บทที่ 2

แนวคิดและทฤษฎีในการตรวจสอบการพัฒนาซอฟต์แวร์ประยุกต์

การพัฒนาซอฟต์แวร์ประยุกต์ให้ได้ซอฟต์แวร์ที่ใช้งานอย่างมีประสิทธิภาพ และสามารถสนับสนุนงานขององค์การได้เป็นอย่างดีนั้น จำเป็นที่จะต้องมีการตรวจสอบในแต่ละขั้นตอนของการพัฒนาซอฟต์แวร์ ซึ่งการตรวจสอบการพัฒนาซอฟต์แวร์มีแนวทางดังที่จะกล่าวต่อไปนี้

มาตรฐาน IEEE ที่ใช้ในการตรวจสอบการพัฒนาซอฟต์แวร์ประยุกต์

แนวทางการตรวจสอบตามมาตรฐาน IEEE Std 1028-1988,IEEE Standard for Software Reviews and audits.

1. ความหมายของการตรวจสอบ (AUDIT)

การตรวจสอบเป็นการประเมินค่าที่แยกเป็นอิสระจากผลิตภัณฑ์หรือขั้นตอนต่าง ๆ ของซอฟต์แวร์เพื่อสืบให้รู้แน่ว่าเป็นไปตามมาตรฐาน (standards) แนวทาง (guidelines) ข้อกำหนดคุณลักษณะ (specification) และแนวทางการปฏิบัติงาน (procedures) บนพื้นฐานของเกณฑ์ต่าง ๆ ที่ได้วางไว้ ซึ่งรวมถึงเอกสารที่ระบุถึง ดังนี้

- รูปแบบหรือเนื้อหาของผลิตภัณฑ์ที่ถูกผลิต
- ขั้นตอนต่าง ๆ ในการสร้างผลิตภัณฑ์
- วิธีการในการวัดว่าเป็นไปตามมาตรฐานหรือกฎเกณฑ์ที่ได้วางไว้

2. ซอฟต์แวร์อีลีเมนต์ (Software element)

ได้แก่เอกสารต่าง ๆ ที่จัดทำในระหว่างการดำเนินการ หรือ ได้รับในช่วงการพัฒนาซอฟต์แวร์ประกอบด้วย

- เอกสารการวางแผนโครงการ (Project planning documents) ตัวอย่างได้แก่ แผนงานการพัฒนาซอฟต์แวร์ (software development plans) และแผนงานในการทวนสอบและตรวจสอบความถูกต้อง (software verification and validation plans)
- ข้อกำหนดคุณลักษณะความต้องการ และการออกแบบซอฟต์แวร์ (Software requirements and design specification)
- เอกสารที่ใช้ในการทดสอบ (Test effort documentation)

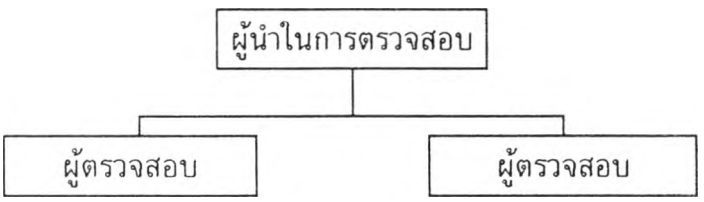
- เอกสารที่ต้องส่งมอบให้กับลูกค้า (Customer-deliverable documentation)
- โปรแกรมต้นฉบับ (Program source code)
- ตัวอย่างของการแก้ปัญหาซอฟต์แวร์
- รายงานต่าง ๆ เช่น รายงานการทบทวน (reviews) รายงานการตรวจสอบ (Audit) รายงานสถานะของโครงการ (project status) และข้อมูล เช่น ข้อมูลข้อผิดพลาดที่พบ ข้อมูลในการทดสอบ

3. สิ่งที่ต้องจัดเตรียมก่อนเริ่มทำการตรวจสอบ (Audits Process Prerequisites)

