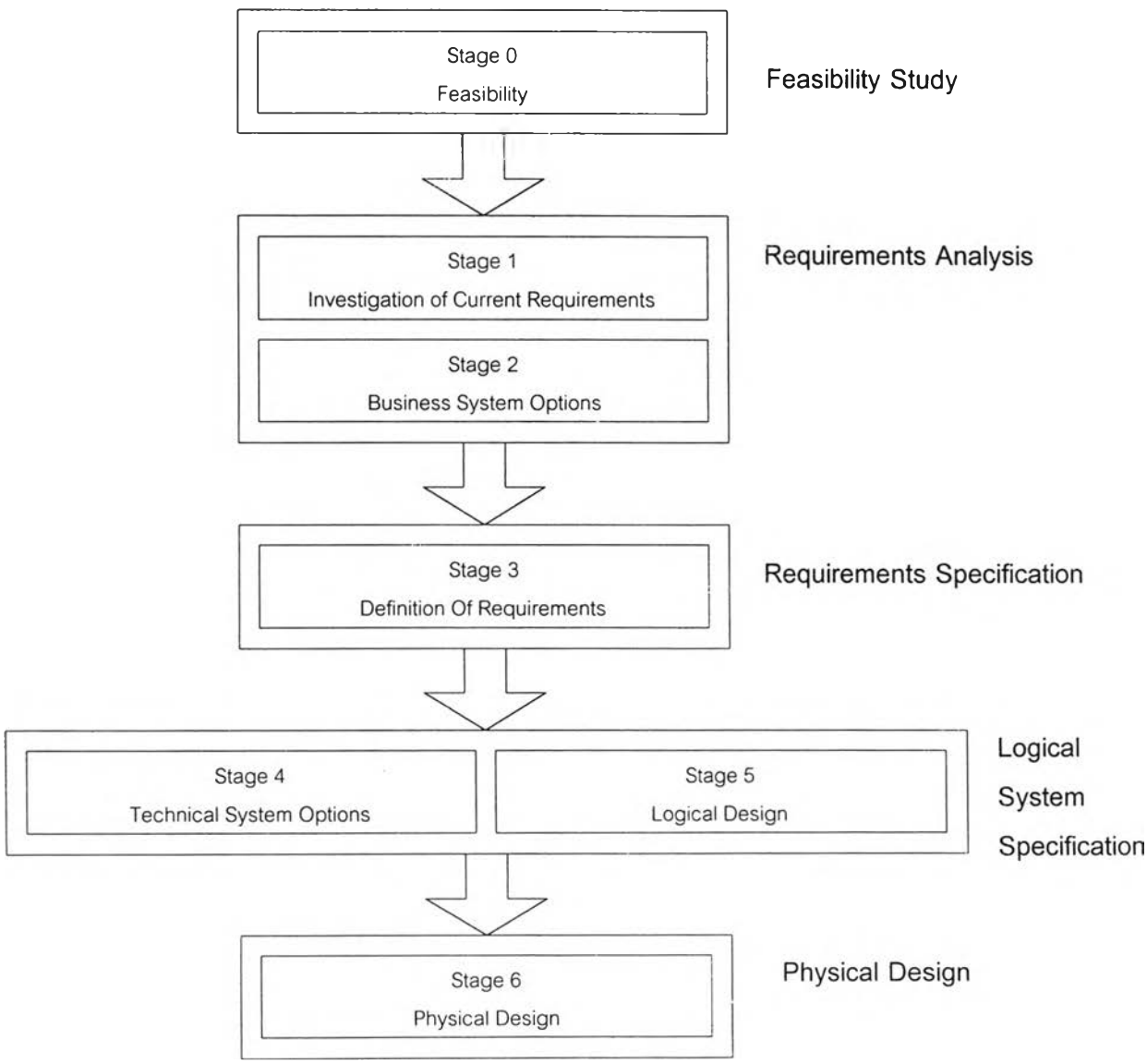
2.2.1.4 ระเบียบวิธีในการวิเคราะห์และออกแบบโครงสร้างของระบบ



รูปที่ 2.4 ระเบียบวิธีในการวิเคราะห์และออกแบบโครงสร้างของระบบ

จากรูปที่ 2.4 แสดงถึงระเบียบวิธีในการวิเคราะห์และออกแบบโครงสร้างของระบบ (Weaver, 1994) ในขั้นตอนของการพัฒนาซอฟต์แวร์ ระเบียบวิธีในการวิเคราะห์และออกแบบโครงสร้างของระบบ จะเน้นเฉพาะในส่วนของการวิเคราะห์และออกแบบระบบเท่านั้น ซึ่งขั้นตอนการทำงานของระเบียบวิธีนี้แบ่งออกเป็น 5 ส่วนหลัก คือ

1. ศึกษาความเป็นไปได้ (Feasibility Study)
2. การวิเคราะห์ความต้องการ (Requirements Analysis)
3. การเขียนข้อกำหนดความต้องการ (Requirements Specification)
4. การเขียนข้อกำหนดระบบเชิงตรรกะ (Logical System Specification)
5. การออกแบบเชิงกายภาพ (Physical Design)



รูปที่ 2.5 ขั้นตอนของระเบียบวิธีในการวิเคราะห์และออกแบบโครงสร้างของระบบ

จากรูปที่ 2.5 แสดงถึงขั้นตอนของระเบียบวิธีในการวิเคราะห์และออกแบบโครงสร้างของระบบ ซึ่งแบ่งขั้นตอนการทำงานออกเป็นขั้นตอนย่อย คือ

1. *ศึกษาความเป็นไปได้ (Feasibility Study)* แสดงให้เห็นถึงภาพรวมของระบบภายในองค์กร พร้อมทั้งศึกษาความเป็นไปได้ของระบบ โดยพิจารณาถึงประโยชน์ที่คาดว่าจะได้รับและประมาณค่าใช้จ่ายที่ใช้ภายในระบบ

2. *การวิเคราะห์ความต้องการ (Requirements Analysis)* แบ่งการทำงานออกเป็นขั้นตอนย่อย 2 ขั้นตอน คือ

2.1 *การสำรวจความต้องการที่มีอยู่ในปัจจุบัน (Investigation of Current Requirements)* เป็นการศึกษาและวิเคราะห์ระบบที่มีอยู่ในปัจจุบัน โดยพิจารณาถึงหน้าที่ ขอบเขตของโครงการ และระบุถึงความต้องการของผู้ใช้หรือลูกค้า

2.2 *ทางเลือกของระบบธุรกิจ (Business System Options)* เป็นการพิจารณาถึงทางเลือกที่หลากหลายที่ใช้ในการแก้ปัญหาทางธุรกิจ ซึ่งทางเลือกหนึ่ง คือ การกำหนดความต้องการของผู้ใช้หรือลูกค้า

3. *การเขียนข้อกำหนดความต้องการ (Requirements Specification)* เป็นขั้นตอนที่สำคัญที่สุด หรือกล่าวได้ว่าเป็นหัวใจของระเบียบวิธีในการวิเคราะห์และออกแบบโครงสร้างของระบบ นักวิเคราะห์ระบบทำการสอบถามความต้องการจากผู้ใช้หรือลูกค้า และระบุเป็นข้อกำหนดที่มีรายละเอียดต่างๆ เกี่ยวกับซอฟต์แวร์

4. *การเขียนข้อกำหนดระบบเชิงตรรกะ (Logical System Specification)* แบ่งการทำงานออกเป็นขั้นตอนย่อย 2 ขั้นตอน คือ

4.1 *ทางเลือกของระบบเทคนิค (Technical System Options)* เป็นการเลือกเทคนิคต่างๆ โดยใช้ผลลัพธ์ที่ได้จากขั้นตอนการกำหนดความต้องการ เป็นข้อมูลที่ช่วยตัดสินใจในการเลือกเทคนิค เพื่อนำไปประยุกต์ใช้ในการออกแบบระบบ

