

บทที่ 4

การพิจารณาเพื่อปรับปรุงองค์กรซอฟต์แวร์เข้าสู่ระบบซีเอ็มเอ็ม

จากรายละเอียดแนวทางในการพัฒนาซีเอ็มเอ็ม และการออกแบบการพัฒนาองค์กรเข้าสู่ระบบซีเอ็มเอ็ม จะแสดงให้เห็นถึงแนวทางและรายละเอียด สิ่งที่องค์กรต้องดำเนินการในแต่ละระดับการเติบโตแล้วนั้น

ในบทนี้จะแสดงถึงรายละเอียดการพิจารณาเพื่อปรับปรุงองค์กรซอฟต์แวร์เข้าสู่ระบบซีเอ็มเอ็ม ข้อดีข้อเสียของการนำซีเอ็มเอ็มมาใช้ และข้อควรคำนึงในการนำมาใช้กับสังคมไทย เพื่อให้บริษัทซอฟต์แวร์ที่ดำเนินการอยู่และมีความประสงค์จะเข้าสู่ระบบซีเอ็มเอ็ม ได้ทราบว่าองค์กรจะต้องเตรียมดำเนินการในด้านใดบ้าง หรือต้องเพิ่มเติม ปรับปรุง แก้ไข วิธีการดำเนินงานในปัจจุบันอย่างไร เพื่อที่จะพัฒนาองค์กรให้เข้าสู่ระบบซีเอ็มเอ็ม

องค์กรซอฟต์แวร์จะต้องเตรียมดำเนินการในด้านต่าง ๆ ดังนี้ คือ

1. การจัดการองค์กรและทรัพยากร (Organization and Resource Management) ประกอบด้วย
 - 1.1 การจัดการโครงสร้างองค์กร (Organization Structure)
 - 1.2 การจัดการบริหารทรัพยากรที่ใช้ บุคลากร และการฝึกอบรม (Resource Personnel and Training)
 - 1.3 การจัดการบริหารเทคโนโลยี (Technology Management)
2. การจัดการบริหาร กระบวนการทางด้านวิศวกรรมซอฟต์แวร์ (Software Engineering Process Management) ประกอบด้วย
 - 2.1 การจัดการมาตรฐานองค์กร และขั้นตอนการทำงาน (Standards and Procedures)
 - 2.2 การวัดผลกระบวนการ (Process Metrics)
 - 2.3 การจัดการข้อมูลและการวิเคราะห์ข้อมูล (Data Management and Analysis)
 - 2.4 การควบคุมกระบวนการ (Process Control)
3. การจัดการบริหารเครื่องมือและเทคโนโลยีที่ใช้ (Tools and Technology Management)

รายละเอียดการพิจารณาเพื่อปรับปรุงองค์กรซอฟต์แวร์เข้าสู่ระบบซีเอ็มเอ็ม มีรายละเอียดดังนี้ คือ

1. การจัดการองค์กรและทรัพยากร (Organization and Resource Management) มีรายละเอียดเกี่ยวกับความรับผิดชอบในหน้าที่งานต่าง ๆ บุคลากร ทรัพยากรที่ใช้ และสิ่งอำนวยความสะดวกต่าง ๆ ซึ่งเน้นในเรื่องความรับผิดชอบ คุณภาพและปริมาณทรัพยากรที่ใช้ ประกอบด้วย

1.1 การจัดการโครงสร้างองค์กร (Organization Structure) องค์กรซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็มจะต้องเตรียมดำเนินการในการจัดการฝั่งโครงสร้างองค์กรใหม่ ตามลักษณะดังนี้ คือ

1.1.1 มีการแต่งตั้งตำแหน่งงานทางด้านวิศวกรรมซอฟต์แวร์ที่จำเป็น ดังนี้คือ

- มีการแต่งตั้งผู้จัดการซอฟต์แวร์ ให้มีหน้าที่รับผิดชอบในการจัดการควบคุม บริหารงานโครงการซอฟต์แวร์
- มีการแต่งตั้งบุคคลหรือทีมงานตรวจสอบรับรองคุณภาพ (Software Quality Assurance – SQA) ทำหน้าที่ในการตรวจสอบ ทวนและทดสอบคุณภาพเพื่อสร้างความเชื่อถือในคุณภาพซอฟต์แวร์
- มีการแต่งตั้งบุคคลหรือทีมงานควบคุมรายละเอียดขอบเขตโครงสร้างของซอฟต์แวร์ (Software Configuration Control – SCC) ทำหน้าที่ในการควบคุมการเปลี่ยนแปลงรายละเอียดต่าง ๆ ของซอฟต์แวร์ให้อยู่ภายในขอบเขตที่กำหนดไว้
- มีการแต่งตั้งบุคคลหรือทีมงานกลุ่มกระบวนการวิศวกรรมซอฟต์แวร์ (Process Group Software Engineering) ทำหน้าที่ในการจัดการควบคุมกระบวนการที่ใช้ในการพัฒนาซอฟต์แวร์
- มีการแต่งตั้งบุคคลหรือทีมงานด้านวิศวกรรมระบบซอฟต์แวร์ (Software System Engineering) ทำหน้าที่ในการออกแบบระบบซอฟต์แวร์ ออกแบบฐานข้อมูล ออกแบบสถาปัตยกรรมของซอฟต์แวร์ ฯลฯ
- มีการแต่งตั้งบุคคลหรือทีมงานควบคุมการติดต่อเชื่อมโยงกับซอฟต์แวร์ (Software Interfaces) ทำหน้าที่ในการจัดการควบคุมการติดต่อเชื่อมโยงกับซอฟต์แวร์

1.1.2 มีการจัดการองค์กรตามสายการบังคับบัญชา ดังนี้คือ

- ผู้จัดการซอฟต์แวร์ รายงานขึ้นตรงกับผู้จัดการโครงการ

- ทีมงานตรวจสอบรับรองคุณภาพ (SQA) รายงานขึ้นตรงกับผู้จัดการโครงการโดยตรง เพื่อให้การรายงานผลการจัดการบริหารหน้าที่ในการสร้างความเชื่อถือในคุณภาพซอฟต์แวร์ (SQA) แยกจากการรายงานผลการจัดการบริหารโครงการพัฒนาซอฟต์แวร์โดยทั่วไป
- ทีมงานด้านวิศวกรรมระบบซอฟต์แวร์ รายงานขึ้นตรงกับทีมงานการออกแบบระบบ (System Design Team)

1.2 การจัดการบริหารทรัพยากรที่ใช้ บุคลากร และการฝึกอบรม (Resource Personnel and Training) มีรายละเอียดเกี่ยวกับการจัดหาและการคัดเลือกบุคลากร การฝึกอบรมทางด้านวิศวกรรมซอฟต์แวร์ การฝึกอบรมทางด้านกระบวนการต่าง ๆ และความเพียงพอของอุปกรณ์อำนวยความสะดวกที่ช่วยสนับสนุน องค์กรซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็มจะต้องเตรียมดำเนินการและความพร้อมในด้านต่าง ๆ ดังนี้ คือ

1.2.1 มีการจัดการบริหารทรัพยากรที่ใช้ ซึ่งได้แก่ การจัดเตรียมหาอุปกรณ์คอมพิวเตอร์ และอุปกรณ์อำนวยความสะดวกรวมถึงโปรแกรมและเครื่องมือเพื่อช่วยสนับสนุนการทำงานของผู้พัฒนาซอฟต์แวร์อย่างเพียงพอ

1.2.2 ปรับปรุงวิธีการจัดหา และคัดเลือกบุคลากรที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ ในการจัดหาและคัดเลือกบุคลากรด้านซอฟต์แวร์ ควรคำนึงรายละเอียดด้านต่าง ๆ ดังนี้ คือ

- บุคลากรควรมีประสบการณ์ในการดำเนินการในโครงการซอฟต์แวร์
- บุคลากรควรมีความรู้ด้านเทคนิค และขั้นตอนการพัฒนาซอฟต์แวร์
- บุคลากรควรมีความรู้ด้านวิศวกรรมซอฟต์แวร์
- บุคลากรควรมีความรู้และประสบการณ์ในระบบงานต่าง ๆ (Application) ที่โครงการจะพัฒนา
- บุคลากรควรมีความสามารถในการใช้เครื่องมือหรือเทคโนโลยีที่นำมาช่วยในการพัฒนาซอฟต์แวร์
- บุคลากรควรได้รับการฝึกอบรมด้านการบริหารและเทคนิคในการทำงานต่าง ๆ

1.2.3 มีการจัดเตรียมโปรแกรมการฝึกอบรมให้ความรู้กับบุคลากรที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ ตามหน้าที่งานที่รับผิดชอบอย่างเพียงพอ ดังนี้คือ

- มีการจัดเตรียมโปรแกรมฝึกอบรมทางด้านการบริหารและวิศวกรรมซอฟต์แวร์ที่จำเป็นสำหรับผู้จัดการพัฒนาโครงการที่ได้รับการแต่งตั้งใหม่ได้แก่
 - การฝึกอบรมเกี่ยวกับขั้นตอนการพัฒนาซอฟต์แวร์

- การฝึกอบรมการใช้เครื่องมือ วิธีการ และเทคนิคที่ใช้ในการพัฒนา
 - การฝึกอบรมให้ความรู้เกี่ยวกับระบบงานที่พัฒนา
 - การฝึกอบรมการบริหารโครงการ
 - การฝึกอบรมการวางแผนงานโครงการ(Project Planning)
 - การฝึกอบรม การวิเคราะห์ และการตัดสินใจ (Decision Making)
 - การฝึกอบรมการวิเคราะห์ และการจัดการความเสี่ยง (Risk Analysis and Management)
 - การฝึกอบรมการตรวจสอบ และควบคุมโครงการ (Project Control and Monitoring)
 - การฝึกอบรม การแก้ไขปัญหาของโครงการ (Problem Solving)
 - การฝึกอบรมการควบคุมต้นทุนของโครงการ
 - การฝึกอบรม การบริหารทรัพยากรที่ใช้ในโครงการ
 - การฝึกอบรมความเป็นผู้นำ และหลักสูตรการบริหารทั่วไป
 - การฝึกอบรม การติดต่อสื่อสาร และการประสานงาน
 - การฝึกอบรมจิตวิทยาในการทำงาน
 - การฝึกอบรมการทำงานเป็นทีมงาน
 - การฝึกอบรมศิลปะการพูดในแบบต่างๆ
- มีการจัดเตรียมโปรแกรมการฝึกอบรมทางด้านวิศวกรรมซอฟต์แวร์ที่จำเป็น สำหรับผู้พัฒนาซอฟต์แวร์ (Software Developers) ได้แก่
- การฝึกอบรมความรู้เกี่ยวกับขั้นตอนการพัฒนาซอฟต์แวร์
 - การฝึกอบรมความรู้ทั่วไปเกี่ยวกับระบบงานที่จะพัฒนา
 - การฝึกอบรมมาตรฐาน และวิธีการที่ใช้ในแต่ละขั้นตอน
 - การฝึกอบรมการใช้เครื่องมือที่ช่วยในการพัฒนาซอฟต์แวร์
 - การฝึกอบรมเทคโนโลยีใหม่ๆ ด้านวิศวกรรมซอฟต์แวร์
 - การฝึกอบรมวิธีจัดทำคู่มือ และเอกสารการพัฒนาซอฟต์แวร์
 - การฝึกอบรมการรายงานสถานะความก้าวหน้าของงาน
- มีการจัดเตรียมโปรแกรมการฝึกอบรมทางด้านวิศวกรรมซอฟต์แวร์ที่จำเป็นสำหรับหัวหน้าระดับต้นของทีมพัฒนาซอฟต์แวร์ (First Line Supervisors) ได้แก่

- การฝึกอบรมความรู้เกี่ยวกับขั้นตอนการพัฒนาซอฟต์แวร์
 - การฝึกอบรมความรู้ทั่วไปเกี่ยวกับระบบงานที่จะพัฒนา
 - การฝึกอบรมมาตรฐาน และวิธีการที่ใช้ในแต่ละขั้นตอน
 - การฝึกอบรมการวางแผนงานโครงการ
 - การฝึกอบรมการควบคุมตรวจสอบ และประเมินงานโครงการ
 - การฝึกอบรมการจัดสรรทรัพยากรที่ใช้ในงาน
 - การฝึกอบรมการวิเคราะห์ข้อมูลโครงการ
 - การฝึกอบรมการรายงานสถานะความก้าวหน้าของงาน
- มีการจัดเตรียมโปรแกรมการฝึกอบรมทางด้านวิศวกรรมซอฟต์แวร์ที่จำเป็น สำหรับผู้ตรวจสอบ ทบทวนการออกแบบ และการเขียนโปรแกรม ได้แก่
- การฝึกอบรมความรู้เกี่ยวกับขั้นตอนการพัฒนาซอฟต์แวร์
 - การฝึกอบรมความรู้ทั่วไปเกี่ยวกับระบบงานที่จะพัฒนา
 - การฝึกอบรมมาตรฐาน และวิธีการในการออกแบบ
 - การฝึกอบรมมาตรฐาน และวิธีการในการเขียนโปรแกรม
 - การฝึกอบรมการวิเคราะห์ และประเมินสถานะโครงการ

1.3 การจัดการบริหารเทคโนโลยี (Technology Management) มีรายละเอียดเกี่ยวกับวิธีการ หรือเทคนิคในการนำเอาเทคโนโลยีใหม่มาใช้ และการควบคุมการใช้เทคโนโลยีนั้น ๆ องค์กรซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็มจะต้องเตรียมดำเนินการ และความพร้อมในด้านต่าง ๆ ดังนี้

1.3.1 จัดเตรียมวิธีการหรือเทคนิคที่ใช้ในการบอกให้ทราบถึงรายละเอียดของเทคโนโลยีทางด้านวิศวกรรมซอฟต์แวร์ ได้แก่

- เตรียมจัดโปรแกรมการฝึกอบรมให้ความรู้เกี่ยวกับเทคโนโลยีทางด้านวิศวกรรมซอฟต์แวร์
- จัดให้มีการศึกษารายละเอียดของเทคโนโลยีทางด้านวิศวกรรมซอฟต์แวร์ โดยการเป็นสมาชิกวารสารด้านคอมพิวเตอร์ หรือสมาชิกเครือข่ายอินเทอร์เน็ต หรือชมรมคอมพิวเตอร์ต่าง ๆ
- จัดให้มีการเผยแพร่ข่าวสารข้อมูลเทคโนโลยีทางด้านวิศวกรรมซอฟต์แวร์ภายในองค์กร เช่น ติดประกาศ ทำจดหมายข่าว แจงให้ทราบในการประชุม

1.3.2 จัดเตรียมวิธีการหรือเทคนิคที่ใช้ในการประเมินเทคโนโลยีที่ใช้ในองค์กรเปรียบเทียบกับเทคโนโลยีที่มีอยู่ภายนอก ได้แก่

- จัดให้มีการศึกษาเทคโนโลยีใหม่ ๆ ที่เกี่ยวข้องกับซอฟต์แวร์ โดยการเป็นสมาชิกองค์กรคอมพิวเตอร์ เป็นสมาชิกวารสารด้านวิชาคอมพิวเตอร์ เข้าร่วมประชุมสัมมนาทางวิชาการคอมพิวเตอร์
- จัดทำเอกสารขั้นตอนและวิธีการประเมินเปรียบเทียบเทคโนโลยี

1.3.3 จัดเตรียมวิธีการหรือเทคนิคที่ใช้เพื่อการพิจารณาว่าเมื่อไรควรจะแทรกเทคโนโลยีใหม่ ๆ เข้าไปยังกระบวนการพัฒนาซอฟต์แวร์ ได้แก่

- มีการจัดตั้งกลุ่มผู้รับผิดชอบด้านการบริหารและเปลี่ยนแปลงเทคโนโลยี
- มีการจัดทำเอกสารแบบฟอร์มความต้องการเทคโนโลยีใหม่ เพื่อให้ผู้ใช้หรือทีมงานพัฒนาซอฟต์แวร์ บันทึกรายละเอียดปัญหา และความต้องการเทคโนโลยีใหม่
- มีการจัดทำเอกสาร ขอบเขต ขั้นตอน และนโยบายที่ใช้ในการเปลี่ยนแปลงเทคโนโลยี
- มีการจัดทำเอกสารแผนงานการเปลี่ยนแปลงเทคโนโลยี

1.3.4 จัดเตรียมวิธีการ หรือเทคนิคที่ใช้ในการจัดการ และสนับสนุนเทคโนโลยีใหม่ในเบื้องต้น ได้แก่

- มีการจัดทำเอกสารการกำหนดส่วนของกระบวนการที่จะได้รับประโยชน์ในการเปลี่ยนแปลงเทคโนโลยี
- มีการจัดทำเอกสารการกำหนดขอบเขตวิธีการขั้นตอนในการเปลี่ยนแปลงเทคโนโลยี
- มีการจัดทำเอกสารการกำหนดรายละเอียดเทคโนโลยีใหม่
- มีการดำเนินการประมาณการวางแผน การขยายเทคโนโลยีใหม่ ตั้งแต่เริ่มต้น แนะนำ จนถึงการเปลี่ยนแทน
- มีการจัดทำเอกสารการศึกษาเปรียบเทียบวิธีการจัดหาเทคโนโลยีใหม่ (ทำเองหรือซื้อ)
- มีการดำเนินการกำหนดแนวทางในการประเมินเทคโนโลยี (ประสิทธิภาพ ต้นทุน ทรัพยากรที่ใช้)
- มีการจัดทำเอกสารการกำหนดขั้นตอนการจัดการเทคโนโลยี และการนำมาใช้งาน
- มีการจัดโปรแกรมการฝึกอบรมเบื้องต้น การฝึกอบรมทั่วไป และการให้คำปรึกษา