จุดประสงค์ของการตรวจสอบซอฟต์แวร์คือ การกำหนดจุดประสงค์ในการประเมินค่าของผลิตภัณฑ์ และ ขั้นตอนต่าง ๆ เพื่อยืนยันว่าเป็นไปตาม มาตรฐาน แนวทาง ข้อกำหนดคุณลักษณะ และแนวทางการปฏิบัติงาน เพื่อให้บรรลุถึงวัตถุประสงค์ดังกล่าวจึงควรมีสิ่งต่าง ๆ เหล่านี้ก่อน คือ

3.1 เกณฑ์ที่ใช้ในการตรวจสอบ เช่น หนังสือสัญญา ความต้องการ แผนงาน ข้อกำหนดคุณลักษณะ และมาตรฐานต่าง ๆ เปรียบเทียบกับซอฟต์แวร์อีลีเมนต์ และขั้นตอนต่าง ๆ ที่สามารถถูกประเมินผลได้

3.2 ทีมงานในการตรวจสอบ ซึ่งควรจะแยกเป็นอิสระไม่เกี่ยวข้องกับผลิตภัณฑ์ และขั้นตอนที่จะถูกตรวจโดยอาจใช้บุคคลจากภายนอกองค์กร ซึ่งโครงสร้างของทีมงานในการตรวจสอบการพัฒนาซอฟต์แวร์จะแบ่งตามหน้าที่ความรับผิดชอบ ซึ่งประกอบด้วย ผู้นำในการตรวจสอบ และผู้ตรวจสอบ ดังรูป 2.1



รูป 2.1 แสดงโครงสร้างของทีมงานตรวจสอบ

3.2.1 ผู้นำในการตรวจสอบ (A lead auditor)

ISO 10011-1ได้ให้แนวทางเกี่ยวกับหน้าที่ของผู้นำในการตรวจสอบดังนี้

- บริหาร และจัดการในเรื่องเกี่ยวกับการตรวจสอบ

- กำหนดงานต่าง ๆ ในการตรวจสอบ
- ช่วยคัดเลือกผู้ตรวจสอบ
- คอยให้คำแนะนำแก่ทีมงานตรวจสอบ
- จัดเตรียมแผนงานการตรวจสอบ
- กำหนดคุณสมบัติต่าง ๆ ของผู้ตรวจสอบ
- จัดเตรียมแบบฟอร์ม และเช็คลิสต์สำหรับการตรวจสอบ
- ทบทวนคุณภาพของเอกสารที่ใช้สนับสนุนระบบ
- รายงานสิ่งที่ไม่ถูกต้องที่อาจทำให้เกิดความเสียหายมากทันที
- คอยประสานงานกับผู้บริการขององค์กรที่ถูกตรวจสอบ และทีมงานตรวจสอบ
- จัดเตรียม และนำเสนอรายงานการตรวจสอบ

3.2.2 ผู้ตรวจสอบ (Auditor)

ISO 10011-1 ได้ให้แนวทางเกี่ยวกับหน้าที่ของผู้ตรวจสอบดังนี้

- ประเมินผลคุณภาพของระบบ
- ดำเนินการตรวจสอบตามหน้าที่ที่ได้รับมอบหมาย
- ดำเนินการตามแนวทางในการตรวจสอบ
- ไม่เปิดเผยเรื่องที่เป็นความลับ
- เก็บรวบรวมหลักฐานต่าง ๆ เกี่ยวกับคุณภาพของระบบ
- เก็บรักษาเอกสารเกี่ยวกับการตรวจสอบ การบันทึกข้อมูล และรายงาน
- จัดทำเอกสารที่ได้จากการตรวจสอบ และสรุปผล

3.3 ทีมงานในการตรวจสอบควรมีอำนาจเพียงพอในการจัดการเกี่ยวกับการตรวจสอบ

4. รูปแบบรายละเอียดของการดำเนินการ (Procedural Description Template)

รายละเอียดต่อไปนี้เป็นรูปแบบที่ใช้เพื่อการวางแผน การเตรียมการ และ การปฏิบัติการสำหรับการตรวจสอบ

4.1 จุดประสงค์ (Objective) อธิบายเป้าหมายของกระบวนการ

4.2 บทนำ (Abstract) อธิบายภาพรวมของกระบวนการ

4.3 หน้าที่ (Special Responsibilities) อธิบายบทบาทหน้าที่ที่รับผิดชอบของกระบวนการที่ระบุ