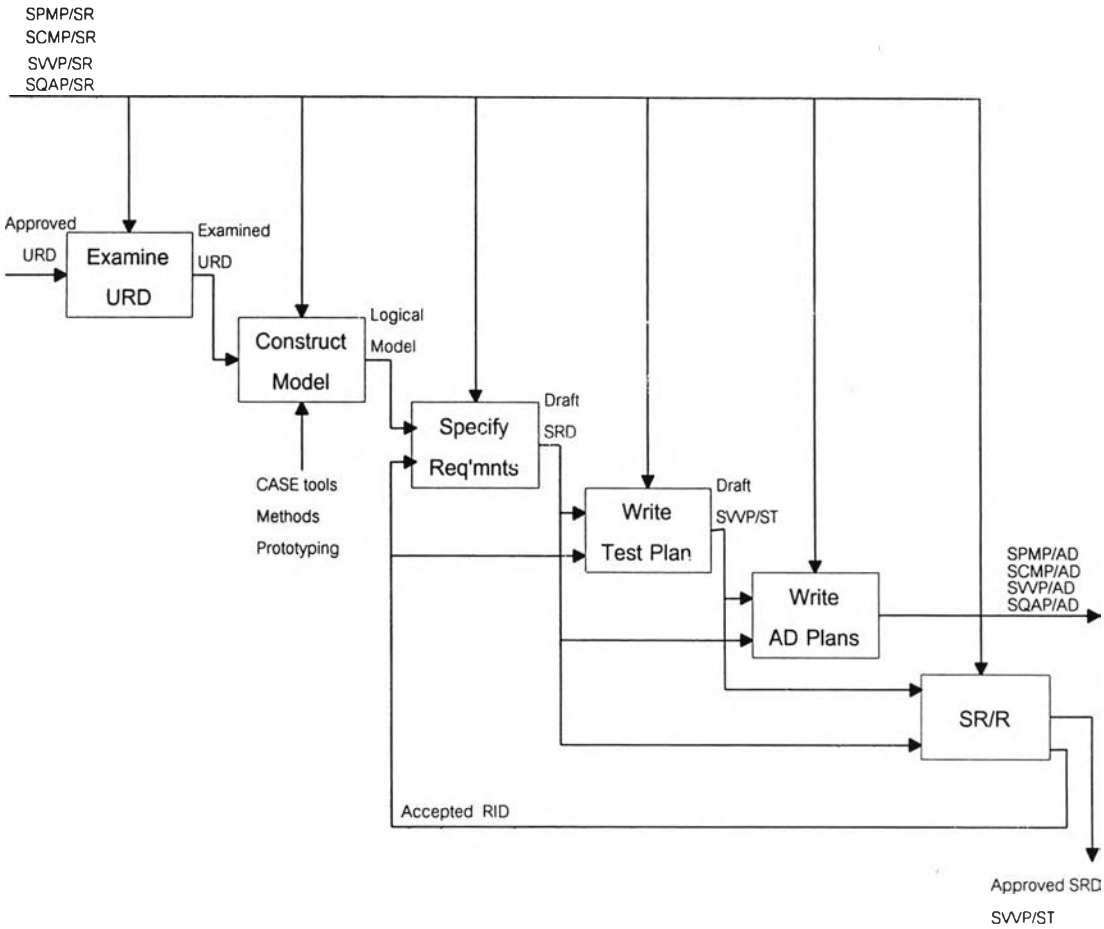
4.2 *การออกแบบเชิงตรรกะ (Logical Design)* เป็นกระบวนการออกแบบระบบโดยไม่มีการใช้เทคนิค โดยออกแบบในลักษณะที่เป็นแนวคิด เช่น การเขียนกระแสการไหลของข้อมูล

5. *การออกแบบเชิงกายภาพ (Physical Design)* เป็นการใช้เทคนิคต่างๆ จากขั้นตอนทางเลือกของระบบเทคนิคในการแปลง (transform) การออกแบบเชิงตรรกะเป็นการออกแบบเชิงกายภาพ เช่น มีการกำหนดชนิดของข้อมูลที่แน่นอน ข้อมูลที่อยู่ในตารางมีความสัมพันธ์กับตารางอื่นที่เกี่ยวข้องกัน เป็นต้น

2.2.2 ขั้นตอนการกำหนดความต้องการซอฟต์แวร์ (Software Requirement Definition Phase)

ขั้นตอนการกำหนดความต้องการซอฟต์แวร์ (Mazza, Fairclough, Melton et al., 1996) เรียกอีกอย่างหนึ่งว่า ขั้นตอนการวิเคราะห์ปัญหา ขั้นตอนนี้มีจุดประสงค์ เพื่อวิเคราะห์ความต้องการของผู้ใช้ในเอกสารความต้องการผู้ใช้ (User Requirements Document) และเพื่อให้ผู้พัฒนาซอฟต์แวร์ได้ผลิตเอกสารข้อ

กำหนดความต้องการซอฟต์แวร์ให้มีความสมบูรณ์ มีความสอดคล้องกัน และมีความถูกต้อง โดยบุคคลที่มีส่วนร่วมในขั้นตอนนี้นอกเหนือจากผู้พัฒนาซอฟต์แวร์แล้ว ยังรวมถึงผู้ใช้ ลูกค้า วิศวกรซอฟต์แวร์ นักวิเคราะห์ระบบ นักออกแบบระบบ ผู้จัดการโครงการ เป็นต้น บุคคลเหล่านี้จะต้องปรึกษาหารือร่วมกัน เพื่อเป็นการลดความเสี่ยง ความไม่สมบูรณ์ และความผิดพลาดที่อาจเกิดขึ้นในโครงการให้น้อยลง



รูปที่ 2.6 กระบวนการทำงานในขั้นตอนการกำหนดความต้องการซอฟต์แวร์

จากรูปที่ 2.6 เป็นการแสดงถึงกิจกรรมต่างๆ และการไหลของเอกสาร (document flow) ในขั้นตอนการกำหนดความต้องการซอฟต์แวร์ โดยเริ่มจากมีข้อมูลนำเข้าคือ เอกสารความต้องการผู้ใช้ แผนการจัดการโครงการซอฟต์แวร์ แผนการจัดการโครงแบบซอฟต์แวร์ แผนการทวนสอบและความสมเหตุสมผลซอฟต์แวร์ และแผนการรับประกันคุณภาพซอฟต์แวร์ ซึ่งข้อมูลนำเข้าเหล่านี้ จะนำไปใช้ในขั้นตอนการกำหนดความต้องการซอฟต์แวร์ 6 ขั้นตอน ได้แก่

1. *การสำรวจเอกสารความต้องการผู้ใช้ (Examine User Requirements Document)* ผู้พัฒนาซอฟต์แวร์นำเอกสารความต้องการผู้ใช้ที่ได้รับการยอมรับแล้วมาพิจารณา และยืนยันว่าเอกสารความต้องการผู้ใช้สามารถเข้าใจได้ง่าย

2. *การสร้างแบบจำลอง (Construct Model)* เป็นการนำเอกสารความต้องการผู้ใช้ไปสร้างแบบจำลอง โดยใช้ต้นแบบ เครื่องมือวิศวกรรมซอฟต์แวร์ใช้คอมพิวเตอร์ช่วย (Computer-Aided Software Engineering tools) หรือระเบียบวิธีต่างๆ ในการสร้างแบบจำลองเชิงตรรกะ (logical model) เพื่อเป็นการอธิบายถึงหน้าที่ของซอฟต์แวร์ แต่ไม่ต้องระบุว่าซอฟต์แวร์ทำงานได้อย่างไร ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ระบุถึงหน้าที่ ประสิทธิภาพ การต่อประสาน คุณภาพของซอฟต์แวร์ ความเชื่อถือได้ ความสามารถในการบำรุงรักษา ความปลอดภัย เป็นต้น

3. *การระบุความต้องการ (Specify Requirements)* เป็นการนำแบบจำลองเชิงตรรกะไปใช้ เพื่อระบุถึงความต้องการต่างๆ และผลิตเอกสารข้อกำหนดความต้องการซอฟต์แวร์ฉบับร่าง โดยจะต้องครอบคลุมความต้องการทั้งหมดในเอกสารความต้องการผู้ใช้