1.3.5 จัดเตรียมวิธีการ หรือเทคนิคที่ใช้ระบุและเปลี่ยนแปลงเทคโนโลยีที่ล้าสมัย ได้แก่

- มีการจัดตั้งกลุ่มผู้รับผิดชอบด้านการบริหารและเปลี่ยนแปลงเทคโนโลยี
- มีการจัดทำเอกสารขอบเขต ขั้นตอน และนโยบายที่ใช้ในการเปลี่ยนแปลงเทคโนโลยี
- มีการจัดทำเอกสารแผนงานการเปลี่ยนแปลงเทคโนโลยี
- มีการจัดทำเอกสารระบุ อธิบายสาเหตุการเปลี่ยนแปลงเทคโนโลยีที่ล้าสมัย
- มีการจัดทำเอกสารแบบฟอร์มการขออนุมัติการเปลี่ยนแปลงเทคโนโลยี

2. การจัดการบริหาร กระบวนการทางด้านวิศวกรรมซอฟต์แวร์ (Software Engineering Process Management) มีรายละเอียดเกี่ยวกับขอบเขต รายละเอียด และความพร้อมของกระบวนการด้านวิศวกรรมซอฟต์แวร์ มีการเตรียมวิธีการในการวัดผล การจัดการและการปรับปรุงพัฒนากระบวนการ ดังนี้ คือ

2.1 การจัดการมาตรฐานของเอกสารและขั้นตอนการทำงาน (Standards and Procedures) มีรายละเอียดเกี่ยวกับขอบเขตและการใช้งานรูปแบบขั้นตอนการทำงาน และเอกสารต่าง ๆ ที่เกิดขึ้นระหว่างขั้นตอนการพัฒนาซอฟต์แวร์ องค์กรซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็มจะต้องเตรียมดำเนินการ และความพร้อมในด้านต่าง ๆ ดังนี้ คือ

2.1.1 ปรับปรุงวิธีการพัฒนาซอฟต์แวร์ขององค์กร ให้ดำเนินการตามขั้นตอนกระบวนการพัฒนาซอฟต์แวร์ที่เป็นเอกสารลายลักษณ์อักษร มีรายละเอียดดังนี้ คือ

- องค์กรจะต้องปรับปรุงวิธีการพัฒนาซอฟต์แวร์ให้ดำเนินการตามขั้นตอนวงจรชีวิตของการพัฒนาซอฟต์แวร์ ประกอบด้วย
 - ขั้นตอนแนวคิดเบื้องต้น (Conception)
 - ขั้นตอนการกำหนดรายละเอียดความต้องการซอฟต์แวร์ (Software Requirements)
 - ขั้นตอนการออกแบบ (Design)
 - ขั้นตอนการนำซอฟต์แวร์ไปใช้งาน (Implementation)
 - ขั้นตอนการรวบรวมระบบงานและการทดสอบ (Integration and Test)
 - ขั้นตอนการบำรุงรักษาระบบงาน (Maintenance)

- องค์กรจะต้องจัดทำเอกสารในแต่ละขั้นตอนการพัฒนาซอฟต์แวร์ ได้แก่

1. ขั้นตอนแนวคิดเบื้องต้นจะต้องจัดทำเอกสารดังนี้

- สรุปรายละเอียดแนวคิดเบื้องต้นของระบบงานที่จะพัฒนา ประกอบด้วย

- ขอบเขตและวัตถุประสงค์
- รายละเอียดแนวคิดเบื้องต้นของระบบงาน
- ความสัมพันธ์ของแต่ละระบบย่อย
- การศึกษาความเป็นไปได้ของระบบงาน
- สรุปเป้าหมายที่คาดว่าจะได้รับ

- แผนงานเบื้องต้นของโครงการ (Master Project Plan) ประกอบด้วย

- รายละเอียดงานหลักที่ต้องดำเนินการ
- เวลาที่ใช้ในการดำเนินงาน
- ผู้รับผิดชอบ

2. ขั้นตอนการกำหนดรายละเอียดความต้องการซอฟต์แวร์ จะต้องจัดทำเอกสาร ดังนี้

- รายละเอียดความต้องการระบบงานซอฟต์แวร์ (Software Requirement Specification) ประกอบด้วย

- ขอบเขตของงาน
- เอกสารที่จะนำมาใช้
- ข้อมูลทั่วไป และข้อมูลการติดต่อเชื่อมโยงระบบงาน
- แผนผังโครงสร้างการเชื่อมโยงระบบงาน (Interface Diagram)
- แผนผังโครงสร้างทางเดินข้อมูล (Data flow Diagrams)
- รายละเอียดหน้าที่ต่าง ๆ (functions)
- รายละเอียดฐานข้อมูล (Quality Assurance)
- แผนผังโครงสร้างระบบงาน (Software Block Diagram)

- แผนงานพัฒนาโครงการ (Project Development Plan) ประกอบด้วย

- ขอบเขตของงาน
- รายละเอียดการส่งมอบ (Tasks and Deliverables)
- ทรัพยากรที่ใช้ (Resources)
- ตารางเวลาการทำงาน (Schedule)

- รายละเอียดต้นทุน (Cost)

- แผนงานการทดสอบซอฟต์แวร์ (Software Test Plan) ประกอบด้วย

- วัตถุประสงค์ และขอบเขตการทดสอบ
- แผนงานการทดสอบอย่างไม่เป็นทางการ
- แผนงานการทดสอบส่วนย่อย (Unit test Plans)
- แผนงานการทดสอบและการรวบรวมระบบงาน
- ทรัพยากรที่ต้องใช้
- แผนงานการทดสอบอย่างเป็นทางการ
- รายละเอียดสิ่งที่ต้องการทดสอบ
- ตารางเวลาการทดสอบ
- การวิเคราะห์ข้อมูลที่จะนำมาทดสอบ
- สมมติฐานที่ใช้
- ข้อจำกัดในการทดสอบ

3. ขั้นตอนการออกแบบซอฟต์แวร์ จะต้องจัดทำเอกสารดังนี้

- รายละเอียดการออกแบบ (Design Specification) ประกอบด้วย

- ขอบเขตของงาน
- เอกสารที่เกี่ยวข้อง (คู่มือ มาตรฐานต่างๆ)
- รายละเอียดการออกแบบ (Design Description)
- ผังทางเดินของข้อมูล/ โครงสร้าง (Data Flow / Structure)
- รายละเอียดผังทางเดินของข้อมูล/ โครงสร้าง
- โครงสร้างของซอฟต์แวร์ (Software Structure)
- รายละเอียดระบบย่อยต่างๆ
- รายละเอียดการประมวล
- รายละเอียดการติดต่อเชื่อมโยงระบบ
- การจัดการข้อมูล
- ข้อคิดเห็นต่างๆ (ข้อจำกัด ข้อผิดพลาดที่พบ)
- รายละเอียดโครงสร้างแฟ้มข้อมูล
- รายละเอียดรูปแบบ และองค์ประกอบของข้อมูล
- ทรัพยากรที่ใช้ (Resources)
- การตรวจสอบรับรองคุณภาพ (Quality Assurances)

- การตรวจสอบความถูกต้องการออกแบบ
 - แผนผังโครงสร้างซอฟต์แวร์ (Software Block Diagram)
 - รายละเอียดข้อมูลเพิ่มเติมต่าง ๆ
 - รายละเอียดกรณีศึกษาที่ต้องการทดสอบ (Test Case Specifications)
4. ขั้นตอนการนำซอฟต์แวร์ไปใช้งานจะต้องจัดทำเอกสารดังนี้
- บันทึกรายละเอียดต่าง ๆ ของโปรแกรมเมอร์ ประกอบด้วย
 - การทดสอบระบบย่อย
 - เอกสารการแก้ปัญหาที่เกิดขึ้น
 - แผนงานการบำรุงรักษาระบบงานซอฟต์แวร์ (Maintenance Plan) ประกอบด้วย
 - ขอบเขตของงาน
 - รายละเอียดการบำรุงรักษาระบบงาน
 - เทคนิคหรือวิธีการที่ใช้
 - รายละเอียดความต้องการในการตรวจสอบความถูกต้อง
 - ทรัพยากรที่ใช้
 - ต้นทุนที่ใช้
 - ตารางเวลาการทำงาน
 - เอกสารคู่มือการใช้งาน (Reference Manual) และคู่มือแนะนำการปฏิบัติงาน ประกอบด้วย
 - คำแนะนำเบื้องต้น
 - เอกสารที่ใช้
 - รายละเอียดหน้าที่งาน
 - ความสามารถของระบบ
 - สภาพแวดล้อมของระบบ
 - รายละเอียดของระบบ
 - คำสั่งในการติดตั้ง
 - กรณีทดสอบต่างๆ และผลลัพธ์
 - คู่มือแนะนำการปฏิบัติงาน (User Guides) ประกอบด้วย
 - คำแนะนำเบื้องต้น
 - เอกสารที่ใช้
 - รายละเอียดหน้าที่งาน

- โครงสร้างไฟล์ และโครงสร้างข้อมูล
- รายละเอียด และรูปแบบข้อมูลนำเข้า
- ตัวอย่างข้อมูลนำเข้า
- รายละเอียด และรูปแบบผลลัพธ์
- ตัวอย่างผลลัพธ์
- การประมวลผลข้อมูลนำเข้า-ผลลัพธ์
- แผนผังทางเดินของข้อมูล
- รายละเอียดข้อผิดพลาดที่อาจเกิดขึ้น และการแก้ไข

5. ขั้นตอนการรวบรวมระบบงาน และการทดสอบ จะต้องจัดทำเอกสาร ดังนี้

- เอกสารการทดสอบระบบงาน และรายงานการทดสอบ ประกอบด้วย

- ขอบเขตของการทดสอบ
- เอกสารที่ใช้ในการทดสอบ
- แผนงานการทดสอบ
- ขั้นตอนการทดสอบ
- รายงานสรุปผลการทดสอบ

- เอกสารการพัฒนาระบบงานซอฟต์แวร์ ประกอบด้วย

- เอกสารรายละเอียดแนวคิดเบื้องต้น
- รายละเอียดความต้องการระบบงานซอฟต์แวร์
- แผนงานพัฒนาโครงการ และแผนงานรายละเอียดต่างๆ
- รายละเอียดการออกแบบ
- รายละเอียดการเขียนโปรแกรม
- รายละเอียดการรวบรวมระบบงาน และการทดสอบ
- รายละเอียดการบำรุงรักษาระบบงานซอฟต์แวร์

- เอกสารคู่มือการใช้งาน (User Manuals) ประกอบด้วย

- วัตถุประสงค์
- รายละเอียดส่วนประกอบของระบบ
- หน้าที่ และความสามารถของระบบ
- ข้อจำกัดของระบบงาน
- การเริ่มต้นระบบงาน
- รายละเอียดข้อมูลนำเข้า
- รายละเอียดผลลัพธ์

- ลำดับการประมวลผลผลลัพธ์
- หัวข้อทางเลือกต่างๆ
- ข้อผิดพลาดที่อาจเกิดขึ้น และแนวทาง
- แก้อัปเดต
- คำแนะนำพิเศษอื่นๆ

6. ขั้นตอนการบำรุงรักษาระบบงาน จะต้องจัดทำเอกสารดังนี้

- เอกสารการบำรุงรักษาระบบงาน (Maintenance Documentation) ได้แก่

- บันทึกรายละเอียดการบำรุงรักษา (Maintenance Logs)
- รายงานปัญหาซอฟต์แวร์ (Software Change Report)

- เอกสารรายละเอียดชุดของซอฟต์แวร์ (Version Release Documentation)

2.1.2 มีการจัดทำคู่มือ หรือคำอธิบายรายละเอียดการใช้เครื่องมือ และเทคนิคต่าง ๆ เพื่อช่วยในการพัฒนาซอฟต์แวร์ในเอกสารขั้นตอนการพัฒนาซอฟต์แวร์ที่เป็นมาตรฐาน ได้แก่

- มีเอกสารรายละเอียดการใช้เครื่องมือที่ช่วยในการวิเคราะห์ความต้องการ (อย่างใดอย่างหนึ่ง) ได้แก่

- ภาษาที่ใช้สำหรับอธิบายรายละเอียดระบบ – PSL (Problem Statement Language)
- ซอฟต์แวร์ที่ช่วยวิเคราะห์ และประเมินผล – PSA (Problem Statement Analyzer)

- มีเอกสารรายละเอียดการใช้เครื่องมือ และเทคนิค ที่ช่วยในการออกแบบ (อย่างใดอย่างหนึ่ง) ได้แก่

- ผังแสดงโครงสร้างโปรแกรม (Flowcharts)
- เครื่องมือสร้างผังแสดงโครงสร้างโปรแกรม (Flowcharter)

- ผังโครงสร้างแบบกล่อง (Box Diagrams) หรือ Nassi – Shneiderman

- ภาษาที่ช่วยในการออกแบบโปรแกรม (Program Design Language – PDL) หรือ Pseudo – Code

- ตารางการตัดสินใจ (Decision Table)

- มีเอกสารรายละเอียดการใช้เครื่องมือที่ช่วยในการนำซอฟต์แวร์ไปใช้งาน (อย่างใดอย่างหนึ่ง) ได้แก่

- มีเอกสารรายละเอียดการใช้เครื่องมือที่ช่วยในการจัดเตรียมรหัสโปรแกรม (Editor)
- มีเอกสารรายละเอียดการใช้เครื่องมือที่ช่วยในการประมวลผลภาษา ได้แก่
 - คอมไพเลอร์ช่วยประมวลผล (Precompiler)
 - ซอฟต์แวร์ช่วยประมวลผลภาษา (Macroprocessor)
 - คอมไพเลอร์ชนิดพิเศษ เพิ่มคุณสมบัติต่างๆ
- มีเอกสารรายละเอียดการใช้เครื่องมือที่ช่วยในการตรวจสอบความถูกต้องโปรแกรม ได้แก่
 - ซอฟต์แวร์ช่วยเปรียบเทียบโปรแกรม (Comparator)
 - ซอฟต์แวร์ช่วยอ้างอิงถึงตัวแปร และเงื่อนไขที่ใช้ (Cross – Reference Generator)
 - ซอฟต์แวร์ช่วยแปลงภาษาเครื่อง ไปเป็นโปรแกรม (Decompiler)
- มีเอกสารรายละเอียดการใช้เครื่องมือที่ช่วยในการทดสอบ (อย่างใดอย่างหนึ่ง) ได้แก่
 - มีเอกสารรายละเอียดการใช้เครื่องมือวิเคราะห์แผนผังโครงสร้างโปรแกรม (Program Flow Analyzer)
 - มีเอกสารรายละเอียดการใช้ซอฟต์แวร์สร้างข้อมูลทดสอบควบคุม และติดตามรายงานผล ได้แก่
 - Test Drivers
 - Scripts
 - Data Generators
 - การทดสอบสภาพแวดล้อมการทำงานจริง (Test Bed)
- มีเอกสารรายละเอียดการใช้เครื่องมือที่ช่วยในการวิเคราะห์ผลการปฏิบัติงาน (Performance Analysis Tools) ได้แก่
 - ซอฟต์แวร์ช่วยตรวจสอบการดำเนินงาน (Monitors)
 - ซอฟต์แวร์ช่วยในการวัดผลการประมวลผลโปรแกรม (Timing Analyzer)

2.1.3 จัดเตรียมวิธีการ หรือเทคนิคเพิ่มเติมที่ใช้สำหรับการดำเนินการในขั้นตอนการพัฒนาซอฟต์แวร์ ได้แก่

- จัดเตรียมวิธีการ หรือเทคนิคที่ใช้สำหรับตรวจสอบบริษัท ที่รับจ้างผลิตซอฟต์แวร์ (Subcontractors) ซึ่งมีรายละเอียดดังนี้
 - มีการจัดตั้งกลุ่มผู้ตรวจสอบการทำงาน

- มีการจัดเตรียมขั้นตอน และวางแผนการตรวจสอบ
 - มีเอกสารการกำหนดขอบเขต วิธีการ และมาตรฐานที่ใช้ในการตรวจสอบ
 - มีการตรวจสอบการทำงานของบริษัทที่รับจ้าง ตามขั้นตอนการพัฒนาซอฟต์แวร์ ซึ่งจะตรวจสอบการใช้ทรัพยากร และมาตรฐานที่ใช้เปรียบเทียบกับแผนงาน โดยกลุ่มผู้ตรวจสอบ
 - มีการจัดทำรายงานการตรวจสอบกิจกรรมต่างๆที่เกิดขึ้นตามขั้นตอนการพัฒนาซอฟต์แวร์ โดยผู้ตรวจสอบ
 - มีการตรวจสอบผลการดำเนินงานของบริษัทที่รับจ้างผลิตซอฟต์แวร์ ตามช่วงเวลา โดยทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์
 - มีการจัดทำเอกสารสรุปผลการตรวจสอบ
- จัดเตรียมวิธีการ หรือเทคนิคที่ใช้ในการประเมินการออกแบบ และรหัสโปรแกรมเพื่อนำกลับมาใช้ใหม่ (Reused) ซึ่งมีรายละเอียดดังนี้
- มีการจัดทำเอกสารแสดงการประเมินความถี่ หรือจำนวนครั้งที่ใช้แบบเดียวกัน
 - มีการจัดทำเอกสารแสดงการผ่านการตรวจสอบแล้วว่าโปรแกรมนั้นถูกต้อง สามารถนำกลับมาใช้ใหม่ได้อีก
 - มีการจัดทำเอกสารแสดงการนำโปรแกรม หรือการออกแบบนั้นไปใช้จริง
 - มีการจัดทำเอกสารแสดงการประเมินความยาก และซับซ้อนของการออกแบบ หรือรหัสโปรแกรม
 - มีการจัดทำเอกสารแสดงการลงทุนในทรัพยากร และเวลาที่ใช้ไปมาก
 - มีการจัดทำเอกสารแสดงการศึกษาผลกระทบการนำกลับมาใช้ใหม่
 - มีการจัดทำเอกสารแสดงการศึกษาความเป็นไปได้ในการนำกลับมาใช้ใหม่
 - มีการจัดทำเอกสารแสดงการวิเคราะห์ผลดี – ผลเสีย การนำกลับมาใช้ใหม่
- จัดเตรียมวิธีการ หรือเทคนิคเพื่อให้ทีมงานออกแบบซอฟต์แวร์เข้าใจถึงความต้องการซอฟต์แวร์ที่แท้จริง