4.4 การนำเข้า (Input) ผลิตภัณฑ์ที่นำมาใช้ในขั้นตอนต่าง ๆ รวมทั้งสารสนเทศที่จำเป็น

4.5 เกณฑ์ในการเริ่มการตรวจสอบ (Entry Criteria) เกณฑ์หรือเงื่อนไขสำหรับการเริ่มการตรวจสอบ

4.6 วิธีการปฏิบัติ (Procedures) มาตรฐานขั้นตอนต่าง ๆ ในการดำเนินการ

4.6.1 การวางแผน (Planning)

4.6.2 บทนำ (Overview)

4.6.3 การเตรียมการ (Preparation)

4.6.4 การพิจารณาตรวจสอบ (Examination)

4.7 เกณฑ์ในการสิ้นสุดการตรวจสอบ (Exit Criteria) เกณฑ์หรือเงื่อนไขในการหยุดการตรวจสอบ

4.8 ผลลัพธ์ (Output) ผลลัพธ์ที่ได้จากการตรวจสอบหลังจากสิ้นสุดการตรวจสอบ

4.9 การนำกลับมาตรวจสอบได้ (Auditability) รายละเอียดของหลักฐานที่จำเป็นต้องใช้เพื่อนำกลับมาใช้ในการตรวจสอบภายหลัง

5. ขั้นตอนในการตรวจสอบ (The Audit Process)

5.1 วัตถุประสงค์ วัตถุประสงค์ของการตรวจสอบซอฟต์แวร์ก็เพื่อกำหนดวัตถุประสงค์ในการที่ยืนยันว่าผลิตภัณฑ์ และ ขั้นตอนที่ทำนั้นเป็นไปตามมาตรฐาน แนวทาง ข้อกำหนดคุณลักษณะ และแนวทางการปฏิบัติงาน ที่ได้วางไว้

5.2 บทนำ การตรวจสอบเป็นการดำเนินการให้เหมาะสมกับแผนงานและกระบวนการที่ได้จัดทำไว้ แผนงานในการตรวจสอบจะเป็นการจัดสร้างกระบวนการต่าง ๆ ที่ใช้ในการตรวจสอบและใช้ติดตามผลการดำเนินงานการตรวจสอบนั้น

ในการดำเนินการตรวจสอบ ผู้ตรวจสอบจะทำการประเมินผลซอฟต์แวร์อีลีเมนต์ที่ได้กล่าวไนข้างต้น และ ขั้นตอนต่าง ๆ สำหรับการผลิตนั้น เปรียบเทียบกับกฎเกณฑ์มาตรฐานในการตรวจสอบ เช่น หนังสือสัญญา ความต้องการ แนวทาง และมาตรฐาน

ผลที่ได้จากการตรวจสอบจะเป็นเอกสารและข้อเสนอแนะ เพื่อการจัดการส่งให้แก่องค์กรที่ตรวจสอบ และองค์กรภายนอกที่เกี่ยวข้องในแผนการตรวจสอบ รายงานการตรวจสอบจะประกอบไปด้วย รายการต่าง ๆ ที่ไม่เป็นไปตามที่กำหนดไว้ หรือการสั่งการเพื่อการทบทวนในภายหลัง โดยจะนำเสนอเมื่อทุกอย่างเป็นไปตามแผนในการตรวจสอบแล้ว ข้อเสนอแนะต่าง ๆ นี้จะถูกรายงานตามผลลัพธ์ที่ได้จากการตรวจสอบ

5.3 หน้าที่ที่เฉพาะเจาะจง เป็นหน้าที่ของหัวหน้าทีมตรวจสอบในการที่จะรวบรวม และ แนะนำการตรวจสอบ และร่วมกันจัดเตรียมแจกจ่ายรายงานสำหรับใช้ในการตรวจสอบ หัวหน้าทีมจะต้องมั่นใจว่าทีมงานได้มีการจัดเตรียมแนวทางวิธีการในการสำหรับการตรวจสอบและรายงานต่าง ๆ ที่เหมาะสมกับขอบเขตของการตรวจสอบ