4. *การเขียนแผนการทดสอบระบบ (Test Plan)* หลังจากได้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ฉบับร่างแล้ว จะทำการเขียนแผนการทดสอบระบบ โดยในแผนนี้จะอธิบายถึงขอบเขต แนวความคิด และทรัพยากรต่างๆ เพื่อทดสอบซอฟต์แวร์ว่าครอบคลุมความต้องการทั้งหมดหรือไม่

5. *การเขียนแผนการออกแบบสถาปัตยกรรม (Architecture Design Plans)* เป็นการนำแผนการทวนสอบและความสมเหตุสมผลซอฟต์แวร์ และเอกสารข้อกำหนดความต้องการซอฟต์แวร์ฉบับร่างมาเขียนแผนการออกแบบสถาปัตยกรรม จะได้ข้อมูลนำออก คือ แผนการจัดการโครงการซอฟต์แวร์ แผนการจัดการโครงแบบซอฟต์แวร์ แผนการทวนสอบและความสมเหตุสมผลซอฟต์แวร์ และแผนการรับประกันคุณภาพซอฟต์แวร์ เพื่อนำไปใช้ในขั้นตอนการออกแบบสถาปัตยกรรม

6. *ตรวจดูความต้องการซอฟต์แวร์ (Software Requirements Review)* ผู้ใช้ ผู้พัฒนาซอฟต์แวร์ และบุคคลที่มีหน้าที่รับผิดชอบในการจัดการและรับประกันคุณภาพจะตรวจดูแผนการทวนสอบและความสมเหตุสมผลซอฟต์แวร์สำหรับขั้นตอนการออกแบบสถาปัตยกรรม และเอกสารข้อกำหนดความต้องการซอฟต์แวร์ฉบับร่างอีกครั้งหนึ่ง เพื่อให้เอกสารมีความชัดเจน สมบูรณ์ และมีข้อมูลเพียงพอที่จะใช้ในการออกแบบในขั้นตอนการออกแบบสถาปัตยกรรม ผลลัพธ์ของการตรวจดูนี้ จะได้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ได้รับการยอมรับแล้ว และได้แผนการทวนสอบและความสมเหตุสมผลซอฟต์แวร์สำหรับขั้นตอนการออกแบบสถาปัตยกรรม ซึ่งใช้สำหรับการทดสอบซอฟต์แวร์ในขั้นตอนการออกแบบโดยละเอียด (Detailed Design Phase) โดยปกติการตรวจดูเอกสารข้อกำหนดความต้องการซอฟต์แวร์ จะทำซ้ำหลายๆ ครั้งจนกว่าจะได้เอกสารที่สมบูรณ์และถูกต้อง ซึ่งรายละเอียดต่างๆ หลังจากทำการตรวจดูเอกสารแล้ว ไม่ว่าเอกสารนั้นจะได้รับการยอมรับหรือปฏิเสธจะเก็บผลของการตรวจดูในที่เก็บเอกสารที่ตรวจสอบแล้ว (Review Item Discrepancy) เมื่อเอกสารได้รับการยอมรับแล้ว ผู้พัฒนาซอฟต์แวร์จะนำไปเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ และนำไปเขียนแผนการทดสอบระบบต่อไป

2.2.3 การสร้างแบบสอบถาม

แบบสอบถาม (ประคอง กรรณสูต, 2524) ผู้วิจัยจะใช้แบบสอบถามเมื่อต้องการสำรวจเพื่อค้นหาแนวความคิดหรือข้อเท็จจริงที่ยังไม่ทราบ แบบสอบถามที่ดี ต้องมีรูปแบบที่น่าสนใจ ใช้ภาษาถูกต้องเข้าใจง่าย ไม่มีคำถามนอกเรื่อง เพราะจะทำให้ผู้ตอบไม่สนใจที่จะตอบ ผู้วิจัยจะต้องพยายามสร้างคำถามให้ตรงจุดมุ่งหมายในการวิจัย และครอบคลุมปัญหาที่ต้องการสำรวจ สร้างได้โดยอาศัยประสบการณ์จากความรู้เดิมและจากการศึกษางานวิจัยและเอกสารต่างๆ

2.2.3.1 ชนิดของคำถามที่ใช้ในแบบสอบถาม

1. *คำถามเปิด (Unstructured or Open-end Question)* เป็นคำถามที่ไม่ได้กำหนดคำตอบไว้ ผู้ตอบจะต้องเขียนตอบเองตามความคิดของตน ซึ่งทำให้ตอบยากและใช้เวลาในการตอบมาก ผู้ตอบจึงมักไม่เต็มใจที่จะตอบคำถามประเภทนี้ ยิ่งถ้าใช้ควบคู่กับข้อคำถามแบบอื่นแล้ว ผู้ตอบส่วนใหญ่มักจะเว้นข้ามไปโดยไม่ตอบข้อคำถามที่เป็นแบบปลายเปิด ลักษณะของคำตอบที่ได้จากคำถามแบบนี้ อาจเบนไปไม่ตรงเป้าหมาย และนำไปใช้วิเคราะห์และแปลความหมายได้ยาก ผู้วิจัยมักใช้ในการสำรวจเบื้องต้น เพื่อเป็นแนวทางในการสร้างข้อคำถามแบบปลายปิดต่อไป

2. *คำถามปิด (Structured or Close-ended Question)* เป็นคำถามที่กำหนดคำตอบหรือตัวเลือกไว้ ผู้ตอบไม่มีอิสระในการตอบ กล่าวคือ จะต้องเลือกตอบภายในตัวเลือกที่กำหนดให้ ทำให้คำตอบที่ได้รับค่อนข้างจำกัด อาจใช้วัดความคิดเห็นที่ลึกซึ้งนักไม่ได้ ข้อคำถามแบบนี้ สร้างยากและใช้เวลาในการสร้างมากกว่าแบบปลายเปิด แต่ผู้ตอบตอบได้ง่าย สะดวกและรวดเร็ว นอกจากนี้ข้อมูลที่ได้รับยังนำมาใช้วิเคราะห์และแปลผลได้ง่าย เนื่องจากผู้ตอบตอบในลักษณะหรือแบบฟอร์มเดียวกัน

3. *แบบให้จัดอันดับความสำคัญ (Priority scale)* เป็นคำถามที่กำหนดคำตอบไว้แล้วให้ผู้ตอบจัดลำดับความสำคัญจากตัวเลือกที่สำคัญมากที่สุดไปหาน้อยที่สุด เช่น ให้ตัวเลือกที่สำคัญที่สุดเป็นอันดับที่ 1 และตัวเลือกที่สำคัญรองลงไปเป็นอันดับ 2, 3, ... ตามลำดับ เป็นต้น

4. *แบบจัดอันดับคุณภาพหรือมาตราส่วนประเมินค่า (Rating scale)* ซึ่งผู้ตอบจะประเมินค่าน้ำหนักของตัวเลือกเป็น 5, 4, 3, 2, 1 หรือ 4, 3, 2, 1 หรือตามแนวคิดที่ผู้สร้างแบบประเมินค่าจะกำหนดไว้

2.2.3.2 ประเภทของแบบสอบถาม

1. *แบบสอบถามแบบมาตราส่วนประมาณค่าของลิเคอร์ท (Likert Scale)* เป็นวิธีการที่ลิเคอร์ทเป็นผู้สร้างขึ้น ซึ่งเป็นวิธีการประมาณค่ารวม โดยมีลักษณะที่ให้ผู้ตอบเป็นผู้ประเมินพฤติกรรม ความคิดเห็น หรือความรู้สึกต่อสิ่งต่างๆ ตามข้อความที่กำหนดให้ เช่น เห็นด้วยอย่างยิ่ง เห็นด้วย ไม่แน่ใจ ไม่เห็นด้วย และไม่เห็นด้วยเลย เป็นต้น ซึ่งเป็นแบบสอบถามชนิดแบบจัดอันดับคุณภาพ หรือมาตราส่วนประเมินค่า แบบสอบถามนี้เป็นแบบที่นิยมใช้กันมาก เพราะสร้างง่ายและใช้ได้กว้างขวาง กล่าวคือ วัดได้ทั้งในด้านทัศนคติและความคิดเห็น ตลอดปัญหาหรือข้อเท็จจริงต่างๆ