- มีการจัดทำเอกสารรายงานสรุปรายละเอียดสิ่งที่ต้องการ (Software Requirements Specification) เพื่อยืนยันความต้องการ
- มีการจัดทำเอกสารผังทางเดินของข้อมูล (Data flow Diagram) เพื่อยืนยันความต้องการ
- มีการประชุมเพื่อรายงานผลการสรุปรายละเอียดสิ่งที่ต้องการ ระหว่างทีมงานออกแบบ และผู้ใช้
- มีการจัดทำเอกสาร หรือโปรแกรมตัวอย่างแบบจำลอง โปรแกรม (Program Prototype) เพื่อยืนยันความต้องการทุกโปรแกรมโดยละเอียด ประกอบด้วย
 - เอกสารหรือโปรแกรมตัวอย่างหน้าจอ ภาพการบันทึกข้อมูล
 - เอกสารหรือโปรแกรมตัวอย่างขั้นตอนการประมวลผล
 - เอกสารหรือโปรแกรมตัวอย่างรายงานทางเครื่องพิมพ์
 - เอกสารหรือโปรแกรมตัวอย่างหน้าจอภาพเมนูโปรแกรม ฯลฯ
- มีการเซ็นชื่ออนุมัติในทุกตัวอย่างโปรแกรมที่ออกแบบ ตามสรุปรายละเอียดสิ่งที่ต้องการ
- มีการติดตามตรวจสอบการออกแบบ (ปริมาณ คุณภาพตามแผนงาน ตรงกับความต้องการ) โดยทีมงานการตรวจสอบคุณภาพซอฟต์แวร์ ตามช่วงระยะเวลาที่กำหนด
- มีการจัดทำเอกสารสรุปการตรวจสอบ และรายงานผลให้กับผู้จัดการซอฟต์แวร์

2.1.4 จัดทำ ปรับปรุง เพิ่มเติมขั้นตอนการทำงานในการพัฒนาซอฟต์แวร์ให้เป็นทางการในแต่ละขั้นตอนที่สำคัญ ได้แก่

- จัดทำ ปรับปรุงเพิ่มเติมขั้นตอนที่ใช้ในการตรวจสอบก่อนที่จะทำสัญญาข้อตกลงดำเนินการพัฒนาซอฟต์แวร์ ซึ่งมีรายละเอียดดังนี้
 - มีการจัดเตรียมขั้นตอน และวางแผนการประมูลงาน
 - มีการแต่งตั้งคณะกรรมการคัดเลือกผู้ดำเนินการพัฒนาซอฟต์แวร์
 - มีการจัดทำรายละเอียดข้อเสนอสิ่งที่ต้องการแจกจ่ายไปให้กับผู้ประมูลงาน ให้ตอบกลับ เพื่อให้ทราบรายละเอียดผลิตภัณฑ์ของผู้ประมูลงานแต่ละราย ประกอบด้วย
 - คำชี้แจงปัญหา
 - ความต้องการทางด้านเทคนิค

- ความต้องทางด้านระบบงาน
- ความต้องการทางด้านต้นทุน
- เอกสารต่างๆ ที่เกี่ยวข้อง
- สิ่งที่ต้องการให้ส่งมอบ
- รายละเอียดข้อเสนอด้านต่างๆ
- ขอบเขตของงานที่ต้องทำ
- ตารางเวลาในการยื่นข้อเสนอ
- การตัดสินใจคัดเลือก

- มีการจัดทำขั้นตอนในการคัดเลือกอย่างเป็นทางการ โดยคณะกรรมการที่รับผิดชอบ

- มีการกำหนดเกณฑ์ในการตัดสินใจ โดยคณะกรรมการที่รับผิดชอบประกอบด้วย

- รายละเอียดระบบงานที่ต้องการ
- เทคโนโลยี และเครื่องมือที่ใช้ในการ
- พัฒนาซอฟต์แวร์
- วิธีการและมาตรฐานที่ใช้ในการพัฒนา ซอฟต์แวร์
- บริษัทอ้างอิงที่เคยใช้ระบบงานนี้มาก่อน
- ประสบการณ์ของบริษัทผู้ประมูลงาน
- โครงสร้างองค์กร และความมั่นคงของบริษัทผู้ประมูลงาน
- ที่ตั้งของบริษัทผู้ประมูลงาน
- บุคลากรที่รับผิดชอบ
- ทรัพยากรอำนวยความสะดวก
- สิ่งที่จะต้องส่งมอบ (โปรแกรมเอกสารการพัฒนา คู่มือการฝึกอบรม ฯลฯ)

- มีการประเมินรายละเอียด เงื่อนไข และความสามารถของผู้ประมูลงาน โดยคณะกรรมการ ซึ่งจะอาศัยรายละเอียดขอบเขตของงานที่ตอบกลับมา

- มีการตัดสินใจคัดเลือกบริษัทผู้พัฒนาซอฟต์แวร์ โดยใช้เกณฑ์ในการตัดสินใจ จากคณะกรรมการ

- มีการตรวจสอบเอกสาร รายละเอียด เงื่อนไขของสัญญา โดยคณะกรรมการที่รับผิดชอบ ซึ่งประกอบด้วย

- รายละเอียดขอบเขตของงาน (แนบเอกสารข้อเสนอสิ่งที่ต้องการ หรือ สรุปรายละเอียดโปรแกรมทั้งหมด)

- มูลค่าของซอฟต์แวร์
- เงื่อนไขการชำระเงิน
- แผนงานตารางเวลาการทำงาน และผู้รับผิดชอบ
- กำหนดการส่งมอบและเงื่อนไขกรณีที ล่าช้าหรือผิดนัด
- รายละเอียด และเงื่อนไข สิ่งที่จะส่งมอบ (โปรแกรม เอกสารการพัฒนา คู่มือ การฝึกอบรม ฯลฯ)
- รายละเอียด และเงื่อนไข การนำโปรแกรมไปใช้งาน
- เงื่อนไขการรับประกัน และระยะเวลาในการรับประกัน
- รายละเอียด และเงื่อนไขการปรับปรุง
- แก้ไขโปรแกรม
- รายละเอียด และเงื่อนไขอื่นๆ

- จัดทำ ปรับปรุง เพิ่มเติมขั้นตอนการตรวจสอบสถานะของโครงการพัฒนาซอฟต์แวร์ โดยผู้จัดการซอฟต์แวร์ตามช่วงเวลาอย่างเป็นทางการ ซึ่งมีรายละเอียดดังนี้

- มีการจัดทำเอกสารรายงานสถานะการทำงานตามช่วงเวลาเป็นลายลักษณ์อักษร ประกอบด้วย

- กิจกรรมต่างๆ ที่ได้ดำเนินการแล้ว
- กิจกรรมต่างๆ ที่วางแผนจะดำเนินการต่อไป
- รายละเอียดปัญหาต่างๆ ที่เกิดขึ้น

- มีการจัดทำรายงานแบบปากเปล่า (Verbal Reports) โดยให้ทีมงานพัฒนา รายงานความก้าวหน้า ผลการดำเนินงาน ปัญหาในการพัฒนาซอฟต์แวร์ ต่อผู้จัดการซอฟต์แวร์โดยตรง

- มีการจัดประชุมสถานะของโครงการ (Project Status meetings) ระหว่างทีมงานพัฒนาซอฟต์แวร์ และผู้จัดการ ประกอบด้วย

- มีการรายงานกิจกรรมโดยทั่วไปของโครงการ
- มีการแจ้งให้ทราบถึงเรื่องต่างๆไป
- มีการรายงานผลของกิจกรรมต่างๆ โดยผู้เข้าร่วมประชุม

- มีการสรุปแนวทางการแก้ไขปัญหาที่เกิดขึ้น

- มีการสาธิตผลิตภัณฑ์ที่กำลังพัฒนา (Product demonstration – Demos) ให้กับผู้จัดการ และทีมงานที่เกี่ยวข้อง ตามช่วงระยะเวลา

- มีการติดตามตรวจสอบการดำเนินงานแบบทันทีทันใดผ่านหน้าจอคอมพิวเตอร์โดยผู้จัดการ

- มีการปรับปรุงตารางเวลาการทำงานให้ทันสมัยอยู่เสมอ
(Updating the Schedule) โดยทีมงานการพัฒนาซอฟต์แวร์ และผู้จัดการ
- จัดทำ ปรับปรุง เพิ่มเติม ขั้นตอน การประมาณขนาดของซอฟต์แวร์อย่างเป็นทางการ ซึ่งมีรายละเอียดดังนี้
 - มีการจัดทำเอกสารกำหนดขอบเขตของงาน หน้าที่ และทรัพยากรที่ใช้ ในการพัฒนาซอฟต์แวร์
 - มีการจัดทำเอกสารการกำหนดวันเริ่มต้น และวันสิ้นสุดของโครงการพัฒนาซอฟต์แวร์
 - มีการจัดทำเอกสารการแบ่งงานของโครงการออกเป็นกิจกรรมงานย่อยๆ (Work Breakdown Structure Tasks)
 - มีการจัดทำเอกสารตารางเวลา พิจารณาตามทรัพยากร ต้นทุน เวลาที่มีอยู่
 - มีการจัดทำแผนผังตารางเวลาการทำงานโดยใช้วิธีการต่างๆ (อย่างใดอย่างหนึ่ง) ดังนี้
 - ตารางรายละเอียดกิจกรรม (Schedule Activity List)
 - GANTT Charts
 - PERT Charts (Program Evaluation and Review Technique)
 - CPM (Critical Path Method) ฯลฯ
 - มีการตรวจสอบอนุมัติตารางเวลาโดยผู้จัดการซอฟต์แวร์
 - มีการติดตามตรวจสอบการดำเนินงาน เปรียบเทียบกับตารางเวลาการทำงาน เพื่อพิจารณาปรับเปลี่ยนแผนงาน
 - มีการจัดการปรับปรุงแก้ไขเอกสารตารางเวลาเป็นประจำ พิจารณาจากวันที่จัดทำ และวันที่แก้ไขในเอกสารตารางเวลา
- จัดทำ ปรับปรุง เพิ่มเติม ขั้นตอนในการประเมินต้นทุนในการพัฒนาซอฟต์แวร์อย่างเป็นทางการ ซึ่งมีรายละเอียดดังนี้
 - มีการแต่งตั้งผู้รับผิดชอบในการประเมินต้นทุน
 - มีการจัดทำเอกสารการกำหนดขอบเขตของซอฟต์แวร์ที่จะทำการประเมินต้นทุน
 - มีการจัดทำเอกสารการกำหนดมาตรฐาน นโยบายและขั้นตอนในการประเมินต้นทุน
 - มีการจัดทำเอกสารการกำหนดวิธีการและเทคนิคที่ใช้ (อย่างใดอย่างหนึ่ง) ได้แก่

- การประเมินโดยอาศัยประสบการณ์ (Experience Method)
 - การประเมินโดยอาศัยข้อจำกัด (Constraint Method)
 - การประเมินจากบนลงล่าง (Top-Down Estimating)
 - การประเมินโดยใช้อัตราส่วน(Ratio Estimating)
 - การประเมินโดยใช้มาตรฐาน (Standards Estimating)
 - การประเมินจากล่างขึ้นบน (Bottom-Up Estimating)
 - การประเมินโดยใช้ปริมาณงาน (Units of Work Method)
 - การประเมินโดยใช้จำนวนของคำสั่ง (Number of Instructions)
 - การประเมินโดยใช้จำนวนของคำสั่ง (Number of Instructions)
 - การประเมินโดยใช้วิธีเชิงปริมาณ (Quantitative Method)
 - การประเมินโดยใช้ต้นทุนต่อคำสั่ง (Cost per Instruction)
 - การประเมินโดยใช้เปอร์เซ็นต์ของฮาร์ดแวร์ (Percent of Hardware Method)
 - Putnam Model
 - COCOMO (Constructive Cost Model)
- มีการดำเนินการประเมินต้นทุนในการพัฒนาซอฟต์แวร์ตามหลักเกณฑ์ที่ตั้งไว้ โดยผู้รับผิดชอบ
 - มีการจัดทำเอกสารสรุปและรายงานผลการประเมินต้นทุนในการพัฒนาซอฟต์แวร์ โดยผู้รับผิดชอบ เสนอให้กับผู้จัดการซอฟต์แวร์
 - จัดทำ ปรับปรุง เพิ่มเติม การดำเนินการตรวจสอบอิสระ ในแต่ละขั้นตอนการพัฒนาซอฟต์แวร์ ซึ่งมีรายละเอียดดังนี้
 - มีการจัดตั้งกลุ่มผู้ตรวจสอบอิสระให้มีการรายงานขึ้นตรงกับผู้จัดการระดับสูงของโครงการ พิจารณาจากฝั่งโครงสร้างองค์กร และการรายงานผล
 - มีการจัดเตรียมขั้นตอน และวางแผนการตรวจสอบ
 - มีการจัดทำเอกสารการกำหนดมาตรฐานในการตรวจสอบ

- มีการจัดทำเอกสารการกำหนดวิธีการ และเครื่องมือที่ใช้ในการตรวจสอบ
- มีการจัดทำตรวจสอบการดำเนินงานในแต่ละขั้นตอนการพัฒนาซอฟต์แวร์ โดยกลุ่มผู้ตรวจสอบ ซึ่งจะพิจารณาจากเอกสารที่ใช้ และผลการดำเนินงาน ประกอบด้วย
 - มีการดำเนินการตรวจสอบการดำเนินงานในขั้นตอนแนวคิดเบื้องต้น
 - มีการดำเนินการตรวจสอบการดำเนินงานในขั้นตอนการกำหนดรายละเอียดความต้องการซอฟต์แวร์
 - มีการดำเนินการตรวจสอบการดำเนินงานในขั้นตอนการออกแบบ
 - มีการดำเนินการตรวจสอบการดำเนินงานในขั้นตอนการนำซอฟต์แวร์ไปใช้งาน
 - มีการดำเนินการตรวจสอบการดำเนินงานในขั้นตอนการรวบรวมระบบงาน และการทดสอบ
 - มีการดำเนินการตรวจสอบการดำเนินงานในขั้นตอนการบำรุงรักษาระบบงาน
 - มีการบันทึกสรุป และรายงานผลการตรวจสอบการดำเนินงานไปยังผู้จัดการระดับสูงของโครงการ

2.1.5 จัดทำมาตรฐานในการทำงานในแต่ละขั้นตอนการพัฒนาซอฟต์แวร์ และนำมาประยุกต์ใช้ ได้แก่

- จัดทำมาตรฐานในการเก็บรวบรวมรายละเอียดของแฟ้มข้อมูลที่ใช้ในการพัฒนาซอฟต์แวร์และนำมาประยุกต์ใช้
- จัดทำมาตรฐานในการเขียนรหัสโปรแกรม (Coding Standards) และนำมาประยุกต์ใช้
- จัดทำมาตรฐานในการจัดเตรียมตัวอย่างกรณีศึกษาที่จะใช้ในการทดสอบส่วนย่อย (Unit test Case Standards)
- จัดทำมาตรฐานในการบำรุงรักษารหัสโปรแกรม (Code Maintainability Standards) และนำมาประยุกต์ใช้
- จัดทำมาตรฐานในการตรวจสอบทบทวนการออกแบบภายในซอฟต์แวร์ (Internal Design Review Standards) และนำมาประยุกต์ใช้
- จัดทำมาตรฐานในการตรวจสอบทบทวนการเขียนรหัสโปรแกรม (Code Review Standards) และนำมาประยุกต์ใช้

- จัดทำมาตรฐานในการติดต่อเชื่อมโยงระหว่างผู้ใช้และคอมพิวเตอร์ และนำมาประยุกต์ใช้

2.2 การวัดผลกระบวนการ (Process Metrics) มีรายละเอียดเกี่ยวกับระดับของกระบวนการด้านวิศวกรรมซอฟต์แวร์ที่จะถูกวัดผลซึ่งจะเกี่ยวกับคุณภาพของซอฟต์แวร์ปริมาณของจำนวนโปรแกรมที่พัฒนา ทรัพยากรที่ใช้ และสิ่งที่บอถึงความก้าวหน้าของกระบวนการ องค์กรซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็ม จะต้องเตรียมดำเนินการ และความพร้อมในด้านต่าง ๆ ดังนี้

2.2.1 มีการจัดเก็บข้อมูลจริงเปรียบเทียบกับแผนงาน เพื่อวัดผลกระบวนการในการพัฒนาซอฟต์แวร์ ในด้านต่าง ๆ ดังนี้

- มีการจัดเก็บข้อมูลพนักงานที่ใช้ไปในการพัฒนาซอฟต์แวร์จริง เปรียบเทียบกับข้อมูลจำนวนพนักงานตามแผนงาน ได้แก่
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการสำรวจความต้องการระบบงานซอฟต์แวร์ ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการออกแบบซอฟต์แวร์ ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการเขียนโปรแกรม ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการทดสอบระบบงาน ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการรวบรวมระบบงาน ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการนำซอฟต์แวร์ไปใช้งาน ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการบำรุงรักษาระบบงาน ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการตรวจสอบ และควบคุมคุณภาพ (Quality Assurance) ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ใช้ในการกำหนดขอบเขตโครงสร้างซอฟต์แวร์ (Configuration Control) ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลจำนวนพนักงานที่ปรึกษาด้านวิศวกรรมซอฟต์แวร์ ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการเก็บข้อมูลขนาดหรือปริมาณของส่วนประกอบ โครงสร้างซอฟต์แวร์ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน ได้แก่

- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการเก็บ รายละเอียดที่ใช้ในการกำหนดขอบเขต (ส่วนประกอบระบบย่อยต่าง ๆ รูปแบบ คุณสมบัติเงื่อนไข ประสิทธิภาพของซอฟต์แวร์) ที่ใช้ไปจริง เปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการเก็บ รายละเอียดระบบย่อยต่าง ๆ ที่เป็นส่วนประกอบของซอฟต์แวร์ (จำนวนโปรแกรม รูปแบบ คุณสมบัติ เงื่อนไข ประสิทธิภาพ และความสัมพันธ์ของระบบย่อย) ที่ใช้ไปจริงเปรียบเทียบกับแผนงาน
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการเก็บ รายละเอียดข้อมูลเวอร์ชันของระบบงานซอฟต์แวร์ และเวอร์ชันของเอกสารที่ใช้ไปจริงเปรียบเทียบกับแผนงาน
- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการออกแบบส่วนย่อยของซอฟต์แวร์ (Software Units Designed) ที่ใช้ไปจริงเปรียบเทียบกับแผนงาน ได้แก่
- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการออกแบบโปรแกรม ประกอบด้วย
 - ส่วนประกอบ และโครงสร้างโปรแกรม
 - รายละเอียดตัวแปรที่ใช้
 - รูปแบบการนำเข้าข้อมูล
 - รูปแบบการประมวลผลข้อมูล
 - อัลกอริทึม และ ลอจิกโปรแกรม
 - รูปแบบการแสดงผลข้อมูล
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการออกแบบการติดต่อเชื่อมผู้ใช้ ประกอบด้วย
 - การออกแบบรูปแบบและวิธีการที่ใช้ในการติดต่อเชื่อมโยงผู้ใช้ (คำสั่ง แบบฟอร์ม เมนู)
 - การออกแบบการแสดงความผิดพลาด
 - การออกแบบการแสดงความช่วยเหลือ
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการออกแบบการติดต่อสื่อสาร ประกอบด้วย
 - การออกแบบรูปแบบ และ รายละเอียดการติดต่อสื่อสาร
 - การออกแบบอุปกรณ์เครื่องมือที่ใช้
 - การออกแบบโปรแกรมการติดต่อสื่อสารที่ใช้
- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการดำเนินการเพื่อให้ส่วนย่อยของซอฟต์แวร์เสร็จสมบูรณ์ (Software Unit Completing) และการทดสอบส่วนย่อย (Unit Testing) ที่ใช้ไปจริงเปรียบเทียบกับกับแผนงาน ได้แก่
- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการเขียนโปรแกรมส่วนย่อยของระบบงานทั้งหมด ที่ใช้ไปจริงเปรียบเทียบกับกับแผนงาน

- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการเขียนโปรแกรมเพื่อติดต่อเชื่อมโยงผู้ใช้ และการติดต่อสื่อสาร ที่ใช้ไปจริงเปรียบเทียบกับกับแผนงาน
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการเขียนโปรแกรมย่อย ที่ใช้ไปจริงเปรียบเทียบกับกับแผนงาน
- มีการจัดเก็บข้อมูลเวลาที่ใช้ในการรวบรวมส่วนย่อยของซอฟต์แวร์ ที่ใช้ไปจริงเปรียบเทียบกับกับแผนงาน ได้แก่
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการรวบรวมโปรแกรมส่วนย่อย เข้าเป็นระบบงานทั้งหมดที่ใช้ไปจริงเปรียบเทียบกับกับแผนงาน
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการรวบรวมโปรแกรมส่วนย่อย เข้ากับโปรแกรมการติดต่อเชื่อมโยงผู้ใช้ที่ใช้ไปจริงเปรียบเทียบกับกับแผนงาน
 - มีการจัดเก็บข้อมูลเวลาที่ใช้ในการรวบรวมโปรแกรมส่วนย่อย เข้ากับโปรแกรมการติดต่อสื่อสาร ใช้ไปจริงเปรียบเทียบกับกับแผนงาน
- มีการจัดเก็บข้อมูลรายละเอียดการพัฒนาซอฟต์แวร์ และการเริ่มใช้ซอฟต์แวร์ (Software Build / Release)เปรียบเทียบกับเวลา ได้แก่
 - มีการเก็บรายละเอียดขอบเขตของซอฟต์แวร์ที่พัฒนา และ จัดทำเป็นเอกสาร
 - มีการเก็บข้อมูล วัน เวลาที่เริ่มทำการพัฒนาซอฟต์แวร์ และ จัดทำเป็นเอกสาร
 - มีการเก็บข้อมูล วัน เวลาที่เริ่มต้น – สิ้นสุดในขั้นตอนต่างๆ ของการพัฒนา และ จัดทำเป็นเอกสาร
 - มีการเก็บข้อมูลรายละเอียดทรัพยากรที่ใช้ และ จัดทำเป็นเอกสาร
 - มีการเก็บข้อมูลวัน เวลา ที่พัฒนาเสร็จ และเริ่มใช้ซอฟต์แวร์ และ จัดทำเป็นเอกสาร
 - มีการเก็บข้อมูลรายละเอียดเวอร์ชันของซอฟต์แวร์ และ จัดทำเป็นเอกสาร ประกอบด้วย
 - ชื่อชุดของเวอร์ชัน
 - วัน เวลาที่เริ่มใช้
 - รายละเอียด คุณสมบัติ ของแต่ละเวอร์ชัน
 - ข้อแตกต่างของแต่ละเวอร์ชัน
- มีการจัดเก็บข้อมูล การคาดการณ์ข้อผิดพลาดในการออกแบบที่อาจเกิดขึ้น เปรียบเทียบกับสิ่งที่เกิดขึ้นจริง ได้แก่
 - มีการคาดการณ์ว่าการออกแบบไม่มีโครงสร้างและมาตรฐาน
 - มีการคาดการณ์ว่าการออกแบบไม่ตรงกับความต้องการ

- มีการคาดการณ์ว่าการออกแบบและการเปลี่ยนแปลงแก้ไขอยู่นอกเหนือขอบเขต
 - มีการคาดการณ์ข้อผิดพลาดในการกำหนดรายละเอียดและหน้าที่งานการออกแบบ
 - มีการคาดการณ์ข้อผิดพลาดของโครงสร้างข้อมูลและผังทางเดินข้อมูล
 - มีการคาดการณ์ข้อผิดพลาดของโครงสร้างซอฟต์แวร์
 - มีการคาดการณ์ข้อผิดพลาดของรายละเอียดระบบย่อยต่าง ๆ
 - มีการคาดการณ์ข้อผิดพลาดของรายละเอียดการติดต่อเชื่อมโยง
 - มีการคาดการณ์ข้อผิดพลาดของรายละเอียดระหว่างระบบย่อย
 - มีการคาดการณ์ข้อผิดพลาดจากการตรวจสอบทบทวนการออกแบบ
 - มีการคาดการณ์ข้อผิดพลาดในเอกสารการออกแบบ
 - มีการจัดทำรายงานสรุปผลการคาดการณ์ข้อผิดพลาดที่อาจเกิดขึ้นเปรียบเทียบกับที่เกิดขึ้นจริง
- มีการจัดเก็บข้อมูล การคาดการณ์ข้อผิดพลาดในการเขียนโปรแกรมและการทดสอบ ที่อาจเกิดขึ้นและเปรียบเทียบกับสิ่งที่เกิดขึ้นจริง ได้แก่
- มีการคาดการณ์ว่าการเขียนโปรแกรมไม่มีโครงสร้างและมาตรฐาน
 - มีการคาดการณ์ว่าการเขียนโปรแกรมไม่ตรงกับรายละเอียดเอกสารการออกแบบ
 - มีการคาดการณ์การว่าการเขียนโปรแกรมและการเปลี่ยนแปลงแก้ไขอยู่นอกเหนือขอบเขต
 - มีการคาดการณ์ข้อผิดพลาดในเอกสารรายละเอียดโปรแกรม
 - มีการคาดการณ์ข้อผิดพลาดในรายละเอียดโปรแกรม ได้แก่
 - การใช้คำสั่งผิด
 - การใช้ไวยากรณ์ผิด
 - การใช้ลอจิกผิด
 - การใช้ตัวแปรผิด
 - การใช้ตัวแปรผิดประเภทกับข้อมูล (Type Mismatch)
 - ข้อผิดพลาดในการตรวจสอบข้อมูลนำเข้า
 - ข้อผิดพลาดในการส่งผ่านข้อมูลระหว่างรูทีน
 - ข้อผิดพลาดในการตรวจสอบข้อมูลผลลัพธ์
 - ข้อผิดพลาดในการคำนวณและประมวลผล ฯลฯ

- มีการคาดการณ์ข้อผิดพลาดในการติดต่อเชื่อมโยงโปรแกรม
 - มีการคาดการณ์ข้อผิดพลาดเกี่ยวกับฐานข้อมูลที่ใช้ในโปรแกรม
 - มีการคาดการณ์ข้อผิดพลาดในการทดสอบโปรแกรม
 - มีการคาดการณ์ข้อผิดพลาดในกรณีศึกษาที่ใช้ในการทดสอบ
 - มีการคาดการณ์ข้อผิดพลาดของข้อมูลที่ใช้ทดสอบ
 - มีการจัดทำรายงานสรุปผลการคาดการณ์ข้อผิดพลาดที่อาจเกิดขึ้นเปรียบเทียบกับที่เกิดขึ้นจริง
- มีการเปรียบเทียบการใช้ประโยชน์จากหน่วยความจำคอมพิวเตอร์ (Computer Memory Utilization) ที่ได้ตั้งเป้าไว้
- มีการจัดทำเอกสารเปรียบเทียบปริมาณหน่วยความจำที่ต้องการใช้ในการประมวลผล ที่ได้ตั้งเป้าไว้และที่เกิดขึ้นจริง
 - มีการจัดทำเอกสารเปรียบเทียบปริมาณหน่วยความจำที่ต้องการใช้ทางอ้อม ที่ได้ตั้งเป้าไว้และที่เกิดขึ้นจริง ประกอบด้วย
 - Dynamic and Static Memory Work Areas
 - Memory Resident Tables
 - Buffers
 - Stack Size
 - มีการจัดทำเอกสารเปรียบเทียบความเร็วในการทำงานของหน่วยความจำ (Refresh Rate) ที่ได้ตั้งเป้าไว้และที่เกิดขึ้นจริง
 - มีการจัดทำเอกสารเปรียบเทียบโอกาสของการพบข้อมูลบนแรม (Hit Ratio) ที่ได้ตั้งเป้าไว้และที่เกิดขึ้นจริง
 - มีการจัดทำเอกสารเปรียบเทียบการนำหน่วยความจำแคช (RAM Cash) มาช่วยเพิ่มประสิทธิภาพ ที่ได้ตั้งเป้าไว้และที่เกิดขึ้นจริง
- มีการเปรียบเทียบปริมาณงานที่ทำโดยระบบคอมพิวเตอร์ (Computer Throughput Utilization) ที่ได้ตั้งเป้าไว้ และที่เกิดขึ้นจริง
- มีการจัดทำเอกสารเปรียบเทียบปริมาณงานที่ทำได้ในช่วงเวลาหนึ่ง (Throughput) ที่ได้ตั้งเป้าไว้ และที่เกิดขึ้นจริง
 - มีการจัดทำเอกสารเปรียบเทียบงานย่อยต่างๆ ที่ประมวลผลโดยระบบคอมพิวเตอร์ (Workload) ที่ได้ตั้งเป้าไว้ และที่เกิดขึ้นจริงประกอบด้วย
 - ความต้องการในการนำเข้าและแสดงผลลัพธ์
 - ปริมาณและชนิดของการคำนวณ
 - ความต้องการใช้ทรัพยากรคอมพิวเตอร์

- มีการจัดทำเอกสารเปรียบเทียบเวลาที่ใช้ตั้งแต่ส่งงานเพื่อประมวลผลจนงานเสร็จ (Turnaround Time) ที่ได้ตั้งเป้าไว้ และที่เกิดขึ้นจริงใช้กรณี Batch Processing
- มีการจัดทำเอกสารเปรียบเทียบเวลาที่ใช้ตั้งแต่จบคำสั่งจนถึงการเริ่มต้นตอบสนองของระบบ (Response Time) ที่ได้ตั้งเป้าไว้ และที่เกิดขึ้นจริงใช้กรณี Interactive Processing

2.2.2 มีการจัดเก็บรวบรวมข้อมูลทางสถิติ เพื่อทำการวิเคราะห์และวัดผลกระบวนการในการพัฒนาซอฟต์แวร์ ในด้านต่าง ๆ ดังนี้

- มีการเก็บรวบรวมสถิติของข้อผิดพลาดที่เกิดขึ้นในการออกแบบซอฟต์แวร์ ได้แก่
 - มีการจัดทำเอกสารรวบรวมรายละเอียดประเภทของข้อผิดพลาด
 - มีการจัดทำเอกสารรวบรวมจำนวนครั้งของข้อผิดพลาดทั้งหมด
 - มีการจัดทำเอกสารรวบรวมจำนวนครั้งของข้อผิดพลาดแต่ละชนิดเปรียบเทียบกัน
 - มีการจัดทำเอกสารรวบรวมเปอร์เซ็นต์เฉลี่ยจำนวนครั้งของข้อผิดพลาดแต่ละชนิดเปรียบเทียบกับทั้งหมด
 - มีการจัดทำเอกสารรวบรวมเปอร์เซ็นต์เฉลี่ยจำนวนครั้งของข้อผิดพลาดในการออกแบบต่อโปรแกรม หรือต่อระบบงาน
 - มีการจัดทำเอกสารรวบรวมข้อผิดพลาดที่เกิดบ่อยที่สุด
 - มีการจัดทำเอกสารรวบรวมข้อผิดพลาดที่เกิดน้อยที่สุด
 - มีการจัดทำเอกสารบันทึกรายละเอียดข้อผิดพลาดในการออกแบบ (Design Error Logs) ประกอบด้วย
 - รายละเอียดข้อผิดพลาด
 - แหล่งที่พบ
 - วันที่พบ
 - ผู้รับผิดชอบในการแก้ไข
 - วันที่เริ่มต้น และสิ้นสุดการแก้ไข
 รายละเอียดการแก้ไข
 - มีการจัดทำเอกสารบันทึกรายละเอียดข้อผิดพลาดแจกแจงตามความสำคัญประกอบด้วย
 - ข้อผิดพลาดที่กระทบต่อแผนงาน
 - ข้อผิดพลาดที่กระทบต่อความต้องการ
 - ซอฟต์แวร์

- ข้อผิดพลาดที่กระทบต่องบประมาณ
 - ข้อผิดพลาดที่กระทบต่อความเชื่อถือของ
- มีการเก็บรวบรวมสถิติของข้อผิดพลาดที่เกิดขึ้นในการเขียนโปรแกรม และการทดสอบ
- มีการจัดทำเอกสารแสดงจำนวนครั้งของข้อผิดพลาดทั้งหมด
 - มีการจัดทำเอกสารแสดงจำนวนครั้งของข้อผิดพลาดแต่ละชนิดเปรียบเทียบกัน
 - มีการจัดทำเอกสารแสดงเปอร์เซ็นต์เฉลี่ยจำนวนครั้งของข้อผิดพลาดแต่ละชนิดเปรียบเทียบกับทั้งหมด
 - มีการจัดทำเปอร์เซ็นต์เฉลี่ยจำนวนครั้งของข้อผิดพลาดต่อโปรแกรม
 - มีการจัดทำเอกสารแสดงข้อผิดพลาดที่เกิดบ่อยที่สุด
 - มีการจัดทำเอกสารแสดงข้อผิดพลาดที่เกิดน้อยที่สุด
 - มีการจัดทำเอกสารแสดงรายละเอียดข้อผิดพลาดแจกแจงตามความสำคัญ ประกอบด้วย
 - ข้อผิดพลาดที่กระทบต่อแผนงาน
 - ข้อผิดพลาดที่กระทบต่องบประมาณ
 - ข้อผิดพลาดที่กระทบต่อความเชื่อถือของซอฟต์แวร์
- มีการวัดระดับและบันทึกเปอร์เซ็นต์ จำนวนบรรทัดของโปรแกรม และการออกแบบที่ถูกต้องตรวจสอบในขั้นตอนการออกแบบ และการเขียนโปรแกรม
- มีการวัดระดับและบันทึกเปอร์เซ็นต์ของคำสั่งโปรแกรมทั้งหมดที่ถูกประมวลในขั้นตอนการทดสอบ

2.2.3 มีการติดตามตรวจสอบกิจกรรมต่าง ๆ ในการพัฒนาซอฟต์แวร์ เพื่อทำการวิเคราะห์และวัดผลกระบวนการในด้านต่าง ๆ ดังนี้

- มีการติดตามตรวจสอบการใช้งานช่องทางการเข้าออกข้อมูลคอมพิวเตอร์ (Computer I/O Channel Utilization)
- มีเอกสารรายงานการติดตามตรวจสอบประสิทธิภาพของระบบบัสข้อมูล
 - มีเอกสารรายงานการติดตามตรวจสอบการนำเข้าข้อมูลผ่านทางอุปกรณ์นำเข้า (คีย์บอร์ด เม้าส์ , สแกนเนอร์ ฯลฯ) ประกอบด้วย
 - ปริมาณข้อมูลนำเข้า
 - ประสิทธิภาพ
 - การจัดการให้ใช้งานได้ประโยชน์สูงสุด
 - มีเอกสารรายงานการติดตามตรวจสอบการสื่อสารข้อมูลผ่านโมเด็มและอุปกรณ์สื่อสารต่าง ๆ