ผู้ที่ริเริ่มในการจัดตั้งเรื่องที่จะตรวจสอบมีหน้าที่ในการกำหนดอำนาจหน้าที่ในการตรวจสอบ องค์กรบริหารการจัดการการตรวจสอบมีหน้าที่ในการตรวจสอบและรวบรวมข้อมูลต่าง ๆ ที่จำเป็นในการตรวจสอบ

5.4 การนำเข้า สิ่งต่าง ๆเหล่านี้เป็นที่ต้องการในการนำเข้าเพื่อให้การตรวจสอบประสบผลสำเร็จ

5.4.1 เป้าหมายและขอบเขตของการตรวจสอบ

5.4.2 เกณฑ์ที่ใช้ในการตรวจสอบ เช่น หนังสือสัญญา แผนงาน ข้อกำหนดคุณลักษณะ ระเบียบการ แนวทางและมาตรฐาน

5.4.3 ซอฟต์แวร์อีลีเมนต์ และกระบวนการที่จะถูกตรวจ

5.4.4 สารสนเทศในอดีตต่าง ๆ เกี่ยวกับผลิตภัณฑ์ และขั้นตอนที่จะถูกตรวจสอบขององค์กรที่รับผิดชอบ เช่น โครงสร้างขององค์กร

5.5 เกณฑ์ในการเริ่มการตรวจสอบ ความต้องการสำหรับการตรวจสอบจะเริ่มขึ้นเมื่อเหตุการณ์อย่างใดอย่างหนึ่งนี้เกิดขึ้น

5.5.1 เมื่อครบกำหนดระยะเวลาที่กำหนดไว้ในแผนงานโครงการ เช่น แผนงานการประกันคุณภาพซอฟต์แวร์ แผนงานพัฒนาซอฟต์แวร์

5.5.2 ผู้ใช้ต้องการให้มีการตรวจสอบตามวันเวลาที่ได้ระบุไว้

5.5.3 หน่วยงานภายใน เช่น ฝ่ายจัดการโครงการ ฝ่ายจัดการด้านหน้าที่ต่าง ๆ ต้องการให้มีการตรวจสอบ

5.5.4 เมื่อครบกำหนดระยะเวลาที่สำคัญของโครงการ

5.6 การปฏิบัติการ

5.6.1 การวางแผน องค์กรที่มีหน้าที่ตรวจสอบต้องมีการพัฒนา และจัดทำเอกสารเกี่ยวกับแผนงานในการตรวจสอบ ซึ่งในแผนงานควรมีขอบเขตต่าง ๆ ในการตรวจสอบดังนี้

- ขั้นตอนของโครงการที่จะถูกตรวจสอบ และ ช่วงเวลาที่ทีมงานตรวจสอบใช้ในการตรวจสอบ

- ซอฟต์แวร์ที่ต้องการให้ตรวจสอบและการนำมาใช้
- รายงานต่าง ๆ ควรระบุให้ชัดเจน เช่น รายงานผลการตรวจสอบ รายงาน ข้อเสนอแนะต่าง ๆ และ รูปแบบทั่วไปที่กำหนดขึ้น
- การเผยแพร่รายงาน
- กิจกรรมที่ต้องดำเนินตาม
- ความต้องการต่าง ๆ เช่น กิจกรรมที่จำเป็น ส่วนประกอบ และกระบวนการเพื่อเป็นไปตามขอบเขตของการตรวจสอบ
- เกณฑ์ที่ใช้ในการตรวจสอบ กำหนดตามพื้นฐานที่เป็นที่ยอมรับตามที่กำหนดไว้ในการนำเข้า
- แนวทางการปฏิบัติงาน และ รายการสำหรับการตรวจสอบ
- บุคคลากร ได้แก่ จำนวน ความชำนาญ ประสบการณ์ และ ความรับผิดชอบ
- องค์กรที่มีส่วนร่วมในการตรวจสอบ เช่น องค์กรที่เป็นเจ้าของผลิตภัณฑ์ และ ขั้นตอนต่างๆ จะถูกตรวจสอบ
- วัน เวลา สถานที่ และหมายกำหนดการ