2. *แบบสอบถามวัดทัศนคติโดยใช้ความหมายทางภาษา (Semantic Differential Scale)* เป็นแบบสอบถามที่ออสกูด (Osgood) เป็นผู้สร้างขึ้นเพื่อใช้วัดทัศนคติโดยเฉพาะ โดยใช้คำหรือข้อความสั้นๆ มาใช้อธิบายทัศนคติ เช่น ชอบ-ไม่ชอบ น่าสนใจ-น่าเบื่อ เป็นต้น

3. *แบบตรวจสอบรายการ (Checklist)* เป็นแบบสำรวจที่อาจให้กลุ่มตัวอย่างประเมินด้วยตนเอง หรืออาจให้ผู้อื่นประเมิน โดยระบุรายการต่างๆ ซึ่งอาจเป็นพฤติกรรม ข้อเท็จจริง หรือรายการสิ่งของ เพื่อให้ผู้ตอบตัดสินใจตอบรับหรือปฏิเสธ ถ้าเป็นกรณีที่ถามคิดเห็น พฤติกรรม หรือปัญหาต่างๆ ลักษณะของแบบตรวจสอบรายการก็จะคล้ายกับแบบมาตราส่วนประมาณค่า แต่มีตัวเลือกเพียง 2 ระดับเท่านั้น เช่น ถูกหรือผิด เห็นด้วยหรือไม่เห็นด้วย เป็นต้น

4. *แบบสอบถามแบบยกสถานการณ์ (Situational Questionnaire)* เป็นแบบสอบถามที่ใช้วิธีการกำหนดสถานการณ์ต่างๆ ขึ้น แล้วให้ผู้ตอบตอบคำถามเกี่ยวกับสถานการณ์นั้นๆ โดยมีจุดมุ่งหมายเพื่อศึกษาลักษณะการตัดสินใจ การโต้ตอบ หรือการกระทำ ตลอดจนความคิดเห็นและความรู้สึกของบุคคลเมื่ออยู่ในสถานการณ์หรือสถานภาพต่างๆ โดยผู้ตอบจะต้องอ่านข้อความในสถานการณ์แล้วพิจารณาตนเองว่า ถ้าประสบเหตุการณ์เช่นนั้นจริงๆ แล้ว จะเกิดความรู้สึกหรือจะเลือกปฏิบัติตามคำตอบในตัวเลือกใด ซึ่งจะต้องเลือกตอบเพียงตัวเลือกเดียวเท่านั้น

5. *แบบสำรวจหรือแบบสัมภาษณ์ (Inventory or Interview Schedule)* เป็นแบบสอบถามประเภทหนึ่ง ซึ่งประกอบด้วยข้อคำถามหลายๆ ชนิดปะปนกัน โดยมีจุดมุ่งหมายเพื่อศึกษาข้อความจริงจากกลุ่มตัวอย่าง ซึ่งอาจจะเป็นการศึกษาความคิดเห็น หรือสภาพต่างๆ ของกลุ่มตัวอย่าง แบบสำรวจนี้นิยมใช้ประกอบการสัมภาษณ์แต่ก็ใช้เป็นแบบสอบถามให้ผู้ตอบเขียนตอบเอง ในกรณีที่กลุ่มตัวอย่างเป็นผู้มีการศึกษา

2.2.4 ทฤษฎีทางสถิติที่ใช้ในการวิเคราะห์ข้อมูล

2.2.4.1 การตรวจสอบความสมเหตุสมผล (Validity)

การตรวจสอบความสมเหตุสมผลของเครื่องมือที่นำมาใช้ในการวิจัย หมายถึง ขนาดที่บอกได้ว่าแบบสอบถามสามารถวัดสิ่งที่ต้องการวัดได้ตามจุดมุ่งหมาย (พรศรี ศรีอัษฎาพร และ ยุวดี วัฒนานนท์, 2529) ซึ่งแบ่งออกเป็นประเภทใหญ่ๆ ได้ 3 ประเภท คือ

1. *ความสมเหตุสมผลเชิงเนื้อหา (Content Validity)* หมายความว่า แบบสอบถามที่นำมาใช้วัด ได้สุ่มเนื้อหา องค์ประกอบ หรือสภาพการณ์ต่างๆ ที่ต้องการศึกษาได้ครบถ้วนเพียงไร ซึ่งในการพิจารณาเนื้อหาของเครื่องมือที่นำมาใช้ในการวิจัย ว่ามีเนื้อหาเหมาะสมหรือไม่ มักให้ผู้เชี่ยวชาญในสาขาที่เราต้องการศึกษาเป็นผู้ตรวจสอบ การตรวจสอบความสมเหตุสมผลเชิงเนื้อหาของเครื่องมือต้องอาศัยการพิจารณาตัดสินของผู้เชี่ยวชาญ เพื่อพิจารณาคำถามในแบบสอบถามที่นำมาใช้วัดนั้นว่าแต่ละคำถามเป็นตัวแทนในสิ่งที่เราต้องการจะวัดหรือต้องการจะศึกษาหรือไม่

2. *ความสมเหตุสมผลเชิงโครงสร้าง (Construct Validity)* หมายถึง ขอบเขตหรือลักษณะตามทฤษฎีที่แบบสอบถามนั้นวัดได้ ตัวอย่างของความหมายตามทฤษฎีนี้ ได้แก่ เชาวน์ปัญญา ความเข้าใจทางกลไก ความชำนาญด้านถ้อยคำ ความกระวนกระวายใจ เป็นต้น

3. *ความสมเหตุสมผลเชิงสัมพันธ์กับเกณฑ์ (Criterion Related Validity)* เป็นการศึกษาโดยการเปรียบเทียบแบบสอบถามกับตัวแปรภายนอก 1 ตัว หรือมากกว่า 1 ตัว หรือเปรียบเทียบกับเกณฑ์ที่ประจักษ์หรือเชื่อว่าได้วัดในสิ่งที่เราต้องการจะศึกษา เช่น แบบทดสอบความถนัด ทำนายความสำเร็จในอนาคต แบบทดสอบสัมฤทธิผล ทำนายความสามารถและความสำเร็จทั้งในปัจจุบันและอนาคต แบบทดสอบเชาวน์ปัญญา ทำนายความสามารถในการเรียนรู้วิธีแก้ปัญหาทั้งในปัจจุบันและอนาคต เป็นต้น

2.2.4.2 การตรวจสอบความเชื่อถือได้

การตรวจสอบความเชื่อถือได้ หมายถึง ความคงที่หรือความคงเส้นคงวาที่ได้จากการวัดสิ่งที่ต้องการวัด หรือความคงที่ของคะแนนที่ได้จากคนกลุ่มเดียวกันด้วยแบบสอบถามชุดเดียวกันในเวลาที่ต่างกัน (ประคอง กรรณสูต, 2524) แบ่งออกเป็น 4 วิธี ได้แก่

1. *การคำนวณหาค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson Product Moment Correlation Coefficient)* วิธีนี้เหมาะที่จะใช้กับแบบทดสอบในห้องเรียน เพราะเป็นวิธีที่สะดวกแต่ต้องแน่ใจว่าแบบทดสอบจะต้องมีข้อสอบที่มีระดับความยากใกล้เคียงกัน เทคนิคนี้ไม่เหมาะที่จะใช้หาความเชื่อถือได้ของแบบทดสอบที่ต้องอาศัยความเร็ว และแบบสอบถาม ซึ่งใช้สูตร

$$r_{XY} = \frac{N\sum XY - \sum X \sum Y}{\sqrt{\left[N\sum X^2 - (\sum X)^2\right]\left[N\sum Y^2 - (\sum Y)^2\right]}}$$

เมื่อ X,Y คือ คะแนนของตัวแปรต่อเนื่อง 2 ชุด ซึ่งมีข้อตกลงว่า X และ Y ต้องสัมพันธ์กันเป็นเส้นตรง