- มีเอกสารรายงานการติดตามตรวจสอบการแสดงผลข้อมูล ผ่านทางอุปกรณ์แสดงผล (จอภาพ เครื่องพิมพ์ ฯลฯ) ประกอบด้วย
 - ปริมาณข้อมูลนำเข้า
 - ประสิทธิภาพ
 - การจัดการให้ใช้งานได้ประโยชน์สูงสุด
- มีการติดตามรายละเอียดกิจกรรมต่างๆที่เป็นผลจากการตรวจสอบทบทวนในขั้นตอนการออกแบบอย่างใกล้ชิด
 - มีการแต่งตั้งผู้รับผิดชอบในการติดตามตรวจสอบ
 - มีการดำเนินการติดตามการนำข้อมูลการออกแบบไปเขียนโปรแกรม (กรณีการออกแบบถูกต้อง)
 - มีการดำเนินการติดตามการแก้ไขการออกแบบ และทบทวนความต้องการ (กรณีการออกแบบไม่ตรงตามความต้องการ)
 - มีการดำเนินการติดตามการแก้ไขการออกแบบ ในรายละเอียด (กรณีการออกแบบผิดพลาด) ประกอบด้วย
 - ข้อผิดพลาดที่พบ และสิ่งที่ต้องปรับปรุงในด้านต่างๆ ได้แก่
 - ความสมบูรณ์ถูกต้อง และเชื่อถือได้ (Reliability)
 - การนำไปใช้งาน
 - ความสะดวกในการบำรุงรักษา(Maintainability)
 - ความสะดวกในการเพิ่มเติมงาน (Extendability)
 - มีการจัดประชุมสรุปสถานะการติดตามผลการดำเนินงานภายหลังการติดตามการตรวจสอบ
 - มีการจัดทำรายงานสรุปผลการตรวจสอบเสนอต่อผู้จัดการซอฟต์แวร์เป็นประจำ
- มีการติดตามตรวจสอบรายงานปัญหาของซอฟต์แวร์ที่เกิดขึ้นจากการทดสอบระบบอย่างใกล้ชิด ได้แก่
 - มีการแต่งตั้งผู้รับผิดชอบในการติดตามตรวจสอบ และแก้ไขปัญหา
 - มีเอกสารบันทึกรายละเอียดปัญหาของซอฟต์แวร์ที่พบในการทดสอบระบบ (Error logs หรือ Software Problem Report) ประกอบด้วย
 - วันที่ เวลา ที่จัดทำรายงาน
 - ผู้จัดทำ
 - ประเภทของปัญหา

- แหล่งที่พบปัญหา (โปรแกรม, ฐานข้อมูล, กรณีศึกษาที่ใช้ทดสอบ, เอกสาร ฯลฯ)
 - รายละเอียดปัญหา และผลกระทบที่เกิดขึ้น
 - สถานะของการแก้ไขปัญหา
- มีการเก็บรวบรวมเอกสารรายงานปัญหาของซอฟต์แวร์ที่เกิดขึ้นไว้ที่ศูนย์กลาง
 - มีการจัดทำรายงานหรือประชุมสรุปผลการดำเนินงานแก้ไขปัญหามาเป็นช่วงเวลา
- มีการติดตามรายละเอียดกิจกรรมต่างๆ ที่เป็นผลจากการตรวจสอบทบทวนการเขียนโปรแกรมอย่างใกล้ชิด ได้แก่
- มีการแต่งตั้งผู้รับผิดชอบในการติดตามตรวจสอบ
 - มีการดำเนินการติดตามการนำโปรแกรมไปทำการรวบรวมระบบงานและทดสอบ (กรณีเขียนโปรแกรมถูกต้อง)
 - มีการดำเนินการติดตามการดำเนินการแก้ไขโปรแกรม และทบทวนการออกแบบ (กรณีการเขียนโปรแกรมไม่ตรงตามการออกแบบ หรือ ไม่สามารถทำได้)
 - มีการดำเนินการติดตามการดำเนินการแก้ไขโปรแกรมในรายละเอียดจากการทดสอบส่วนย่อย เช่นการนำเข้าข้อมูล การประมวลผล ผลลัพธ์ที่ได้
 - มีการจัดประชุมสรุปสถานะการติดตามผลการดำเนินงานภายหลังการติดตามการตรวจสอบ
 - มีการจัดทำรายงานสรุปผลการตรวจสอบเสนอต่อผู้จัดการซอฟต์แวร์เป็นประจำ
- มีการติดตามความก้าวหน้าของการทดสอบ (โดยเฉพาะส่วนของซอฟต์แวร์ที่ต้องส่งมอบ) เปรียบเทียบกับแผนงานที่ตั้งไว้ ได้แก่
- มีการแต่งตั้งผู้รับผิดชอบในการติดตามความก้าวหน้าการทดสอบ
 - มีการจัดทำเอกสารการติดตามความก้าวหน้าในการกำหนดขอบเขตรายละเอียดในการทดสอบ
 - มีการจัดทำเอกสารการติดตามความก้าวหน้าในทุกขั้นตอนของการทดสอบเปรียบเทียบกับแผนงาน ประกอบด้วย
 - การทดสอบส่วนย่อย (Unit Testing)
 - การทดสอบระบบย่อย (Subsystem testing)
 - การทดสอบระบบงานทั้งหมด (System testing)

- การทดสอบหาข้อผิดพลาดของระบบ (Regression testing)
- การทดสอบแบบอัลฟา (Alpha testing)
- การทดสอบแบบเบต้า (Beta testing)
- การทดสอบการยอมรับระบบงาน (Acceptance testing)
- มีการจัดทำเอกสารการติดตามความก้าวหน้าของการจัดทำข้อมูลทดสอบ
- มีการจัดทำเอกสารการติดตามความก้าวหน้าของการสรุปผลการทดสอบ

2.3 การจัดการข้อมูล (Data Management and Analysis) มีรายละเอียดเกี่ยวกับการจัดเก็บรวบรวมข้อมูล และการรักษาการวัดผลกระบวนการให้คงไว้ เพื่อให้มั่นใจว่าข้อมูลต่าง ๆ ได้รับความถูกต้อง และมีการจัดการอย่างมีประสิทธิภาพ องค์การซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็มจะต้องเตรียมดำเนินการ และความพร้อมในด้านต่าง ๆ ดังนี้คือ

2.3.1 มีการจัดเตรียมวิธีการหรือเทคนิค เพื่อใช้ในการดำเนินการในการจัดการข้อมูล และการวิเคราะห์ข้อมูลในด้านต่าง ๆ ดังนี้

- มีการจัดการและควบคุมฐานข้อมูลของกระบวนการเพื่อใช้ในการวัดผลประสิทธิภาพของกระบวนการต่าง ๆ ของโครงการ
- มีการกำหนดผู้รับผิดชอบการจัดการ และควบคุมฐานข้อมูลของกระบวนการ
- มีการจัดทำเอกสารเก็บรวบรวมข้อมูลต่าง ๆ ของกระบวนการไว้ที่ศูนย์กลางได้แก่
 - ข้อมูลการตรวจสอบทบทวน และประสิทธิภาพการตรวจสอบ
 - ข้อมูลการทดสอบต่าง ๆ และประสิทธิภาพการทดสอบ
 - ข้อมูลรายละเอียดต่าง ๆ ในแต่ละขั้นตอนกระบวนการ
 - ข้อมูลการวัดผลความเชื่อถือได้ของซอฟต์แวร์
 - ข้อมูลกิจกรรมการแก้ไข เปลี่ยนแปลงต่าง ๆ
 - ข้อมูลเอกสารที่ใช้ในแต่ละกระบวนการ
 - ข้อมูลเครื่องมือและวิธีการที่ใช้
 - ข้อมูลรายละเอียดผลิตภัณฑ์ที่พัฒนา
 - ปริมาณข้อผิดพลาดที่พบในแต่ละกระบวนการ

- มีการดำเนินการและควบคุมฐานข้อมูลโดยกลุ่มผู้ควบคุมกระบวนการ
 - มีการดำเนินการกำหนดสิทธิ์ของผู้ใช้ฐานข้อมูล
- มีวิธีการหรือเทคนิคใช้สำหรับวิเคราะห์สาเหตุของความผิดพลาดที่เกิดขึ้นในขั้นตอนการพัฒนาซอฟต์แวร์ (Error Cause Analysis)
- มีการกำหนดผู้รับผิดชอบในการวิเคราะห์หาสาเหตุความผิดพลาด
 - มีเอกสารการกำหนดขอบเขต ขั้นตอน วิธีการที่ใช้ในการวิเคราะห์หาสาเหตุความผิดพลาด
 - มีเอกสารการจำแนกกลุ่มของความผิดพลาด เป็นประเภทต่าง ๆ ดังนี้
 - ความผิดพลาดจากการคำนวณ
 - ความผิดพลาดในลอจิกโปรแกรม
 - ความผิดพลาดจากข้อมูลนำเข้า และผลลัพธ์
 - ความผิดพลาดในการติดต่อเชื่อมโยง
 - ความผิดพลาดในการจัดการข้อมูล (อ่าน เขียน เคลื่อนย้าย จัดเก็บ และปรับปรุงแก้ไข)
 - ความผิดพลาดจากระบบปฏิบัติการ หรือโปรแกรมช่วยสนับสนุนระบบงาน
 - ความผิดพลาดด้านขอบเขตโครงสร้างซอฟต์แวร์
 - การเปลี่ยนแปลงความต้องการของผู้ใช้
 - ความผิดพลาดจากฐานข้อมูล
 - ความผิดพลาดจากการใช้ตัวแปร
 - ความผิดพลาดที่เกิดขึ้นซ้ำ ๆ
 - ความผิดพลาดจากเอกสารที่ใช้
 - ความผิดพลาดเกิดจากความเข้าใจผิด (ในขั้นตอนสำรวจความต้องการไปยังการออกแบบ และต่อไปยังการเขียนโปรแกรม)
 - ความผิดพลาดที่ไม่สามารถคาดการณ์ได้
 - มีการดำเนินการวิเคราะห์ความผิดพลาดที่เกิดขึ้นในด้านต่าง ๆ ดังนี้
 - มีการวิเคราะห์รายละเอียดความผิดพลาด
 - มีการวิเคราะห์หาสาเหตุการเกิดความผิดพลาด
 - มีการวิเคราะห์แหล่งที่เกิดความผิดพลาด
 - มีการวิเคราะห์ปัจจัยที่ทำให้เกิดความผิดพลาด
 - มีการวิเคราะห์แนวทางในการป้องกัน และแก้ไขความผิดพลาด

- มีการจัดทำเอกสารรายละเอียดสาเหตุความผิดพลาด โดยทีมงานตรวจสอบ
 - มีการประชุมตรวจสอบ ทบทวน สาเหตุของความผิดพลาดระหว่างทีมงานพัฒนาซอฟต์แวร์ ทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์
- มีการตรวจสอบทบทวนสาเหตุของความผิดพลาด เพื่อพิจารณาการเปลี่ยนแปลงกระบวนการเพื่อป้องกันไม่ให้เกิดความผิดพลาดขึ้นอีก ได้แก่
- มีการกำหนดวัตถุประสงค์ของการตรวจสอบทบทวนสาเหตุของความผิดพลาด
 - มีการจัดทำเอกสารกำหนดขอบเขต ขั้นตอน วิธีการที่ใช้ในการตรวจสอบ ทบทวน
 - มีการแต่งตั้งผู้รับผิดชอบในการตรวจสอบทบทวน
 - มีการดำเนินการตรวจสอบทบทวน สาเหตุ ของความผิดพลาดที่มีผลกระทบต่อกระบวนการในด้านต่าง ๆ ดังนี้
 - มีการตรวจสอบเพื่อปรับปรุงคุณภาพซอฟต์แวร์
 - มีการตรวจสอบเพื่อเพิ่มประสิทธิภาพการดำเนินงาน
 - มีการตรวจสอบเพื่อลดเวลาในการพัฒนาผลิตภัณฑ์
 - มีการตรวจสอบเพื่อป้องกันไม่ให้เกิดความผิดพลาดขึ้นอีก
 - มีการจัดทำเอกสารรายละเอียดการทบทวน โดยผู้รับผิดชอบ
 - มีการประชุมตรวจสอบทบทวน สาเหตุของความผิดพลาดระหว่างทีมงานพัฒนาซอฟต์แวร์ ทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์
- มีวิธีการหรือเทคนิคที่ใช้ในการป้องกันความผิดพลาดที่อาจเกิดขึ้นในกระบวนการพัฒนาซอฟต์แวร์ตั้งแต่เริ่มต้น ได้แก่
- มีการจัดทำเอกสารการกำหนดขอบเขตของความผิดพลาดที่ยอมรับได้
 - มีการดำเนินการวิเคราะห์และประเมินความผิดพลาดที่เคยเกิดขึ้นในอดีต
 - มีการดำเนินการวิเคราะห์และประเมินความผิดพลาดที่อาจเกิดขึ้นในอนาคต
 - มีการแต่งตั้งผู้รับผิดชอบในการดำเนินการป้องกันความผิดพลาด
 - มีการจัดทำเอกสารแผนงานกิจกรรมป้องกันความผิดพลาด
 - มีการตรวจสอบการดำเนินงานทุกขั้นตอนในการพัฒนาซอฟต์แวร์เป็นประจำ
 - มีการจัดทำเอกสารสรุปความผิดพลาดจากการตรวจสอบ ประกอบด้วย

- เอกสารรายละเอียดความผิดพลาด
- เอกสารแนวทางแก้ไขความผิดพลาด
- เอกสารวิธีป้องกันความผิดพลาด

- มีการจัดประชุมสรุปสถานะโครงการเป็นประจำ

2.3.2 มีการวิเคราะห์ข้อมูลดำเนินการในขั้นตอนการพัฒนาซอฟต์แวร์ในด้านต่าง ๆ ได้แก่

- มีการวิเคราะห์ข้อมูลที่เก็บรวบรวมได้ในขั้นตอนการตรวจสอบทบทวนการออกแบบ ได้แก่

- มีการดำเนินการวิเคราะห์ข้อมูลที่เก็บรวบรวมได้ ดังนี้
 - ข้อมูลการตรวจสอบรายละเอียดการออกแบบ
 - ข้อมูลการตรวจสอบโครงสร้างข้อมูลและผังทางเดินข้อมูล
 - ข้อมูลการตรวจสอบโครงสร้างซอฟต์แวร์
 - ข้อมูลการตรวจสอบรายละเอียดระบบย่อยต่าง ๆ
 - ข้อมูลการตรวจสอบการติดต่อเชื่อมโยง
 - ข้อมูลการตรวจสอบความสัมพันธ์ระบบย่อยต่าง ๆ
- มีการดำเนินการวิเคราะห์ข้อมูลในด้านต่าง ๆ ได้แก่
 - เปรียบเทียบกับมาตรฐานและขอบเขต
 - ความถูกต้องชัดเจน
 - ต้นทุนและทรัพยากรที่ใช้
 - ความเชื่อถือได้ของการออกแบบ
 - ความซับซ้อนของการออกแบบ
- มีการจัดทำรายงานสรุปการวิเคราะห์ข้อมูล ในการตรวจสอบทบทวนการออกแบบ
- มีการวิเคราะห์ข้อมูลความผิดพลาดต่าง ๆ จากการตรวจสอบทบทวนโปรแกรมและการทดสอบ เพื่อใช้ตรวจสอบความผิดพลาดที่ยังคงเหลือในผลิตภัณฑ์ซอฟต์แวร์ ได้แก่
 - มีการดำเนินการวิเคราะห์ข้อมูลความผิดพลาดที่เก็บรวบรวมได้ ตามประเภทต่างดังนี้
 - ข้อมูลความผิดพลาดในรายละเอียดโปรแกรม
 - ข้อมูลความผิดพลาดในรายละเอียดระบบย่อย
 - ข้อมูลความผิดพลาดในการเปลี่ยนแปลงแก้ไขโปรแกรม
 - ข้อมูลความผิดพลาดในการติดต่อเชื่อมโยงโปรแกรม

- การเขียนโปรแกรมไม่เป็นไปตามมาตรฐานโครงสร้าง
 - การเขียนโปรแกรมไม่ได้อยู่ภายในขอบเขตที่กำหนด
 - การเขียนโปรแกรมไม่ตรงกับการออกแบบหรือไม่ตรงกับความต้องการ
- มีการดำเนินการวิเคราะห์ข้อมูลความผิดพลาดในด้านต่าง ๆ ได้แก่
 - การจำแนกประเภทของความผิดพลาด
 - แหล่งที่เกิดความผิดพลาด
 - ปริมาณความถี่ หรือจำนวนครั้ง ที่เกิดความผิดพลาด
 - สาเหตุของความผิดพลาด
 - ความรุนแรง และผลกระทบจากความผิดพลาด
 - การป้องกันความผิดพลาด
 - การหาแนวทางแก้ไขความผิดพลาด
 - มีการจัดทำรายงานสรุปผลการวิเคราะห์ความผิดพลาดด้านต่าง ๆ
- มีการวิเคราะห์ความผิดพลาดที่เกิดขึ้น เพื่อหาสาเหตุที่เกี่ยวข้องกับกระบวนการซอฟต์แวร์ได้แก่
- มีการดำเนินการวิเคราะห์ความผิดพลาดจากการตรวจสอบคุณภาพซอฟต์แวร์
 - มีการดำเนินการวิเคราะห์ความผิดพลาดจากการบริหารขอบเขตโครงสร้างซอฟต์แวร์
 - มีการดำเนินการวิเคราะห์ความผิดพลาดจากการทดสอบซอฟต์แวร์
 - มีการดำเนินการวิเคราะห์ความผิดพลาดจากการสนับสนุนงานด้านเอกสาร เป็นต้น
 - มีการจัดทำรายงานสรุปผลการวิเคราะห์ความผิดพลาดด้านต่าง ๆ
- มีการวิเคราะห์ประสิทธิภาพของการตรวจสอบทบทวน (Review Efficiency Analyzed) ในแต่ละโครงการได้แก่
- มีการดำเนินการวิเคราะห์เปอร์เซ็นต์ความผิดพลาดที่พบในขั้นตอนการตรวจสอบทบทวน และการทดสอบ มีรายละเอียดดังนี้
 - คำนวณจากปริมาณข้อผิดพลาดที่พบในขั้นตอนการตรวจสอบทบทวน หาดด้วยข้อผิดพลาดทั้งหมดที่พบในการตรวจสอบทบทวนและการทดสอบ
 - ข้อผิดพลาดทั้งหมด คือ ข้อผิดพลาดที่พบในระหว่างขั้นตอนการพัฒนาผลิตภัณฑ์และการทดสอบการรวบรวมระบบ