5.6.2 บทนำ

- คำอธิบายข้อตกลงต่าง ๆ ที่ใช้อยู่ เช่น ขอบเขตของการตรวจสอบ แผนงานและข้อตกลงต่าง ๆ ที่เกี่ยวข้อง
- อธิบายถึงผลิตภัณฑ์ และ ขั้นตอนต่าง ๆ ที่จะถูกตรวจสอบ
- อธิบายถึงขั้นตอนในการตรวจสอบรวมทั้งวัตถุประสงค์และผลลัพธ์
- สิ่งที่คาดหวังในการสนับสนุนการตรวจสอบ เช่น ำนวนคนที่จะให้สัมภาษณ์ และ สิ่งอำนวยความสะดวกต่าง ๆ
- หมายกำหนดการต่าง ๆ ในการตรวจสอบ

5.6.3 การเตรียมการ

5.6.3.1 สิ่งที่ทีมงานตรวจสอบต้องมีก่อนมีดังนี้

- ความเข้าใจองค์กร ทีมงานต้องรู้หน้าที่ที่ต้องรับผิดชอบ และ กิจกรรมที่ต้องปฏิบัติในองค์กรการตรวจสอบ
- ความเข้าใจในผลิตภัณฑ์ และขั้นตอนต่าง ๆ ทีมงานต้องมีการเรียนรู้ผลิตภัณฑ์และขั้นตอนต่าง ๆ ที่จะถูกตรวจสอบจากการอ่านและการสรุปความจากผู้รู้
- ความเข้าใจในเกณฑ์ที่ใช้ในการตรวจสอบ เป็นสิ่งที่จำเป็นที่ทีมงานตรวจสอบต้องมีการเรียนรู้ และมีความคุ้นเคยกับเกณฑ์ที่ใช้ในการตรวจสอบ

- การจัดเตรียมรายงานการตรวจสอบ เป็นสิ่งที่จำเป็นที่ต้องมีการจัดทำรายงานที่จะต้องใช้ในการตรวจสอบ เพื่อให้ทีมงานใช้เป็นรูปแบบเดียวกัน ตามที่ระบุไว้ในแผนงานการตรวจสอบ

- รายละเอียดของแผนงานการตรวจสอบ เลือกวิธีการที่เหมาะสมสำหรับแต่ละขั้นตอนในโปรแกรมการตรวจสอบ

5.6.3.2 หัวหน้าทีมงานจำเป็นต้องมีการเตรียมการสิ่งเหล่านี้

- ทีมงานและการอบรมที่จำเป็น
- สิ่งอำนวยความสะดวกสำหรับการสัมภาษณ์
- วัตถุดิบ เอกสาร และเครื่องมือ ที่ใช้ในการตรวจสอบ
- ซอฟต์แวร์อีลีเมนต์ที่จะถูกตรวจสอบ เช่น เอกสาร แฟ้มข้อมูลคอมพิวเตอร์ บุคคลที่จะถูกสัมภาษณ์
- หมายกำหนดการการสัมภาษณ์

5.6.4 การพิจารณาตรวจสอบ สิ่งที่ได้ถูกเลือกเพื่อตรวจสอบจะถูกนำมาประเมินผลเทียบกับเกณฑ์ที่ใช้ในการตรวจสอบ หลักฐานต่าง ๆ ควรมีการตรวจสอบให้แน่ชัด การตรวจสอบควรทำให้สอดคล้องกับขอบเขตของการตรวจสอบ ดังนี้

- ทบทวนกระบวนการ และ คำสั่ง
- ตรวจสอบโครงสร้างของการทำงาน
- ตรวจสอบหลักฐานที่นำมาใช้และควบคุมให้เหมาะสม
- สัมภาษณ์บุคคลที่เกี่ยวข้องเพื่อสอบถามถึงสถานะและหน้าที่ในแต่ละขั้นตอนของผลิตภัณฑ์
- ตรวจสอบเอกสารต่าง ๆ
- ทดสอบอีลีเมนต์