N คือ จำนวนคนกลุ่มเดียวกันที่ได้รับคะแนน X และ Y โดยขนาดของ N ต้องใหญ่พอสมควร คือประมาณตั้งแต่ 60 คนขึ้นไป

2. *การคำนวณโดยใช้สูตรคูเดอร์-ริชาร์ดสัน (Kuder-Richardson Reliability)* เป็นวิธีหนึ่งที่เลี่ยงปัญหาการที่ต้องตัดสินใจหาวิธีแบ่งแบบสอบถามออกเป็นครึ่ง และใช้ได้กับแบบสอบถามที่ตัดสินได้ว่าถูกหรือผิดเท่านั้น ซึ่งใช้สูตร

$$r_{xx} = \frac{n}{n-1}\left[1 - \frac{\sum pq}{S_x^{\,2}}\right]$$

เมื่อ r_{xx} คือ สัมประสิทธิ์แห่งความเชื่อถือได้

n คือ จำนวนข้อในแบบสอบถาม

p คือ สัดส่วนของคนที่ตอบแบบสอบถามได้ถูกต้อง

q คือ สัดส่วนของคนที่ตอบแต่ละข้อผิด

pq คือ ความแปรปรวนของคะแนนแต่ละข้อ

S_x^2 คือ ความแปรปรวนของคะแนนของผู้ตอบแบบสอบถามทั้งหมด

3. *การคำนวณโดยใช้สูตรสัมประสิทธิ์อัลฟา (α coefficient)* โดยวิธีนี้เป็นการวัดความสอดคล้องภายใน ครอนบาช (Cronbach) เป็นผู้พัฒนาสูตรนี้เมื่อปี ค.ศ. 1951 ใช้กับแบบสอบถามที่ไม่ได้เป็นชนิดให้คะแนนถูกเป็น 1 และผิดเป็น 0 แต่ใช้กับข้อสอบแบบเรียงความ แบบวัดทัศนคติ แบบประเมินผลสมรรถภาพในด้านต่างๆ ที่ต้องประเมินผลตามสเกลและแบบสอบถามชนิดประเมินค่า (rating scale) เป็นต้น โดยใช้สูตร

$$\alpha = \frac{n}{n-1}\left(1 - \frac{\sum s_i^2}{s_x^2}\right)$$

เมื่อ n คือ จำนวนข้อของแบบสอบถาม

S_i^2 คือ ความแปรปรวนของคะแนนแต่ละข้อ

S_x^2 คือ ความแปรปรวนของคะแนนของผู้ตอบแบบสอบถามทั้งหมด

4. *การวิเคราะห์ความแปรปรวนของฮอยท์ (Hoyt's Analysis of Variance)* ซึ่ง ซี เจ ฮอยท์ เป็นผู้พัฒนาสูตรนี้ เมื่อปี ค.ศ. 1941 เหมาะสำหรับใช้กับข้อสอบอัตนัยที่มีการตรวจให้คะแนนต่างๆ กันในแต่ละข้อ นอกจากนี้ยังเหมาะสำหรับแบบทดสอบ แบบสอบถาม แบบสำรวจต่างๆ การวัดทัศนคติ และแบบสัมภาษณ์ โดยใช้สูตร

$$r_{tt} = 1 - \frac{MS_E}{MS_P}$$

เมื่อ r_{tt} คือ ค่าความเชื่อถือได้

MS_E คือ คะแนนความแปรปรวนของความคลาดเคลื่อน (Error)

MS_P คือ คะแนนความแปรปรวนระหว่างคน (between people)

ถ้าตารางข้อมูลเป็นดังนี้

ข้อ / คน	1	2	.....	k	รวม
1	X_{11}	X_{12}	.....	X_{1k}	P_1
2	X_{21}	X_{22}	.....	X_{2k}	P_2
.					.
.					.
.					.
n	X_{n1}	X_{n2}	.....	X_{nk}	P_n
รวม	T_1	T_2	.....	T_k	T

ขั้นตอนในการวิเคราะห์ความแปรปรวนของฮอยท์ จะทำดังนี้

1. หาค่าผลบวกของคะแนนกำลังสอง (Sum of square) จากสูตร

$$SS_{tt} = \sum_{i=1}^{n}\sum_{j=1}^{k} X_{ij}^2 - \frac{T^2}{nk}$$

$$SS_k = \frac{\sum T_i^2}{n} - \frac{T^2}{nk}$$

$$SS_p = \frac{\sum P_i^2}{k} - \frac{T^2}{nk}$$

$$SS_E = SS_{tt} - SS_k - SS_p$$

เมื่อ SS_{tt} คือ ผลบวกของคะแนนกำลังสองของข้อมูลทั้งหมด

SS_k คือ ผลบวกของคะแนนกำลังสองระหว่างข้อ

SS_p คือ ผลบวกของคะแนนกำลังสองระหว่างคน

SS_E คือ ผลบวกของคะแนนกำลังสองความคลาดเคลื่อน

n คือ จำนวนคน

k คือ จำนวนข้อ

X_{ij} คือ คะแนนของแต่ละคนในแต่ละข้อ

T^2 คือ กำลังสองของผลรวมของคะแนนทั้งหมด

T_i^2 คือ กำลังสองของผลรวมในแต่ละข้อ

P_i^2 คือ กำลังสองของผลรวมของแต่ละคน

2. หาค่าคะแนนความแปรปรวน (MS) จากสูตร

$$MS_P = \frac{SS_P}{df_P} \quad ; \quad df_P = \text{n-1}$$

$$MS_E = \frac{SS_E}{df_E} \quad ; \quad df_E = \text{(n-1)(k-1)}$$

เมื่อ MS_E คือ คะแนนความแปรปรวนของความคลาดเคลื่อน

MS_P คือ คะแนนความแปรปรวนระหว่างคน

df_P คือ ตัวแปรอิสระระหว่างคน

df_E คือ ตัวแปรอิสระของความคลาดเคลื่อน

3. หาค่าความเชื่อถือได้ จากสูตร

$$r_{tt} = 1 - \frac{MS_E}{MS_P}$$

เมื่อ r_{tt} คือ ค่าความเชื่อถือได้

MS_E คือ คะแนนความแปรปรวนของความคลาดเคลื่อน

MS_P คือ คะแนนความแปรปรวนระหว่างคน

ผลที่ได้จากการวิเคราะห์ความเชื่อถือได้ของฮอยท์

ค่าความเชื่อถือได้ของแบบสอบถาม ซึ่งจะมีค่าอยู่ในช่วง 0-1 ถ้าค่าความเชื่อถือได้มีค่าต่ำกว่า 0.5 แสดงว่าแบบสอบถามนี้เชื่อถือไม่ได้ ถ้าค่าความเชื่อถือได้มีค่าสูงกว่าหรือเท่ากับ 0.5 ขึ้นไป หรือมีค่าเข้าใกล้ 1 แสดงว่าแบบสอบถามนี้มีความเชื่อถือได้

2.2.4.3 **ค่าเฉลี่ย** (Mean)

ค่าเฉลี่ย (ประคอง กรรณสูต, 2524) คือ จุดสมดุลย์ของคะแนนในข้อมูลชุดหนึ่งๆ สูตรที่ใช้คำนวณ คือ

$$\text{ค่าเฉลี่ย} = \frac{\sum fX}{N}$$

เมื่อ f คือ ความถี่ของคะแนน

X คือ คะแนน

N คือ จำนวนตัวอย่าง

ค่าเฉลี่ยที่ได้ จะเป็นตัวแทนที่ดีของข้อมูลได้ก็ต่อเมื่อคะแนนในกลุ่มมีค่าใกล้เคียงกัน หรือมีการแจกแจงเป็นโค้งปกติเท่านั้น ซึ่งอาจพิจารณาจากค่าส่วนเบี่ยงเบนมาตรฐาน ถ้าค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1 หรือเข้าใกล้ 1 แสดงว่า การแจกแจงของข้อมูลชุดนั้นมีลักษณะเป็นโค้งปกติ

2.2.4.4 **ค่าส่วนเบี่ยงเบนมาตรฐาน** (Standard Deviation)