งาน (แต่ไม่รวมการทดสอบการยอมรับระบบ หรือการทดสอบการใช้งาน)

- มีการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในแต่ละขั้นตอนของกระบวนการที่ใช้ในการพัฒนาซอฟต์แวร์ได้แก่
- มีการกำหนดผู้รับผิดชอบในการวิเคราะห์ประสิทธิภาพซอฟต์แวร์
- มีการดำเนินการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในแต่ละขั้นตอนของกระบวนการหลักที่ใช้ในการพัฒนาซอฟต์แวร์ โดยทีมงานผู้รับผิดชอบ ดังนี้ คือ
 - มีการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในขั้นตอนแนวคิดเบื้องต้น
 - มีการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในขั้นตอนการกำหนดรายละเอียดความต้องการซอฟต์แวร์
 - มีการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในขั้นตอนการออกแบบ
 - มีการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในขั้นตอนการนำซอฟต์แวร์ไปใช้งาน
 - มีการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในขั้นตอนการรวบรวมระบบงานซอฟต์แวร์และการทดสอบ
 - มีการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในขั้นตอนการบำรุงรักษาระบบงาน
- มีการจัดทำเอกสารการกำหนดขอบเขต วิธีการ และนโยบายที่ใช้ในการวิเคราะห์ประสิทธิภาพ
- มีการดำเนินการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ในด้านต่าง ๆ ดังนี้ คือ
 - มีการวิเคราะห์และประเมินผลทางสถิติ
 - มีการวิเคราะห์เปรียบเทียบกับขอบเขต มาตรฐาน แผนงาน
 - มีการวิเคราะห์เปรียบเทียบกับความต้องการซอฟต์แวร์
 - มีการวิเคราะห์การใช้ทรัพยากร แรงงาน ต้นทุน เวลา เปรียบเทียบกับแผนงาน
 - มีการวิเคราะห์ปริมาณ และความรุนแรงของความผิดพลาดที่เกิดขึ้น
 - มีการวิเคราะห์ความถูกต้องเชื่อถือได้ของซอฟต์แวร์
- มีการจัดทำรายงานและเอกสารบันทึกผลการวิเคราะห์ประสิทธิภาพของซอฟต์แวร์เป็นลายลักษณ์อักษร

2.4 การควบคุมกระบวนการ (Process Control) เกี่ยวข้องกับการกำหนดรายละเอียดกระบวนการพัฒนาซอฟต์แวร์ และวิธีการหรือเทคนิคที่ใช้ในการระบุถึงปัญหาต่าง ๆ ของกระบวนการ การแก้

ไขข้อมูลบกพร่องของกระบวนการ และการป้องกันไม่ให้เกิดปัญหาขึ้นอีก องค์กรซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็มจะต้องเตรียมดำเนินการและความพร้อมในด้านต่าง ๆ ดังนี้

2.4.1 มีการดำเนินการเพื่อควบคุมกระบวนการพัฒนาซอฟต์แวร์ ได้แก่

- มีการตรวจสอบ อนุมัติ ลงนาม ในตารางเวลาการทำงาน และการประมาณต้นทุนในแต่ละส่วนโดยผู้จัดการโครงการพัฒนาซอฟต์แวร์ระดับต้น ที่รับผิดชอบโดยตรง
- มีขั้นตอนกระบวนการในการจัดการอย่างเป็นทางการ เพื่อประเมินว่าแบบจำลองหน้าที่ของซอฟต์แวร์ (Software Function Prototyping) เป็นส่วนที่เหมาะสมของกระบวนการออกแบบ ได้แก่
 - มีการจัดทำเอกสารการกำหนดขอบเขต รายละเอียด วิธีการใช้แบบจำลองหน้าที่ของซอฟต์แวร์
 - มีการจัดทำเอกสารกำหนดมาตรฐานของแบบจำลองหน้าที่ของซอฟต์แวร์
 - มีการฝึกอบรมให้ความรู้ และฝึกฝนปฏิบัติในการใช้แบบจำลอง
 - มีการกำหนดกลุ่มผู้รับผิดชอบในการประเมินแบบจำลอง
 - มีการดำเนินการวิเคราะห์เปรียบเทียบการใช้แบบจำลองในด้านต่าง ๆ ดังนี้ คือ
 - รายละเอียดความชัดเจนในการสื่อความหมาย
 - ประโยชน์ใช้สอยในการออกแบบ
 - มีการจัดทำรายงานสรุปผลการวิเคราะห์ และประเมิน
- มีการดำเนินการตรวจสอบทบทวนการออกแบบภายในซอฟต์แวร์ (Internal Software Design Review) ได้แก่
 - มีการกำหนดทีมงานรับผิดชอบในการตรวจสอบทบทวน
 - มีการดำเนินการตรวจสอบรายละเอียดการออกแบบ ประกอบด้วย
 - การตรวจสอบเอกสารขอบเขต และรายละเอียดการออกแบบ
 - การตรวจสอบเอกสารรายละเอียดการออกแบบเปรียบเทียบกับความต้องการซอฟต์แวร์ (Requirement Cross Reference)
 - การตรวจสอบเอกสารโครงสร้างข้อมูล และผังทางเดินข้อมูล
 - การตรวจสอบเอกสารโครงสร้างซอฟต์แวร์
 - การตรวจสอบเอกสารรายละเอียดระบบย่อยต่าง ๆ
 - การตรวจสอบเอกสารรายละเอียดการติดต่อเชื่อมโยง

- มีการดำเนินการตรวจสอบประสิทธิภาพของการแบ่งขนาดการออกแบบเป็นระบบย่อยๆ (Design Breakdown) โดยพิจารณาจากผลการดำเนินงาน ขนาดของระบบ ปริมาณโปรแกรม ความซับซ้อน ฯลฯ
 - มีการดำเนินการตรวจสอบความสัมพันธ์ระหว่างระบบย่อยต่างๆที่ออกแบบ (Connections Between Modules)
 - มีการประชุมสรุปผลการตรวจสอบตามช่วงเวลา ระหว่างทีมงานออกแบบ ทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์
 - มีการจัดทำรายงานสรุปผลการตรวจสอบให้กับผู้จัดการซอฟต์แวร์เป็นประจำ
- มีการเก็บบันทึกรายละเอียดความก้าวหน้าของการพัฒนาระบบงานซอฟต์แวร์ในแต่ละส่วนย่อยๆ อย่างเป็นทางการได้แก่
- มีการจัดทำเอกสารรายละเอียดความก้าวหน้าเป็นลายลักษณ์อักษร
 - มีการปรับปรุงเอกสารแผนงานตารางเวลาการพัฒนาซอฟต์แวร์ตามช่วงเวลา
 - มีการลงนามอนุมัติ ชื่อผู้จัดทำ และชื่อผู้ตรวจสอบ ในเอกสารแผนงานตารางเวลา
 - มีการประชุมสรุปความก้าวหน้าของโครงการเป็นประจำ
 - ระหว่างทีมงานพัฒนา และผู้จัดการซอฟต์แวร์
 - มีการจัดทำรายงานสรุปผลความก้าวหน้าของโครงการ
 - เสนอให้กับผู้จัดการซอฟต์แวร์ เป็นประจำ
- มีการดำเนินการตรวจสอบทบทวนการเขียนโปรแกรมซอฟต์แวร์ (Software Code Reviews) ได้แก่
- มีการกำหนดผู้รับผิดชอบในการตรวจสอบทบทวน
 - มีการจัดทำเอกสารการกำหนดวัตถุประสงค์ของการตรวจสอบทบทวน
 - มีการจัดทำเอกสารการกำหนดขอบเขต ขั้นตอนการตรวจสอบทบทวน
 - มีการดำเนินการตรวจสอบทบทวนการเขียนโปรแกรมในรายละเอียดต่างๆ ดังนี้คือ
 - ตรวจสอบโปรแกรมเปรียบเทียบกับเอกสารการออกแบบ
 - ตรวจสอบรายละเอียดระบบย่อย (Module Specification) เปรียบเทียบกับเอกสารการออกแบบ
 - ตรวจสอบเอกสารการเปลี่ยนแปลงโปรแกรม (Change Documentation)
 - ตรวจสอบข้อมูลที่ใช้ทดสอบ (Test Data)

- มีการดำเนินการใช้วิธีการตรวจสอบทบทวนการเขียนโปรแกรม (อย่างใดอย่างหนึ่ง) ดังนี้คือ
 - ตรวจสอบทบทวนโดยคณะกรรมการ (Peer Review)
 - ตรวจสอบทบทวนอย่างไม่เป็นทางการ (Code Walkthrough)
 - ตรวจสอบทบทวนอย่างเป็นทางการ (Code Inspection)
 - มีการประชุมสรุปผลการตรวจสอบตามช่วงเวลา ระหว่างทีมงานเขียนโปรแกรม ทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์
 - มีการจัดทำรายงานสรุปผลการตรวจสอบให้กับผู้จัดการซอฟต์แวร์เป็นประจำ
- มีการดำเนินการตรวจสอบทบทวนกรณีศึกษาต่างๆที่ใช้ในการทดสอบอย่างเป็นทางการ (Formal Test Case Reviews) ได้แก่
- มีการจัดทำเอกสารกำหนดขอบเขต กรณีศึกษาต่างๆที่ใช้ในการทดสอบ
 - มีการจัดทำเอกสารการกำหนดขั้นตอน และนโยบายที่ใช้ในการตรวจสอบทบทวน
 - มีการกำหนดผู้รับผิดชอบในการตรวจสอบทบทวน
 - มีเอกสารการกำหนดปริมาณและประเภทของกรณีทดสอบแบบต่างๆ
 - มีการดำเนินการตรวจสอบทบทวนกรณีศึกษาต่างๆที่ใช้ในการทดสอบในด้านต่างๆดังนี้
 - ปริมาณและประเภทของกรณีศึกษาต่างๆ
 - ความสัมพันธ์ที่เป็นผลกระทบกับระบบย่อยอื่นๆ
 - การเป็นตัวแทนของการดำเนินงานจริง
 - มีการดำเนินการปรับปรุงพัฒนากรณีศึกษาใหม่ๆแทนกรณีศึกษาเดิมที่ล้าสมัย
 - มีการประชุมสรุปผลการตรวจสอบ โดยทีมงานพัฒนา ทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์
 - มีการจัดทำเอกสารสรุป และบันทึกผลการตรวจสอบทบทวนกรณีศึกษาที่ใช้ทดสอบ

2.4.2 มีวิธีการหรือเทคนิคในการควบคุมกระบวนการ เพื่อใช้ในการระบุถึงปัญหาต่างๆ ของกระบวนการ การแก้ไขข้อมูลบกพร่องของกระบวนการ และการป้องกันไม่ให้เกิดปัญหาขึ้นอีก ได้แก่

- มีวิธีการ หรือเทคนิคเพื่อให้ผู้จัดการอาวุโสใช้ในการตรวจสอบทบทวนสถานะของโครงการพัฒนาซอฟต์แวร์อย่างสม่ำเสมอได้แก่

- มีการจัดทำเอกสารรายงานสถานะของการทำงานตามช่วงเวลาเป็นลายลักษณ์อักษร
 - มีการรายงานสถานะการทำงานแบบปากเปล่า เพื่อให้ทีมงานพัฒนา รายงานผลการดำเนินงานให้กับผู้จัดการโดยตรง
 - มีการประชุมสถานะของโครงการ ระหว่างทีมงานพัฒนาซอฟต์แวร์ และผู้จัดการอาวุโส
 - มีการสาธิตผลิตภัณฑ์ที่กำลังพัฒนา หรือรายงานผลการดำเนินงานตามช่วงเวลา ให้กับผู้จัดการอาวุโส
 - มีการติดตามตรวจสอบการดำเนินงานผ่านหน้าจอคอมพิวเตอร์ โดยผู้จัดการอาวุโส
 - มีการเผยแพร่ข้อมูลหรือการได้รับข้อมูลข่าวสารโครงการ ผ่านทางจดหมายอิเล็กทรอนิกส์ ระหว่างทีมงานพัฒนา และผู้จัดการโครงการ
 - มีการปรับปรุงเอกสารตารางเวลาการทำงานให้ทันสมัยอยู่เสมอ และส่งให้กับผู้จัดการ เป็นประจำ
- มีวิธีการ หรือเทคนิคที่ใช้ในการประเมินกระบวนการด้านวิศวกรรมซอฟต์แวร์ตามช่วงเวลา เพื่อปรับปรุงพัฒนากระบวนการได้แก่
- มีการกำหนดผู้รับผิดชอบในการประเมินกระบวนการ
 - มีเอกสารการกำหนดขอบเขต วิธีการ และนโยบายที่ใช้ในการประเมิน
 - มีเอกสารการกำหนดมาตรฐานที่ใช้ในการประเมินกระบวนการ
 - มีการกำหนดเอกสารแผนงานในการประเมินกระบวนการตามช่วงเวลา
 - มีการดำเนินการประเมินกระบวนการในด้านต่าง ๆ (ประสิทธิภาพ ต้นทุน เวลา ทรัพยากรที่ใช้)เปรียบเทียบกับมาตรฐานที่ตั้งไว้
 - มีการสรุปวิเคราะห์ และบันทึกผลการประเมินกระบวนการเป็นลายลักษณ์อักษร
 - มีการจัดเอกสารลำดับความสำคัญของสิ่งที่ต้องปรับปรุงพัฒนา
 - มีการวางแผนและกำหนดกลยุทธ์ที่จะใช้ในการปรับปรุงพัฒนากระบวนการ
- มีวิธีการหรือเทคนิคที่ใช้ในการระบุ ดัดสิน และขจัดปัจจัยต่างๆ ทางด้านวิศวกรรมซอฟต์แวร์ที่มีผลกระทบกับซอฟต์แวร์ได้แก่
- มีเอกสารการระบุแจกแจงปัจจัยมาตรฐานที่มีผลกระทบต่อซอฟต์แวร์ได้แก่
 - ปัจจัยที่มีผลกระทบต่อแผนงาน
 - ปัจจัยที่มีผลกระทบต่อขั้นตอนการพัฒนาซอฟต์แวร์

- ปัจจัยที่มีผลกระทบต่องบประมาณ
 - ปัจจัยที่มีผลกระทบต่อความเชื่อถือ และประสิทธิภาพซอฟต์แวร์
- มีการจัดตั้งผู้รับผิดชอบในการควบคุมดูแลปัจจัยดังกล่าว (Software Process Group)
 - มีการดำเนินการติดตามตรวจสอบ และรายงานผลปัจจัยต่างๆ เป็นประจำ
 - มีการดำเนินการแก้ไขทันทีที่มีปัญหาเกิดขึ้น
- มีวิธีการหรือเทคนิคที่ทำให้ผู้จัดการโครงการสนใจดูแลปัจจัยในการรวบรวมระบบงาน และการทดสอบได้แก่
 - มีการชี้แจงความสำคัญของปัจจัยต่างๆ ในการรวบรวมระบบงาน และการทดสอบที่อาจมีผลกระทบต่อซอฟต์แวร์ ให้กับผู้จัดการโครงการ ได้แก่
 - ปัจจัยที่มีผลกระทบต่อแผนงาน
 - ปัจจัยที่มีผลกระทบต่อขั้นตอนการพัฒนาซอฟต์แวร์
 - ปัจจัยที่มีผลกระทบต่องบประมาณ
 - ปัจจัยที่มีผลกระทบต่อความเชื่อถือ และประสิทธิภาพซอฟต์แวร์
 - มีการดำเนินการควบคุมดูแลปัจจัยต่างๆ ทางด้านวิศวกรรมซอฟต์แวร์ เป็นประจำ โดยผู้รับผิดชอบในการควบคุมดูแล
 - มีการจัดประชุมสรุปสถานะการดำเนินงานโครงการเป็นประจำ
 - มีการจัดทำรายงานสรุปผลเสนอต่อผู้จัดการซอฟต์แวร์ เป็นประจำ
 - มีวิธีการหรือเทคนิคที่ใช้สำหรับการแลกเปลี่ยนทางด้านเทคนิคระหว่างผู้พัฒนาซอฟต์แวร์กับผู้ใช้เป็นประจำได้แก่
 - มีการแนะนำเทคนิคต่างๆ ให้กับผู้ใช้งานโดยตรง
 - มีการจัดทำคู่มือการใช้งาน เพื่อแนะนำเทคนิคใหม่เป็นประจำ
 - มีการจัดประชุมสัมมนา ฝึกอบรม
 - มีการเผยแพร่ข้อมูลต่างๆ ทางด้านเทคนิคโดยการจัดทำวารสาร จดหมายอิเล็กทรอนิกส์ ติดประกาศ ฯลฯ
 - มีวิธีการหรือเทคนิคเพื่อตรวจสอบว่ามีการยอมรับและปฏิบัติตามมาตรฐานของวิศวกรรมซอฟต์แวร์ ได้แก่
 - มีการดำเนินการติดตามตรวจสอบผลการดำเนินการแต่ละขั้นตอนในการพัฒนาซอฟต์แวร์อยู่เป็นประจำ ประกอบด้วย
 - ความสำเร็จของงาน