5.6.5 การรายงาน หลังจากที่ได้ทำการตรวจสอบแล้วทีมงานจะต้องมีการรายงานผลแก่องค์กรตรวจสอบ เพื่อการทบทวน และ การแนะนำ แล้วจึงนำกลับมาทำการทบทวน และ แก้ไขข้อความต่าง ๆ ที่ยังคลุมเคลือไม่ชัดเจน หลังจากนั้นจึงนำเสนอรายงานในรูปแบบที่เป็นทางการ

5.7 เกณฑ์ในการสิ้นสุดการตรวจสอบ การตรวจสอบจะเสร็จสมบูรณ์เมื่อ

- แต่ละอีลีเมนต์ที่อยู่ในขอบเขตของการตรวจสอบได้ถูกตรวจสอบจนครบ
- ผลของการตรวจสอบได้นำเสนอแก่ องค์กรการตรวจสอบแล้ว
- ผลการตรวจสอบฉบับร่างได้ถูกประเมินผลแล้ว
- ผลการตรวจสอบครั้งสุดท้ายอย่างเป็นทางการได้ถูกนำเสนอแก่องค์กรการตรวจสอบ

- รายงานการตรวจสอบได้ถูกจัดเตรียม และ ส่งให้ผู้ที่เกี่ยวข้องตามแผนงานการตรวจสอบเรียบร้อยแล้ว

- ถ้าในแผนงานต้องการให้มีรายงานข้อเสนอแนะต่าง ๆ และรายงานข้อเสนอแนะดังกล่าวได้ถูกจัดเตรียม และส่งให้ผู้ที่เกี่ยวข้องตามแผนงานการตรวจสอบเรียบร้อยแล้ว

- องค์กรที่ถูกตรวจสอบได้ดำเนินการตามขอบเขต หรือสัญญาสำหรับการตรวจสอบ

5.8 ผลลัพธ์ ตามรูปแบบที่เป็นมาตรฐานของรายงานการตรวจสอบนั้น รายงานการตรวจสอบควรจะมีสิ่งต่าง ๆ เหล่านี้

- การระบุการตรวจสอบ ได้แก่ ชื่อรายงาน วันเดือนปี องค์กรผู้ตรวจสอบ องค์กรที่ ถูกตรวจสอบ

- ขอบเขต ขอบเขตของการตรวจสอบประกอบไปด้วย มาตรฐานที่ใช้ ข้อกำหนดคุณลักษณะ วิธีปฏิบัติ และ ขั้นตอนต่าง ๆ การสร้างเกณฑ์ในการตรวจสอบ เปรียบเทียบกับสิ่งที่จะถูกตรวจสอบซึ่งได้แก่ ซอฟต์แวร์อีลีเมนต์ และขั้นตอนต่าง ๆ

- ข้อสรุป บทสรุปและการแสดงความหมายของสิ่งที่ค้นพบในการตรวจสอบ รวมทั้งรายการต่าง ๆ ที่ไม่เป็นไปตามที่กำหนด

- สรุปความ แสดงรายการต่าง ๆ ของ ซอฟต์แวร์อีลีเมนต์ ขั้นตอนต่าง ๆ และ คำวินิจฉัยสิ่งที่ถูกตรวจสอบ

- การติดตาม ชนิดและเวลาของการติดตามการตรวจสอบ

5.9 การนำกลับมาใช้ตรวจสอบได้ใหม่ เอกสารต่าง ๆ ในการตรวจสอบต้องมีการบำรุงรักษาโดยองค์กรตรวจสอบ เพื่อนำกลับมาใช้สำหรับกำหนดช่วงเวลาในการตรวจสอบในภายหลัง ซึ่งประกอบด้วย

- โปรแกรมการทำงานทั้งหมด รายการตรวจสอบและคำอธิบายประกอบ
- ทีมงาน
- บันทึกการสัมภาษณ์, บันทึกการสังเกตการณ์
- หลักฐานการทดสอบต่าง ๆ
- สำเนาของการตรวจสอบรายการต่าง ๆ และคำอธิบาย
- โครงร่างรายงานที่ได้จากองค์กรการตรวจสอบ
- บันทึกการติดตามที่จำเป็น