ค่าส่วนเบี่ยงเบนมาตรฐาน (ประคอง กรรณสูต, 2524) ค่าที่คำนวณได้ ใช้ในการประกอบการพิจารณาว่าลักษณะข้อมูลเป็นอย่างไร ค่าแนวโน้มเข้าสู่ส่วนกลางที่เสนอไว้เหมาะสมกับลักษณะของข้อมูลหรือไม่ สูตรในการคำนวณค่าส่วนเบี่ยงเบนมาตรฐาน คือ

$$S.D. = \sqrt{\frac{\sum fX^2}{N} - \left(\frac{\sum fX}{N}\right)^2}$$

เมื่อ f คือ ความถี่ของคะแนน

X คือ คะแนน

N คือ จำนวนตัวอย่าง

ค่าส่วนเบี่ยงเบนมาตรฐานสามารถแสดงลักษณะของข้อมูลได้ดังนี้

1. เมื่อค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0 แสดงว่า ข้อมูลชุดนั้นไม่มีการกระจาย นั่นคือ คะแนนทุกจำนวนในข้อมูลชุดนั้นมีค่าเท่ากัน ถ้าคะแนนในข้อมูลแทนความคิดเห็นของคนกลุ่มหนึ่ง ก็หมายความว่า คนกลุ่มนั้นมีความคิดเห็นตรงกัน

2. เมื่อค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1 แสดงว่าการแจกแจงของข้อมูลชุดนั้นมีลักษณะเป็นโค้งปกติ หรือมีลักษณะสมมาตร (symmetry) ทั้งสองข้างของคะแนน กรณีนี้จะใช้ค่าเฉลี่ยของข้อมูลแสดงขนาดของข้อมูลโดยเฉลี่ยได้ดีที่สุด

3. เมื่อค่าส่วนเบี่ยงเบนมาตรฐานมากกว่าค่าเฉลี่ยของข้อมูลชุดเดียวกันนั้น ก็ไม่ควรใช้ค่าเฉลี่ยในการเสนอขนาดเฉลี่ยของข้อมูล สมควรจะต้องพิจารณาใช้มัธยฐานหรือฐานนิยมตามความเหมาะสมกับเรื่องที่จะรายงานต่อไป

4. เมื่อค่าส่วนเบี่ยงเบนมาตรฐานเข้าใกล้ 0 ข้อมูลชุดนั้นก็มีการกระจายน้อยลงทุกที ถ้าคะแนนในข้อมูลเป็นน้ำหนักความคิดเห็น คนกลุ่มนั้นก็มีความคิดเห็นใกล้เคียงกันกว่าข้อมูลชุดที่มีค่าส่วนเบี่ยงเบนมาตรฐานมากกว่า

5. เมื่อค่าส่วนเบี่ยงเบนมาตรฐานเข้าใกล้ 1 หรือใกล้เคียงกัน การแจกแจงของข้อมูลชุดนั้นก็ใกล้เคียงกับโค้งปกติกว่าข้อมูลชุดที่มีการแจกแจงห่างจาก 1 มากกว่า

2.2.4.5 การทดสอบกลุ่มตัวอย่างกลุ่มเดียวของโกลโมโกรอฟ-สเมอนอฟ (The Kolmogorov-Smirnov One Sample Test)

การทดสอบกลุ่มตัวอย่างกลุ่มเดียวของโกลโมโกรอฟ-สเมอนอฟ (วิเชียร เกตุสิงห์, 2535) เป็นวิธีทดสอบเพื่อดูว่าการแจกแจงของข้อมูลที่สังเกตได้แตกต่างจากการแจกแจงตามทฤษฎี (หรือที่หวังไว้ตามสมมุติฐาน) หรือไม่ ซึ่งมีขั้นตอนของการทดสอบ ดังนี้

ข้อตกลงเบื้องต้น 1. ไม่มีข้อตกลงเกี่ยวกับการกระจายของประชากร

2. ลักษณะของข้อมูลเป็นความถี่ที่ได้จากกลุ่มตัวอย่างกลุ่มเดียว

สมมุติฐาน ไม่มีความแตกต่างระหว่างจำนวนที่สังเกตได้กับจำนวนที่คาดหวัง นั่นคือ

H_0 : $fO_i = fE_i$

H_1 : H_0 ไม่เป็นจริง (ไม่เป็นไปตาม H_0)

การทดสอบ ใช้สูตร $D = \text{maximum} \mid F_O - F_E \mid$

เมื่อ F_O เป็นความถี่สะสมของข้อมูลที่สังเกตได้

F_E เป็นความถี่ของข้อมูลที่คาดหวัง

D ก็คือ ผลต่างของ F_O กับ F_E ที่มีค่าสูงสุด (ไม่คิดเครื่องหมาย)

การกระจายทางสถิติ เป็นไปตามตารางแสดงค่าของ D

ลักษณะการทดสอบ เป็นแบบสองหาง

การตัดสินใจ ไม่ยอมรับสมมุติฐาน H_0 ถ้าค่า D ที่คำนวณได้มีค่ามากกว่าค่า D ที่คำนวณได้จากสูตรในตารางค่าวิกฤตของ D (วิเชียร เกตุสิงห์, 2535) นั่นคือ มีความแตกต่างระหว่างจำนวนที่สังเกตได้กับจำนวนที่คาดหวังไว้

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 งานวิจัยเกี่ยวกับการประเมินผลมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้ง 4 มาตรฐาน

Giakoumakis และ Xylomenos (1996) ทำการประเมินมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้ง 4 มาตรฐาน ได้แก่ มาตรฐาน IEEE Std 830-1993 มาตรฐาน ESA PSS-05-0 มาตรฐาน NASA-DID-P200 และมาตรฐาน DoD-STD-498 โดยมีการสร้างเกณฑ์และมีการประเมินผลมาตรฐาน 8 ข้อ คือ

1. *ความเป็นอิสระ (Independence)*

พิจารณาจากระเบียบวิธีที่ใช้ในการพัฒนาซอฟต์แวร์ ผู้พัฒนาซอฟต์แวร์สามารถใช้ระเบียบวิธีที่เป็นอิสระได้ เพราะเป็นการหลีกเลี่ยงความไม่แน่นอนของมาตรฐาน ซึ่งการประเมินผลแบ่งเป็น 3 ระดับ คือ

1.1 *ระดับสูง* มาตรฐานมีรูปแบบวัฏจักรของการพัฒนาซอฟต์แวร์หลายรูปแบบที่ใช้ในการออกแบบระบบ และประยุกต์ใช้กับระเบียบวิธีของการพัฒนาซอฟต์แวร์ที่แตกต่างกันได้

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.2 *ระดับกลาง* มาตรฐานบางมาตรฐานไม่สามารถประยุกต์ใช้กับวัฏจักรของการพัฒนาซอฟต์แวร์หลายรูปแบบ

1.3 *ระดับต่ำ* มาตรฐานบางมาตรฐานได้รับการพิจารณาว่าไม่เหมาะสมที่จะนำมาใช้ในการพัฒนาซอฟต์แวร์

2. *การรวมเป็นหนึ่งเดียว (Integration)*

พิจารณาจากความสมบูรณ์ของมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ว่าครอบคลุมขั้นตอนของการพัฒนาซอฟต์แวร์ทั้งหมดหรือไม่ ซึ่งการประเมินผล แบ่งเป็น 3 ระดับ คือ

2.1 *ระดับสูง* มาตรฐานมีความสมบูรณ์ สามารถใช้มาตรฐานเพื่อเปลี่ยนแปลงแก้ไขซอฟต์แวร์ได้ง่าย

2.2 *ระดับกลาง* มีบางส่วนของมาตรฐานที่ไม่สมบูรณ์ ไม่ครอบคลุมขั้นตอนบางขั้นตอนของการพัฒนาซอฟต์แวร์