- งบประมาณ และทรัพยากรที่ใช้
 - ความถูกต้อง และประสิทธิภาพที่ได้
 - ผลของการให้ความร่วมมือ
- มีการเผยแพร่ข้อมูล และประโยชน์ของมาตรฐานของวิศวกรรมซอฟต์แวร์ในองค์กรเป็นประจำ ได้แก่
- การตีตประกาศ
 - การเผยแพร่ข้อมูลผ่านระบบคอมพิวเตอร์เครือข่าย
 - การจัดทำวารสารของโครงการ
- มีการสำรวจความคิดเห็น หรือการตรวจสอบอย่างไม่เป็นทางการเกี่ยวกับการดำเนินการตามมาตรฐานของวิศวกรรมซอฟต์แวร์
 - มีการประชุมสรุปสถานะโครงการเป็นประจำ
 - มีการจัดทำสรุปรายงานการตรวจสอบให้กับผู้จัดการซอฟต์แวร์เป็นประจำ
- มีวิธีการ หรือเทคนิคที่ใช้ในการติดตามตรวจสอบระดับของความสัมพันธ์ในการดำเนินการระหว่างขั้นตอน (Traceability) ได้แก่
- มีการแต่งตั้งผู้รับผิดชอบในการติดตามตรวจสอบ
 - มีการดำเนินการตรวจสอบทบทวนการออกแบบขั้นต้น (Preliminary Design Review – PDR) โดยผู้รับผิดชอบ ได้แก่
 - มีการตรวจสอบเอกสารการออกแบบขั้นต้นเปรียบเทียบกับรายละเอียดความต้องการ
 - มีการจัดทำเอกสารสรุปรายละเอียดการออกแบบอ้างอิงเอกสารรายละเอียดความต้องการ (Cross References)
 - มีการลงนามอนุมัติในเอกสารการออกแบบ โดยผู้ตรวจสอบ ผู้จัดการซอฟต์แวร์ และผู้ใช้
 - มีการประชุมสรุปการทำงานตามช่วงเวลา ระหว่างทีมงานพัฒนา และผู้ใช้
 - มีขั้นตอน แบบฟอร์มการขออนุมัติการเปลี่ยนแปลงแก้ไขการออกแบบ
 - มีการจัดทำรายงานสรุปผลการตรวจสอบให้กับผู้จัดการซอฟต์แวร์เป็นประจำ
- มีวิธีการหรือเทคนิคที่ใช้ในการควบคุมการเปลี่ยนแปลงความต้องการซอฟต์แวร์ ได้แก่
- มีการจัดตั้งผู้ที่มีหน้าที่ควบคุมขอบเขตโครงร่างของซอฟต์แวร์ (Software Configuration Control)

- มีการจัดทำเอกสารขอบเขตรายละเอียดความต้องการซอฟต์แวร์ (Software Requirements Specification)
 - มีการอนุมัติ และลงนามในสรุปรายละเอียดความต้องการโดยผู้ใช้
 - มีขั้นตอน แบบฟอร์ม ในการขออนุมัติการเปลี่ยนแปลงความต้องการซอฟต์แวร์
 - มีการลงนามอนุมัติการเปลี่ยนแปลงแก้ไข หรือ เพิ่มเติมความต้องการซอฟต์แวร์ทุกครั้งก่อนดำเนินการ โดยทีมงาน SCC
 - มีเอกสารบันทึกรายละเอียดการเปลี่ยนแปลงความต้องการ และผลกระทบต่างๆ
- มีวิธีการหรือเทคนิคที่ใช้ในการติดตามตรวจสอบระดับของความสัมพันธ์ในการดำเนินการระหว่างขั้นตอน การออกแบบขั้นต้น และการออกแบบรายละเอียด (Traceability) ได้แก่
- มีการกำหนดผู้รับผิดชอบในการติดตามตรวจสอบ
 - มีการดำเนินการตรวจสอบทบทวนการออกแบบรายละเอียด (Detailed Design Reviews - DDR) โดยผู้รับผิดชอบ ได้แก่
 - มีการตรวจสอบทบทวนเอกสารการออกแบบรายละเอียดเปรียบเทียบกับออกแบบขั้นต้น โดยวิธีการอย่างใดอย่างหนึ่งดังนี้
 - การตรวจสอบโดยไม่เป็นทางการ (Informal Walkthrough)
 - การตรวจสอบอย่างเป็นทางการ (Structured Walkthrough)
 - การตรวจสอบโดยผู้เชี่ยวชาญอย่างเป็นทางการ (Design Inspections)
 - มีการจัดทำเอกสารสรุปรายละเอียดการออกแบบอ้างอิง เอกสารการออกแบบขั้นต้น (Cross References)
 - มีขั้นตอน แบบฟอร์มการอนุมัติการเปลี่ยนแปลงแก้ไขการออกแบบ
 - มีการลงนามอนุมัติในเอกสารการออกแบบ โดยผู้ตรวจสอบ และผู้จัดการซอฟต์แวร์
 - มีการประชุมสรุปผลการตรวจสอบตามช่วงเวลา ระหว่างทีมงานออกแบบ ทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์

- มีการจัดทำรายงานสรุปผลการตรวจสอบให้กับผู้จัดการซอฟต์แวร์เป็นประจำ
- มีวิธีการหรือเทคนิคที่ใช้ในการควบคุมการเปลี่ยนแปลงแก้ไขการออกแบบซอฟต์แวร์ได้แก่
 - มีการจัดทำเอกสารกำหนดขอบเขต ขั้นตอน และวิธีการที่ใช้
 - มีการแต่งตั้งผู้ที่มีหน้าที่รับผิดชอบควบคุม SCC.
 - มีการดำเนินการควบคุมการเปลี่ยนแปลงแก้ไขให้อยู่ภายใต้ขอบเขตที่กำหนดตามเอกสาร โดยทีมงาน SCC
 - มีขั้นตอน แบบฟอร์มการขออนุมัติการเปลี่ยนแปลงแก้ไข
 - มีการบันทึกรายละเอียด การเปลี่ยนแปลงแก้ไขการออกแบบซอฟต์แวร์
 - มีการตรวจสอบทบทวน ก่อน และหลังการเปลี่ยนแปลง
 - แก้ไขการออกแบบ และจัดทำเป็นเอกสาร
 - มีการปรับปรุงเอกสารโปรแกรม และรายละเอียดอื่นที่เกี่ยวข้อง
 - มีการจัดทำรายงานสรุปผลการเปลี่ยนแปลง
- มีวิธีการหรือเทคนิคที่ใช้ในการติดตามตรวจสอบระดับของความสัมพันธ์ในการดำเนินการระหว่างขั้นตอน การออกแบบรายละเอียดซอฟต์แวร์ และการเขียนโปรแกรม (Traceability) ได้แก่
 - มีการแต่งตั้งผู้รับผิดชอบในการติดตามตรวจสอบ
 - มีการดำเนินการตรวจสอบทบทวนการเขียนโปรแกรม โดยผู้รับผิดชอบได้แก่
 - มีการดำเนินการตรวจสอบรายละเอียดการเขียนโปรแกรม เปรียบเทียบกับเอกสารการออกแบบโดยใช้วิธีการอย่างใดอย่างหนึ่งดังนี้
 - Peer Review
 - Code Inspection
 - Code Walkthrough
 - Static Analysis
 - Metrics
 - มีการจัดทำเอกสารสรุปโปรแกรมอ้างอิงเอกสารการออกแบบ (Cross References)
 - มีการลงนามอนุมัติในเอกสารรายละเอียดการจัดทำโปรแกรม
 - มีขั้นตอน แบบฟอร์มการอนุมัติการเปลี่ยนแปลงแก้ไข

- โปรแกรม โดยผู้ตรวจสอบ และผู้จัดการซอฟต์แวร์
 - มีการประชุมสรุปผลการตรวจสอบตามช่วงเวลา ระหว่างทีมงานออกแบบ ทีมงานตรวจสอบ และผู้จัดการซอฟต์แวร์
 - มีการจัดทำรายงานสรุปผลการตรวจสอบให้กับผู้จัดการซอฟต์แวร์เป็นประจำ
- มีวิธีการหรือเทคนิคที่ใช้ในการควบคุมการเปลี่ยนแปลงแก้ไขโปรแกรม ได้แก่
- มีการจัดทำเอกสารขอบเขตรายละเอียดของโปรแกรมต่างๆ
 - มีการกำหนดสิทธิ์ระดับต่างๆ (Security) ในการเปลี่ยนแปลงแก้ไขโปรแกรม โครงสร้างข้อมูล โครงสร้างไฟล์ ฯลฯ
 - มีการจัดตั้งผู้ที่มีหน้าที่ควบคุมขอบเขตโครงสร้างของซอฟต์แวร์
 - มีขั้นตอน แบบฟอร์ม ในการขออนุมัติการเปลี่ยนแปลง
 - แก้ไขโปรแกรม
 - มีการลงนามอนุมัติการเปลี่ยนแปลงแก้ไขหรือเพิ่มเติมโปรแกรมทุกครั้ง ก่อนดำเนินการ
 - มีการจัดทำเอกสารบันทึกรายละเอียดการเปลี่ยนแปลง
 - แก้ไขโปรแกรม ประกอบด้วย
 - วัน เวลา ที่ดำเนินการ
 - ประเภทของการดำเนินการ (เพิ่มเติมเปลี่ยนแปลง แก้ไข)
 - รายละเอียดการเปลี่ยนแปลงแก้ไข
 - ผู้รับผิดชอบ หรือ ผู้ดำเนินการ
 - ผู้อนุมัติ
 - เวลา และทรัพยากรที่ใช้
 - มีการเก็บสำรองโปรแกรมก่อน และหลังการเปลี่ยนแปลงแก้ไข ไว้ในสื่อที่ใช้ในการเก็บข้อมูล เช่น ฮาร์ดดิสก์ เทป ดิสเก็ตต์
- มีวิธีการหรือเทคนิคที่ใช้ในการกำหนดคุณสมบัติหรือรายละเอียดของเครื่องมือด้านซอฟต์แวร์ (Software Tools) ได้แก่
- มีการกำหนดผู้รับผิดชอบในการกำหนดคุณสมบัติ
 - มีการจัดทำเอกสารการกำหนดขั้นตอนในการพิจารณาการนำเครื่องมือด้านซอฟต์แวร์มาใช้
 - มีการจัดทำเอกสารการกำหนดคุณสมบัติของเครื่องมือโดยพิจารณารายละเอียดในด้านต่างๆ ดังนี้คือ

- ความต้องการใช้เครื่องมือด้านซอฟต์แวร์ในแต่ละขั้นตอนการพัฒนา
 - ประโยชน์ใช้สอยของเครื่องมือแต่ละชนิด
 - ความสะดวกสบายในการใช้งาน
 - ความทันสมัยเปรียบเทียบกับเทคโนโลยีภายนอก
 - ความสามารถในการช่วยเพิ่มประสิทธิภาพงาน
- มีขั้นตอน แบบฟอร์ม การขออนุมัติการใช้งาน การเปลี่ยนแปลง และการยกเลิกเครื่องมือด้านซอฟต์แวร์
 - มีการจัดทำรายงานสรุปคุณสมบัติหรือรายละเอียดของเครื่องมือด้านซอฟต์แวร์
- มีวิธีการหรือเทคนิคที่ใช้ในการตรวจสอบว่าตัวอย่างต่างๆ ที่กลุ่มควบคุมคุณภาพซอฟต์แวร์ ได้ทำการทดสอบเป็นตัวแทนที่แท้จริงของงานที่ดำเนินการอยู่ได้แก่
- มีเอกสารการกำหนดขอบเขตตัวอย่างที่ทำการทดสอบ
 - มีการจัดทำเอกสารรายละเอียดเกณฑ์ในการคัดเลือกตัวอย่างที่ถูกต้องและเหมาะสม ได้แก่
 - มีการกำหนดปริมาณและประเภทของตัวอย่างแบบต่างๆ
 - คัดเลือกตามปริมาณความถี่หรือโอกาสในการใช้งาน
 - คัดเลือกตัวอย่างที่เป็นผลกระทบกับระบบย่อยอื่นๆ
 - คัดเลือกตัวอย่างที่สำคัญต่อการดำเนินงาน
 - คัดเลือกตัวอย่างที่เป็นตัวแทนของงานที่ดำเนินการ
- มีการวิเคราะห์ตรวจสอบตัวอย่างที่จะทำการทดสอบโดยกลุ่มผู้รับผิดชอบ
- มีการสุ่มตัวอย่างตรวจสอบเป็นประจำ
 - มีการประชุมสรุปผลการตรวจสอบเป็นประจำ
 - มีการจัดทำเอกสารสรุป และบันทึกผลการตรวจสอบทบทวนตัวอย่างที่ใช้ทดสอบ
- มีวิธีการหรือเทคนิคในการตรวจสอบที่มีการทดสอบหาข้อผิดพลาดของระบบ ที่เกิดจากการปรับปรุงแก้ไข (Regression testing) เป็นประจำได้แก่
- มีการจัดทำเอกสารการกำหนดขอบเขต ของการทดสอบหาข้อผิดพลาดของระบบ

- มีการจัดทำเอกสารการกำหนดวิธีการ ขั้นตอน และมาตรฐานในการทดสอบ
 - มีการกำหนดกิจกรรมการทดสอบหาข้อผิดพลาดที่เกิดจาก
 - การปรับปรุงแก้ไข ให้อยู่ในแผนงานการทดสอบ
 - มีการดำเนินการทดสอบหาข้อผิดพลาดอยู่เป็นประจำ โดยทีมงาน SCC พิจารณาจากรายงานการดำเนินงาน และผลการดำเนินงาน
 - มีการดำเนินการทดสอบหาข้อผิดพลาดก่อนและหลังการปรับปรุงแก้ไข พิจารณาจากรายงานผลการทดสอบ
 - มีการจัดทำรายงานผลการทดสอบตามช่วงเวลาเป็นประจำ
- มีวิธีการหรือเทคนิคที่ใช้เพื่อตรวจสอบว่าการทดสอบหาข้อผิดพลาดของระบบจากการปรับปรุงแก้ไขได้ดำเนินการอย่างพอเพียง ได้แก่
- มีการกำหนดผู้รับผิดชอบการทดสอบ
 - มีการดำเนินการทดสอบหาข้อผิดพลาดก่อน และหลังการปรับปรุงแก้ไข
 - มีการกำหนดให้กิจกรรมการทดสอบหาข้อผิดพลาดจากการปรับปรุงแก้ไข ให้อยู่ในเอกสารแผนงานการทดสอบ
 - มีการดำเนินการตรวจสอบหาข้อผิดพลาดอยู่เป็นประจำ
 - ปริมาณข้อผิดพลาดของระบบที่พบมีจำนวนน้อย
 - มีการจัดทำรายงานสรุปผลการทดสอบ เป็นประจำ

3. การจัดการบริหารเครื่องมือและเทคโนโลยีที่ใช้ (Tools and Technology Management) มีรายละเอียดเกี่ยวกับเครื่องมือและเทคโนโลยีที่ใช้ในกระบวนการวิศวกรรมซอฟต์แวร์ ซึ่งจะมุ่งเน้นเพื่อให้เกิดความมั่นใจว่ากระบวนการต่าง ๆ ได้ถูกจัดเตรียมเครื่องมือ และวิธีการพื้นฐานที่จะใช้ไว้เรียบร้อยแล้ว ประกอบด้วย

3.1 จัดให้มีเครื่องมือ และเทคโนโลยีช่วยในการดำเนินการพัฒนาซอฟต์แวร์ เพื่อช่วยในการดำเนินการพัฒนาซอฟต์แวร์มีประสิทธิภาพมากขึ้น องค์กรซอฟต์แวร์ที่จะเข้าสู่ระบบซีเอ็มเอ็มจะต้องเตรียมดำเนินการ และความพร้อมในด้านต่าง ๆ ดังนี้คือ

- มีการใช้ภาษาที่ใช้ในการออกแบบ (Program design Language - PDL) ใช้ในขั้นตอนการออกแบบ
- มีการใช้ภาษาโปรแกรมระดับสูง (High - Order Language) ในการพัฒนาผลิตภัณฑ์ซอฟต์แวร์
- มีเครื่องมือในการสร้างข้อมูลนำเข้าเพื่อการทดสอบอัตโนมัติ (Data Generator)
- มีซอฟต์แวร์ หรือเครื่องมือทางด้านคอมพิวเตอร์ที่ใช้ในการวัดผลการทดสอบ (Test Coverage)

- มีอุปกรณ์อำนวยความสะดวกในด้านการจัดทำเอกสารต่าง ๆ แบบทันทีทันใด (Interactive Documentation Facilities)

3.2 จัดให้มีเครื่องมือ และเทคโนโลยีช่วยในการติดตามตรวจสอบการดำเนินงานในการพัฒนาซอฟต์แวร์ ได้แก่