6. การตรวจสอบในช่วงการดำเนินการ (In-Process Audit)

เป็นการตรวจสอบในช่วงการออกแบบ และการพัฒนาซอฟต์แวร์ก่อนที่จะมีการตรวจสอบหน้าที่ เพื่อที่จะตรวจสอบความสอดคล้องของการออกแบบว่า

6.1 การติดต่อระหว่างฮาร์ดแวร์ กับ ซอฟต์แวร์นั้นมีความสอดคล้องกับความต้องการการออกแบบใน ข้อกำหนดคุณลักษณะความต้องการของซอฟต์แวร์

6.2 โปรแกรมได้ถูกทดสอบเรียบร้อยตามแผนงานการทวนสอบ และการตรวจสอบความถูกต้อง เพื่อมั่นใจได้ว่าความต้องการหน้าที่ต่าง ๆ (Functional requirements) ของข้อกำหนดคุณลักษณะความต้องการเป็นที่น่าพอใจ

6.3 การออกแบบผลิตภัณฑ์เป็นไปตามรายละเอียดการออกแบบซอฟต์แวร์ รวมทั้งเหมาะสมกับความต้องการหน้าที่ต่าง ๆ ในข้อกำหนดคุณลักษณะตวามต้องการซอฟต์แวร์

6.4 โปรแกรมนั้นมีความสอดคล้องกับรายละเอียด การออกแบบซอฟต์แวร์

ข้อมูลที่นำเข้าควรจะมีรายละเอียดของขั้นตอนต่าง ๆ และตัวอย่างที่แสดงให้เห็นขั้นตอนการทำงานของผลิตภัณฑ์

ผลลัพธ์ที่ได้ในการตรวจสอบในช่วงการดำเนินการควรจะเป็นเอกสารรายงานการตรวจสอบในช่วงการดำเนินการที่ระบุถึงสิ่งที่ไม่ตรงกัน

7. การตรวจสอบหน้าที่ (The Functional Audit (FA))

จุดประสงค์ของการตรวจสอบหน้าที่ก็เพื่อการกำหนดการประเมินผลที่แยกเป็นอิสระสำหรับผลิตภัณฑ์ซอฟต์แวร์ เพื่อดูว่าหน้าที่และประสิทธิภาพนั้นสอดคล้องตามข้อกำหนดคุณลักษณะความต้องการที่ได้กำหนดไว้ การตรวจสอบนี้ควรทำก่อนที่จะมีการส่งมอบซอฟต์แวร์

7.1 การกำหนดซอฟต์แวร์อีลีเมนต์ที่จะถูกตรวจสอบโดย

7.1.1 การกำหนดชื่อ

7.1.2 หมายเลขแสดงคุณลักษณะ

7.1.3 หมายเลขรายการแสดงรูปแบบ

7.2 วัตถุดิบหรือข้อมูลที่ถูกกำหนดให้ตรวจสอบประกอบด้วย

7.2.1 ข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์

7.2.2 สำเนาออปเจคโค๊ด

7.2.3 รายการที่แสดงสิ่งที่ถูกคัดออกเทียบกับรูปแบบรายการที่ระบุ

7.2.4 โปรแแกรมทดสอบที่สามารถใช้ได้ เพื่อใช้ทดสอบรูปแบบรายการกับอุปกรณ์ทดสอบอัตโนมัติ

7.2.5 รายงานการทวนสอบ และ การตรวจสอบซอฟต์แวร์

7.2.6 รายงานการตรวจสอบในช่วงการดำเนินการ

7.2.7 เอกสารการทดสอบต่าง ๆ เช่น แผนงาน, ข้อกำหนดคุณลักษณะ, ระเบียบการ และรายงานต่าง ๆ