2.3 *ระดับต่ำ* มาตรฐานมีความไม่สมบูรณ์ ไม่ครอบคลุมขั้นตอนของการพัฒนาซอฟต์แวร์

3. *ความถูกต้อง (Precision)*

พิจารณาจากความกะทัดรัดและความชัดเจนของเนื้อหา ต้องมีความเข้าใจที่ตรงกัน หรือไม่มีความคลุมเครือเกิดขึ้น พร้อมทั้งมีความยืดหยุ่น ซึ่งการประเมินผล แบ่งเป็น 3 ระดับ คือ

3.1 *ระดับสูง* รายละเอียดในความต้องการมีความสมบูรณ์ ชัดเจน และมีตัวอย่างที่ละเอียดในแต่ละหัวข้อ

3.2 *ระดับกลาง* มาตรฐานมีความชัดเจน แต่ยังคงมีค่าใช้จ่ายและความเสี่ยงเล็กน้อย

3.3 *ระดับต่ำ* มาตรฐานมีความคลุมเครือพร้อมทั้งมีค่าใช้จ่ายและความเสี่ยงมาก

4. *คุณลักษณะโดยทั่วไป (Generality)*

พิจารณาจากความเหมาะสมของมาตรฐานที่จะใช้กับโครงการต่างๆ ในองค์กร ซึ่งการประเมินผล แบ่งเป็น 2 ลักษณะ คือ

4.1 *โครงการขนาดใหญ่*

4.2 *โครงการขนาดเล็ก*

5. *การจัดระเบียบของเอกสาร (Organization)*

พิจารณาจากเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ต้องมีความยืดหยุ่น สามารถเข้าใจได้ง่าย และใช้งานง่าย ซึ่งการประเมินผล แบ่งเป็น 3 ระดับ คือ

5.1 ระดับสูง เอกสารข้อกำหนดความต้องการซอฟต์แวร์มีความยืดหยุ่นสูง

5.2 ระดับกลาง เอกสารข้อกำหนดความต้องการซอฟต์แวร์ไม่มีความยืดหยุ่น มีโครงสร้างเป็นแบบคงที่

5.3 *ระดับต่ำ* เอกสารข้อกำหนดความต้องการซอฟต์แวร์ยากต่อการเข้าใจ ทำให้เสียค่าใช้จ่ายสูงในการบำรุงรักษาระบบ

6. *ความสมบูรณ์ของเนื้อหา (Content completeness)*

พิจารณาจากความต้องการทุกประเภทต้องอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้งหมด จุดประสงค์เพื่อใช้เป็นแนวทางในการวิเคราะห์ระบบและเพื่อตรวจสอบซอฟต์แวร์ ซึ่งการประเมินผล แบ่งเป็น 3 ระดับ คือ

6.1 *ระดับสูง* มาตรฐานต้องครอบคลุมความต้องการทั้งหมด รายละเอียดในมาตรฐานไม่จำเป็นต้องมีรายละเอียดมาก แต่ควรที่จะใช้ง่าย และไม่มีความคลุมเครือ

6.2 *ระดับกลาง* มาตรฐานพยายามที่จะครอบคลุมความต้องการทั้งหมด เพื่อไม่ให้มีความไม่สอดคล้องกัน และความคลุมเครือเกิดขึ้นในเอกสาร

6.3 *ระดับต่ำ* ความต้องการบางอย่างมีความสมบูรณ์ แต่ไม่ครอบคลุมความต้องการทั้งหมดในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

7. *ความสมบูรณ์ของมุมมองภายนอก (View completeness)*

พิจารณาจากความพึงพอใจของผู้ใช้ นักวิเคราะห์ระบบ นักออกแบบระบบ นักทดสอบระบบ และนักบำรุงรักษาระบบ

7.1 *ผู้ใช้* ความต้องการต่างๆ มาจากผู้ใช้ เอกสารข้อกำหนดความต้องการซอฟต์แวร์ต้องได้รับการตรวจสอบว่าถูกต้องและเข้าใจง่ายหรือไม่

7.2 *นักวิเคราะห์ระบบ* ทำการรวบรวมความต้องการจากผู้ใช้หรือลูกค้า และเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ให้มีความสมบูรณ์ เพื่อใช้เป็นแนวทางในการวิเคราะห์ระบบ

7.3 *นักออกแบบระบบ* ใช้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ในการพัฒนาซอฟต์แวร์

7.4 *นักทดสอบระบบ* ทำการตรวจสอบเอกสารข้อกำหนดความต้องการซอฟต์แวร์ว่าตรงกับความต้องการของผู้ใช้หรือลูกค้าหรือไม่

7.5 *นักบำรุงรักษาระบบ* ใช้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ เพื่อเปลี่ยนแปลงแก้ไขซอฟต์แวร์ได้ง่าย

8. *การเปลี่ยนแปลงแก้ไขได้ (Modifiability)*

พิจารณาจากวิธีการที่จะแก้ไขเอกสารข้อกำหนดความต้องการซอฟต์แวร์โดยไม่ส่งผลกระทบต่อคุณภาพของเอกสาร ซึ่งการประเมินผล แบ่งเป็น 2 ลักษณะ คือ

8.1 *เพียงพอ (adequate)* ผู้พัฒนาซอฟต์แวร์สามารถกระจายความต้องการออกเป็นโมดูล ตลอดจนสามารถมีการติดตาม (trace) ย้อนหลัง และไปข้างหน้าได้

8.2 *ไม่เพียงพอ (inadequate)* การเปลี่ยนแปลงแก้ไขส่งผลกระทบต่อเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ผู้พัฒนาซอฟต์แวร์ไม่สามารถใช้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ในการติดตามซอฟต์แวร์ได้

งานวิจัยของ Giakoumakis และ Xylomenos (1996) ได้สรุปผลการประเมินผลมาตรฐานทั้งสี่ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 การประเมินผลของมาตรฐาน IEEE Std 830-1993 ESA PSS-05-0 NASA-DID-P200 และ DoD-STD-498 (Giakoumakis and Xylomenos, 1996)

เกณฑ์	การประเมินผล			
	IEEE Std 830-1993	ESA PSS-05-0	NASA-DID-P200	DoD-STD-498
ความเป็นอิสระ	ปานกลาง	ปานกลาง	สูง	สูง
การรวมเป็นหนึ่งเดียว	ปานกลาง	ปานกลาง	สูง	สูง
ความถูกต้อง	สูง	ต่ำ	สูง	สูง
คุณลักษณะโดยทั่วไป	ไม่เหมาะกับโครงการขนาดใหญ่	ไม่เหมาะกับโครงการขนาดใหญ่	ไม่เหมาะกับโครงการขนาดเล็ก	ไม่เหมาะกับโครงการขนาดเล็ก
การจัดระเบียบของเอกสาร	สูง	ต่ำ	ต่ำ	ปานกลางถึงสูง
ความสมบูรณ์ของเนื้อหา	สูง	ปานกลาง	สูง	สูง
ความสมบูรณ์ของมุมมองภายนอก	เหมาะกับผู้ใช้ทุกคน	เหมาะกับนักทดสอบระบบเท่านั้น	เหมาะกับนักออกแบบระบบและนักทดสอบระบบ	เหมาะกับผู้ใช้ทุกคน
การเปลี่ยนแปลงแก้ไขได้	เพียงพอ	ไม่เพียงพอ	ไม่เพียงพอ	เพียงพอ

2.3.2 งานวิจัยเกี่ยวกับการเขียนข้อกำหนดของความต้องการด้วยภาษาธรรมชาติอย่างมีประสิทธิภาพ

(Wilson, 1999) ได้ทำการวิเคราะห์การเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ให้มีประสิทธิภาพ โดยองค์การนาซาได้ทำการวิเคราะห์เอกสารข้อกำหนดความต้องการซอฟต์แวร์จำนวน 40 ฉบับ

เอกสารข้อกำหนดความต้องการซอฟต์แวร์ เป็นเอกสารที่สำคัญที่ใช้เป็นหลักฐานในการส่งมอบซอฟต์แวร์ให้กับลูกค้า เพื่อตรวจสอบว่าซอฟต์แวร์สามารถทำงานตรงตามความต้องการของลูกค้าหรือไม่ ซึ่งศูนย์เทคโนโลยีรับรองซอฟต์แวร์ (Software Assurance Technology Center) แห่งนาซา ได้พบข้อบกพร่องดังนี้