- มีเครื่องมือและเทคโนโลยีที่ใช้ในการควบคุม และติดตามกิจกรรมการเปลี่ยนแปลงต่าง ๆ ในทุกขั้นตอนการพัฒนาซอฟต์แวร์
- มีซอฟต์แวร์หรือเครื่องมือทางด้านคอมพิวเตอร์ ใช้ช่วยติดตามในการดำเนินงานในขั้นตอนการวิเคราะห์ความต้องการ ไปยังขั้นตอนการออกแบบซอฟต์แวร์
- มีซอฟต์แวร์หรือเครื่องมือทางด้านคอมพิวเตอร์ ใช้ช่วยติดตามในการดำเนินงานในขั้นตอนการออกแบบซอฟต์แวร์ ไปยังขั้นตอนการเขียนเขียนโปรแกรม
- มีซอฟต์แวร์หรือเครื่องมือทางด้านคอมพิวเตอร์ ใช้ช่วยติดตามตรวจสอบทุกหน้าที่ที่ต้องการ
- มีการใช้เครื่องมือในการตรวจสอบหาความผิดพลาดของโปรแกรมโดยทันที
- มีซอฟต์แวร์หรือเครื่องมือทางด้านคอมพิวเตอร์ที่ใช้ในการติดตามตรวจสอบ และรายงานผลสถานะของซอฟต์แวร์เก็บอยู่ในไลบรารีศูนย์กลางในคอมพิวเตอร์หลักของการพัฒนาซอฟต์แวร์

3.3 จัดให้มีเครื่องมือช่วยในการวิเคราะห์ ในการดำเนินการพัฒนาซอฟต์แวร์ ได้แก่

- มีเครื่องมือที่ใช้ในการวิเคราะห์ขนาด และกิจกรรมการเปลี่ยนแปลงต่าง ๆ ในองค์ประกอบของซอฟต์แวร์
- มีเครื่องมือที่ใช้ในการวิเคราะห์ความซับซ้อนของซอฟต์แวร์ (Software Complexity)
- มีเครื่องมือที่ใช้ในการวิเคราะห์การเชื่อมโยงระหว่างระบบย่อยต่าง ๆ (Cross Reference Between Modules)

3.4 จัดให้มีการใช้แบบจำลองในการดำเนินการพัฒนา ได้แก่

- มีการใช้แบบจำลองในการออกแบบองค์ประกอบ ที่มีความสำคัญของซอฟต์แวร์
- มีการใช้แบบจำลองในการออกแบบองค์ประกอบ ที่มีความสำคัญในการติดต่อเชื่อมโยงระบบงานซอฟต์แวร์

ข้อควรคำนึงในการนำมาใช้กับสังคมไทย

จากรายละเอียดการพิจารณาเพื่อปรับปรุงองค์กรซอฟต์แวร์เข้าสู่ระบบซีเอ็มเอ็ม จะทำให้องค์กรทราบว่าจะต้องเตรียมดำเนินการในด้านใดบ้าง หรือต้องเพิ่มเติม ปรับปรุง แก้ไข วิธีการดำเนินงานในปัจจุบันอย่างไร เพื่อที่จะพัฒนาองค์กรให้เข้าสู่ระบบซีเอ็มเอ็มไปแล้วนั้นยังต้องพิจารณาถึงข้อควรคำนึงในการนำมาใช้กับสังคมไทยด้วย เนื่องจากวิธีการบางอย่างตามรายละเอียดดังกล่าวไม่สามารถดำเนินการได้จริงในสังคมไทยซึ่งมีรายละเอียดดังนี้

1. การจัดการองค์กรและทรัพยากร (Organization and Resource Management)

ซึ่งแบ่งออกเป็น

1.1 การจัดการโครงสร้างองค์กร (Organization Structure) ซึ่งในรายละเอียดการ

พิจารณาเพื่อปรับปรุงองค์กรซอฟต์แวร์เข้าสู่ระบบซีเอ็มเอ็ม จะแนะนำว่าควรมีการแต่งตั้งจัดเตรียมทีมงานทางด้านวิศวกรรมซอฟต์แวร์ที่จำเป็น (เช่น ผู้จัดการซอฟต์แวร์ ทีมงานตรวจสอบรับรองคุณภาพ ทีมงานควบคุมรายละเอียดขอบเขตโครงสร้างซอฟต์แวร์ ทีมงานกลุ่มกระบวนการวิศวกรรมซอฟต์แวร์ ทีมงานด้านวิศวกรรมระบบซอฟต์แวร์ ทีมงานควบคุมการติดต่อเชื่อมโยงกับซอฟต์แวร์) การนำมาใช้กับสังคมไทย มีข้อควรคำนึงดังนี้

- ไม่สามารถดำเนินการได้ เนื่องจากบุคลากรไม่เพียงพอ
- ไม่สามารถดำเนินการได้ เนื่องจากมีงบประมาณสนับสนุนน้อย เพราะมักจะมุ่งเน้นในเรื่องการลงทุนในอุปกรณ์คอมพิวเตอร์ฮาร์ดแวร์มากกว่าซอฟต์แวร์
- ไม่สามารถดำเนินการได้ เนื่องจากไม่มีความรู้ทางด้านวิศวกรรมซอฟต์แวร์

1.2 การจัดการบริหารทรัพยากรที่ใช้ บุคลากรและการฝึกอบรม (Resource Personnel and Training) ซึ่งแบ่งออกเป็น

1.2.1 การจัดการบริหารทรัพยากรที่ใช้ (จัดเตรียมหาอุปกรณ์คอมพิวเตอร์และอุปกรณ์อำนวยความสะดวกเพื่อช่วยสนับสนุนการทำงาน) การนำมาใช้กับสังคมไทย มีข้อควรคำนึงถึงดังนี้

- ไม่สามารถจัดหาอุปกรณ์ที่เพียงพอ เนื่องจากมีข้อจำกัดด้านงบประมาณ
- ไม่จัดเตรียมอุปกรณ์อำนวยความสะดวก (โปรแกรมและเครื่องมือเพื่อช่วยสนับสนุน) เนื่องจากไม่เห็นความสำคัญในการใช้อุปกรณ์ช่วยสนับสนุน ไม่เห็นความสำคัญในการเพิ่มประสิทธิภาพงาน

1.2.2 การจัดหาและคัดเลือกบุคลากรที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ การนำมาใช้กับสังคมไทย มีข้อควรคำนึงถึงดังนี้

- บุคลากรทางด้านคอมพิวเตอร์ ที่มีคุณสมบัติเหมาะสมหาได้ยาก
- บุคลากรมักไม่มีความรู้ด้านวิศวกรรมคอมพิวเตอร์ เป็นอย่างดี
- บุคลากรมักไม่มีความสามารถในการใช้เครื่องมือหรือเทคโนโลยีทางด้านวิศวกรรมซอฟต์แวร์

1.2.3 การจัดโปรแกรมการฝึกอบรมให้ความรู้กับบุคลากร การนำมาใช้กับสังคมไทย มีข้อควรคำนึงถึงดังนี้

- โปรแกรมการฝึกอบรมทางด้านการบริหาร สามารถจัดทำได้แต่ขึ้นอยู่กับ การสนับสนุน และเงินทุนขององค์กร
- โปรแกรมการฝึกอบรมทางวิศวกรรมซอฟต์แวร์ หาเรียนได้ยาก ถ้าจัดการฝึกอบรมเองก็ขาดบุคลากรผู้เชี่ยวชาญ
- การฝึกอบรมเทคโนโลยีใหม่ทางด้านวิศวกรรมซอฟต์แวร์ อาจไม่ได้รับความร่วมมือจาก ผู้บริหาร ผู้จัดการอาวุโส เนื่องจากวิสัยทัศน์สูงทำให้ไม่ ยอมรับการฝึกอบรม หรือขาดความกระตือรือร้น

1.3 การจัดการบริหารเทคโนโลยี (Technology Management) ซึ่งเกี่ยวข้องกับวิธีการ หรือเทคนิคในการนำเอาเทคโนโลยีใหม่มาใช้ การควบคุมการใช้เทคโนโลยีนั้น ๆ และการเปลี่ยนแปลงเทคโนโลยี การนำมาใช้กับสังคมไทย มีข้อควรคำนึงถึงดังนี้

- องค์กรส่วนใหญ่ไม่เห็นความสำคัญในการจัดการบริหารเทคโนโลยี เพราะจะมุ่งเน้นเพียงการพัฒนาซอฟต์แวร์ให้ทันตามกำหนดเวลาเพียงอย่างเดียว
- องค์กรส่วนใหญ่ขาดการทำงานเป็นขั้นตอน และการดำเนินการเป็นลายลักษณ์อักษร เนื่องจากการพัฒนาซอฟต์แวร์ในปัจจุบันมุ่งเน้นเพื่อให้ทันตามกำหนดเวลา และการดำเนินการส่วนใหญ่เป็นแบบไม่เป็นทางการ ไม่มีเอกสารในการดำเนินงาน จึงไม่สนใจการปรับปรุงพัฒนาการดำเนินการให้มีลักษณะเป็นขั้นตอน หรือเป็นลายลักษณ์อักษร

2.การจัดการบริหาร กระบวนการทางด้านวิศวกรรมซอฟต์แวร์ (Software Engineering Process Management) ซึ่งประกอบด้วยการจัดการมาตรฐานของเอกสาร และขั้นตอนการทำงาน การวัดผลกระบวนการ การจัดการข้อมูลและการวิเคราะห์ข้อมูล การควบคุมกระบวนการ ซึ่งการนำมาใช้กับสังคมไทย มีข้อควรคำนึงดังนี้

- องค์กรส่วนใหญ่ขาดการทำงานเป็นขั้นตอน ตามกระบวนการพัฒนาซอฟต์แวร์ที่เป็นลายลักษณ์อักษร เนื่องจากการดำเนินการตามประสบการณ์ส่วนใหญ่ และมักมีการดำเนินการที่ไม่เป็นทางการ ทำให้การจัดการมาตรฐานของเอกสารและขั้นตอนการดำเนินงานทำได้ยาก ใช้เวลานานรวมถึงใช้ต้นทุนในการดำเนินการสูง
- บุคลากรขาดความรู้ด้านวิศวกรรมซอฟต์แวร์ที่เพียงพอ ทำให้การดำเนินงานล่าช้า งบประมาณบานปลาย และผลงานไม่มีประสิทธิภาพทำให้การจัดการบริหารกระบวนการ ทำได้ยาก ใช้เวลานาน รวมถึงการใช้ต้นทุนในการดำเนินงานสูง ได้แก่
 - การขาดความรู้ในการกำหนดขอบเขตของซอฟต์แวร์

- การขาดความรู้ด้านการประมาณขนาด ต้นทุนของซอฟต์แวร์ และทรัพยากรที่ต้องการ
 - การขาดความรู้และทักษะในการดำเนินโครงการซอฟต์แวร์
 - การขาดความรู้ด้านการจัดทำตารางเวลาการทำงาน
 - การขาดความรู้ด้านการวิเคราะห์และประเมินความเสี่ยง
 - การขาดความรู้และทักษะในการเจรจาต่อรอง
 - การขาดความรู้และทักษะในการติดตามตรวจสอบการดำเนินการ
- องค์กรส่วนใหญ่ไม่มีการจัดทำเอกสารข้อตกลงเป็นลายลักษณ์อักษร ทำให้อาจเกิดข้อโต้แย้งและปัญหาต่าง ๆ ฉะนั้นข้อตกลงต่าง ๆ และสรุปการประชุมโครงการ ควรจัดทำเป็นเอกสารลายลักษณ์อักษร และควรได้รับการยอมรับและเห็นด้วยจากทุกฝ่ายที่เกี่ยวข้อง
- องค์กรส่วนใหญ่ไม่มีการจัดทำเอกสารคู่มือ หรือมาตรฐานการทำงาน ในแต่ละขั้นตอนในการพัฒนาซอฟต์แวร์ รวมถึงเอกสารคู่มือการใช้งาน โปรแกรมหรือเครื่องมือช่วยในการสนับสนุน ทำให้การดำเนินการอาจล่าช้า งบประมาณบานปลาย และไม่มีประสิทธิภาพ การจัดทำคู่มือหรือมาตรฐานขึ้นมาใหม่ จะทำได้ยาก ใช้เวลานานรวมถึงใช้ต้นทุนในการดำเนินงานสูง
- องค์กรส่วนใหญ่ไม่มีการกำหนดผู้รับผิดชอบด้านการควบคุมขอบเขต โครงร่างซอฟต์แวร์ และกลุ่มควบคุมคุณภาพ เนื่องจากเห็นว่าไม่สำคัญ และต้องการประหยัดต้นทุน แต่อาจก่อให้เกิดผลเสีย คือการดำเนินงานต่าง ๆ อยู่นอกเหนือขอบเขต และไม่ปฏิบัติตามมาตรฐานที่กำหนดไว้ ทำให้แผนงานล่าช้า และใช้ทรัพยากรมากเกินไป
- องค์กรส่วนใหญ่ขาดการจัดเก็บข้อมูลเพื่อใช้ในการวิเคราะห์ด้านต่าง ๆ อย่างเป็นระบบ ทำให้ขาดข้อมูลในการวิเคราะห์กระบวนการติดตามการดำเนินงาน และปรับปรุงประสิทธิภาพกระบวนการ
- นิสัยคนไทยมีความยึดหยุ่น เกรงใจ และประนีประนอมมากเกินไป ทำให้ข้อตกลงพันธะสัญญาต่าง ๆ ไม่เป็นไปตามที่กำหนดไว้ไม่สามารถดำเนินการตามข้อตกลง ไม่สามารถควบคุมขอบเขตและมาตรฐาน ทำให้แผนงานล่าช้าและใช้ทรัพยากรมากเกินไป
- องค์กรส่วนใหญ่ขาดการตรวจสอบ และการติดตามการดำเนินงานที่ดีพอ ทำให้ไม่สามารถดำเนินการให้เป็นไปตามแผนงานที่ตั้งไว้ ควรจัดให้มีการตรวจสอบ และติดตามการดำเนินงานเป็นประจำ

3. การจัดการบริหารเครื่องมือและเทคโนโลยีที่ใช้ (Tools and Technology Management) ซึ่งเกี่ยวข้องกับการจัดเตรียมเครื่องมือและเทคโนโลยีช่วยในการดำเนิน

การพัฒนาซอฟต์แวร์ ช่วยในการติดตามตรวจสอบการดำเนินงาน ช่วยในการวิเคราะห์และจัดทำแบบจำลอง การนำมาใช้กับสังคมไทย มีข้อควรคำนึงดังนี้

- องค์กรส่วนใหญ่ขาดเครื่องมือและเทคโนโลยีที่ช่วยในการดำเนินการพัฒนาซอฟต์แวร์ ในขั้นตอนต่าง ๆ การจัดหาเครื่องมือหรือเทคโนโลยีเข้าช่วยในการดำเนินงาน อาจต้องใช้เวลาและใช้ต้นทุนสูง
- บุคลากรส่วนใหญ่ขาดความรู้ในการใช้เครื่องมือ และเทคโนโลยีที่ช่วยในการดำเนินการพัฒนาซอฟต์แวร์ การนำเครื่องมือหรือเทคโนโลยีเข้าช่วยในการดำเนินงาน จะต้องจัดเตรียมการฝึกอบรม เพื่อพัฒนาความรู้ให้กับบุคลากร

ข้อดีข้อเสียของการนำซีเอ็มเอ็มมาใช้

ข้อดี

1. สามารถนำซีเอ็มเอ็มมาประยุกต์ใช้ได้กับทั้งองค์กรผู้ผลิตซอฟต์แวร์ที่ต้องการจะปรับปรุงพัฒนาการดำเนินงาน และสามารถใช้เป็นเกณฑ์ในการตัดสินใจคัดเลือกบริษัทซอฟต์แวร์ที่จะประมูลงานได้ด้วย
2. สามารถนำไปประยุกต์ใช้ได้กับองค์กรผลิตซอฟต์แวร์ทุกขนาด ตั้งแต่ขนาดเล็ก ไปจนถึงขนาดใหญ่
3. ซีเอ็มเอ็มแสดงรายละเอียดขั้นตอนการพัฒนาองค์กรซอฟต์แวร์ไปยังระดับการเติบโตแต่ละระดับ โดยการอธิบายเป็นเอกสารอย่างละเอียด สามารถศึกษาเป็นแนวทางในการพัฒนาได้ด้วยตนเอง

ข้อเสีย

1. การพัฒนาองค์กรซอฟต์แวร์เข้าสู่ระดับซีเอ็มเอ็มในแต่ละระดับ จะต้องใช้งบประมาณสูงเพื่อปรับปรุงพัฒนาโครงสร้างองค์กร ทรัพยากรที่ใช้ การจัดเตรียมบุคลากรและการฝึกอบรม การจัดการบริหารกระบวนการทางด้านวิศวกรรมซอฟต์แวร์ และการจัดการบริหารเครื่องมือ และเทคโนโลยีที่ใช้
2. การพัฒนาองค์กรซอฟต์แวร์เข้าสู่ระดับซีเอ็มเอ็มในแต่ละระดับ จะต้องใช้เวลานาน และต้องปรับปรุงพัฒนาการดำเนินงานอย่างต่อเนื่อง เนื่องจากการที่จะบรรลุไปยังระดับการเติบโตที่สูงๆขึ้นไป จะต้องผ่านการพัฒนาระดับต้นก่อน
3. ปัจจุบันยังไม่มียุทธศาสตร์ หรือบริษัทตัวแทนที่ปรึกษาซีเอ็มเอ็มในประเทศไทย ฉะนั้นการศึกษารายละเอียดความรู้เกี่ยวกับซีเอ็มเอ็ม การทำแบบทดสอบระดับการเติบโต รวมถึงการพัฒนาองค์กรไปสู่ระดับการเติบโตที่สูงขึ้น จะต้องดำเนินการด้วยตนเอง โดยอาศัย เอกสารรายละเอียดซีเอ็มเอ็มที่มีอยู่