2.3.2.1 การจัดระเบียบของเอกสาร

ปัญหาในการจัดระเบียบของเอกสาร คือ มีความซ้ำซ้อนเกิดขึ้นในเอกสาร ทำให้บำรุงรักษาเอกสารได้ยาก ดังนั้น ควรมีการเรียบเรียงหัวข้อหรือเนื้อหาในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่มีความสัมพันธ์กัน โดยจัดระเบียบให้อยู่รวมกัน เช่น

มาตรฐาน IEEE Std 830-1993 มีเนื้อหาในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ดังนี้

1. ความเชื่อถือได้ (Reliability)
2. สภาพพร้อมใช้งาน (Availability)
3. ความมั่นคง (Security)
4. ความสามารถในการบำรุงรักษา (Maintainability)
5. การใช้ได้หลายระบบ (Portability)

จากเนื้อหาข้างต้น ควรจัดเนื้อหาให้อยู่ในกลุ่มของลักษณะประจำของซอฟต์แวร์ (Software attributes)

2.3.2.2 โครงสร้างของการเขียนข้อกำหนด

ในการเขียนข้อกำหนดนั้น ผู้พัฒนาซอฟต์แวร์เขียนข้อกำหนดโดยมีโครงสร้างของประโยคที่แตกต่างกัน ถ้าข้อกำหนดมีโครงสร้างที่ไม่ดี อาจทำให้เกิดความสับสนและมีความเข้าใจที่ไม่ตรงกัน ดังนั้น ถ้าข้อกำหนดใดมีเครื่องหมายวรรคตอนหรือคำเชื่อมเกิน 3 แห่ง ควรทำการปรับโครงสร้างของรูปแบบประโยคใหม่ เพื่อให้ง่ายในการอ่าน

ตัวอย่างของข้อกำหนดที่มีคำเชื่อมเกิน 3 แห่ง คือ

บริษัทผู้เสนอราคาจะต้องเป็นนิติบุคคลที่จดทะเบียนในประเทศไทยและมีบุคลากรซึ่งมีความรู้และประสบการณ์ในงานด้านกิจการท่าอากาศยาน และสนับสนุนกิจการท่าอากาศยาน และประกอบธุรกิจเกี่ยวกับการขายและ/หรือการให้เช่า/ให้เช่าซื้อระบบคอมพิวเตอร์โดยตรงมาแล้วเป็นเวลาไม่น้อยกว่า 3 ปี นับจนถึงวันยื่นซองประกวดราคา มีทุนการจดทะเบียนไม่น้อยกว่า 10 ล้านบาท (ชำระเต็ม) และต้องมีประสบการณ์ในการติดตั้งระบบเบ็ดเสร็จ (Turn Key) มีมูลค่าไม่ต่ำกว่า 100 ล้านบาทต่อ 1 สัญญา และเป็นผลงานที่เป็นคู่สัญญาโดยตรงกับส่วนราชการ หน่วยงานตามกฎหมายว่าด้วยระเบียบบริหารราชการส่วนท้องถิ่น หรือหน่วยงานอื่นซึ่งมีกฎหมายบัญญัติให้มีฐานะเป็นราชการบริหารส่วนท้องถิ่น รัฐวิสาหกิจ หรือหน่วยงานเอกชนที่การท่าอากาศยานแห่งประเทศไทยเชื่อถือ

จากตัวอย่างข้างต้น จะเห็นว่าในการเขียนข้อกำหนดมีคำเชื่อมเกิน 3 แห่ง ทำให้ผู้อ่านเข้าใจความหมายได้ยาก ดังนั้น ควรมีการปรับโครงสร้างของรูปแบบประโยคใหม่ เพื่อให้ง่ายในการอ่าน ดังนี้

คุณสมบัติของบริษัทผู้เสนอราคา

1. บริษัทผู้เสนอราคาจะต้องเป็นนิติบุคคลที่จดทะเบียนในประเทศไทย

2. มีบุคลากรซึ่งมีความรู้และประสบการณ์ในงานด้านกิจการท่าอากาศยาน และสนับสนุนกิจการท่าอากาศยาน

3. ประกอบธุรกิจเกี่ยวกับการขาย และ/หรือการให้เช่า/ให้เช่าซื้อระบบคอมพิวเตอร์โดยตรงมาแล้วเป็นเวลาไม่น้อยกว่า 3 ปี นับจนถึงวันยื่นซองประกวดราคา มีทุนการจดทะเบียนไม่น้อยกว่า 10 ล้านบาท (ชำระเต็ม)

4. ต้องมีประสบการณ์ในการติดตั้งระบบเบ็ดเสร็จ (Turn Key) มีมูลค่าไม่ต่ำกว่า 100 ล้านบาทต่อ 1 สัญญา และเป็นผลงานที่เป็นคู่สัญญาโดยตรงกับส่วนราชการหรือหน่วยงานที่มีคุณสมบัติอย่างใดอย่างหนึ่ง ดังต่อไปนี้

4.1 หน่วยงานตามกฎหมายว่าด้วยระเบียบบริหารราชการส่วนท้องถิ่น

4.2 หน่วยงานอื่น ซึ่งมีกฎหมายบัญญัติให้มีฐานะเป็นราชการบริหารส่วนท้องถิ่น รัฐวิสาหกิจ

4.3 หน่วยงานเอกชนที่การท่าอากาศยานแห่งประเทศไทยเชื่อถือ

2.3.2.3 **ภาษาที่ใช้เขียนข้อกำหนด**

ปัญหาของภาษาธรรมชาติที่ใช้เขียนข้อกำหนด คือ ทำให้ข้อกำหนดเกิดความคลุมเครือ ความไม่ถูกต้อง และความไม่สอดคล้องกัน

ควรใช้คำต่อไปนี้ เพื่อแสดงถึงความสอดคล้องกันในการเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ในกรณีเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์เป็นภาษาอังกฤษ

1. "shall" ใช้ในลักษณะเป็นการกำหนด
2. "will" ใช้ในลักษณะเป็นการพรรณนา
3. "must" และ "must not" ใช้ในลักษณะเป็นข้อบังคับ
4. "should" ใช้ในลักษณะเป็นคำแนะนำ

หลีกเลี่ยงคำต่อไปนี้ เนื่องจากอาจทำให้มีความเข้าใจที่ไม่ตรงกัน

1. น้อยที่สุด ต่ำที่สุด อย่างน้อยที่สุด (as a minimum)
2. สามารถทำได้ น่าที่จะ (be able to, capable of)
3. ไม่ถูกจำกัด ไม่จำกัดที่ ไม่มีจำกัด ไม่ขึ้นกับ (not limited to)
4. ได้รับการกำหนด ถูกกำหนด (To Be Defined)

หลีกเลี่ยงคำหรือวลีที่ทำให้ผู้พัฒนาซอฟต์แวร์มีทางเลือกหลายทาง

1. อาจจะ (may)
2. ถ้าต้องการ (if required)
3. ถ้าเหมาะสม เมื่อมีความเหมาะสม (as appropriate)
4. เหมาะสมที่จะปฏิบัติ ถ้าใช้งานได้จริง ในทางปฏิบัติ (if practical)

ให้ระบุเป็นจำนวนตัวเลขที่แน่นอน และควรหลีกเลี่ยงคำขยายต่อไปนี้

1. ใหญ่ (large)
2. รวดเร็ว (rapid)
3. มากมาย หลายๆ หลากหลาย (many)
4. ถูกกาล ถูกเวลา (timely)
5. ส่วนมาก (most)
6. เกือบ (close)

หลีกเลี่ยงคำที่ทำให้เกิดความไม่ถูกต้อง ซึ่งทำให้ดูเหมือนว่าผู้พัฒนาซอฟต์แวร์ไม่มีความพยายามหรือไม่มีความสามารถในการพัฒนาซอฟต์แวร์

1. ง่าย (easy)
2. ปกติ (normal)
3. เพียงพอ (adequate)
4. ได้ผล อย่างแท้จริง (effective)