7.2.8 รายการหน้าที่ต่าง ๆ ที่ได้ผ่านการทดสอบแล้ว

7.2.9 รายการของแผนงาน รายการสิ่งที่ยังไม่ได้ทดสอบ และรายการการตรวจสอบหน้าที่

7.3 กิจกรรมต่าง ๆ ที่ใช้ในการตรวจสอบหน้าที่

7.3.1 การตรวจสอบเอกสารการทดสอบต่าง ๆ อย่างเป็นทางการเทียบกับข้อมูลที่ใช้ทดสอบ ตรวจว่าผลลัพธ์ที่ได้ครบถ้วน และ ถูกต้อง รวมทั้งเอกสารข้อบกพร่องทั้งหมด

7.3.2 การตรวจสอบรายงานการทวนสอบ และ การตรวจสอบความถูกต้องที่ถูกจัดทำขึ้น เพื่อดูว่ารายการต่าง ๆ ที่แสดงไว้ในรายงานนั้นถูกต้อง มีการทดสอบรายการต่าง ๆ อย่างครบถ้วน และ รายงานต่าง ๆ ที่แสดงถึงปัญหาต่าง ๆ พร้อมทั้งการแก้ไขให้ถูกต้องนั้นก็ควรนำมาทบทวนเพื่อให้มั่นใจได้ว่าปัญหานั้นได้เคยมีการตรวจสอบทางเทคนิคมาแล้ว

7.3.3 เอกสารที่ส่งมอบควรมีการทบทวน เพื่อความมั่นใจในความถูกต้อง และความสมเหตุสมผล

7.3.4 การทบทวนผลลัพธ์ของการออกแบบขั้นต้น การทบทวนผลลัพธ์จุดสำคัญในการออกแบบควรจะถูกสุ่มมาตรวจเพื่อดูว่าสิ่งที่พบเป็นไปตามต้องการและสมบูรณ์ทีมงานตรวจสอบต้องเปรียบเทียบโค๊ด (code) กับเอกสารเกี่ยวกับความต้องการต่าง ๆ ของซอฟต์แวร์ตามที่มีอยู่ในเอกสารข้อกำหนดคุณลักษณะเพื่อดูว่าโค๊ดต่าง ๆ นั้นเป็นไปตามเอกสารความต้องการ

7.4 การตรวจสอบนี้ไม่เพียงจะเกี่ยวข้องเฉพาะเรื่องหน้าที่ต่าง ๆ อย่างเดียวเท่านั้น แต่ยังรวมถึงการตรวจสอบถึงประสิทธิภาพด้วย เช่น การทดสอบกับเอกสารการทดสอบที่เกี่ยวข้องและข้อมูลที่ถูกต้องควรที่จะเพียงพอกับประสิทธิภาพ สำหรับพารามิเตอร์ที่เกี่ยวกับประสิทธิภาพที่ไม่สามารถตรวจสอบได้อย่างสมบูรณ์ในช่วงการทดสอบ การจำลองแบบหรือการวิเคราะห์อื่น ๆ ควรจะทำให้เสร็จสิ้นเพื่อสร้างความมั่นใจว่ารูปแบบรายการนั้นเป็นไปตามเกณฑ์ของประสิทธิภาพที่ได้กำหนดไว้เมื่อนำไปใช้งาน

รายงานการตรวจสอบหน้าที่ควรที่จะประเมินผลในแต่ละซอฟต์แวร์อีลีเมนต์ที่ถูกตรวจสอบ และกำหนดการประเมินผลนี้เป็นมาตรฐานในรายงานการตรวจสอบ การประเมินผลจะมีรูปแบบ คือ

- การอนุมัติ (Approval) ถ้าซอฟต์แวร์อีลีเมนต์ที่ตรวจสอบมีหน้าที่ที่ต้องการอยู่ในเกณฑ์ขั้นต่ำตามมาตรฐานที่กำหนด
- การอนุมัติเป็นกรณีพิเศษ (Contingent Approval) ถ้าผลของการประเมินผลซอฟต์แวร์อีลีเมนต์ที่ถูกตรวจสอบนั้นสมบูรณ์ครบถ้วนตามเกณฑ์ที่วางไว้ทุกอย่าง
- ไม่อนุมัติ (Disapproval) ถ้าซอฟต์แวร์อีลีเมนต์ที่ตรวจสอบไม่เป็นไปตามเกณฑ์ที่สำคัญ ๆ

8. การตรวจสอบคุณภาพของระบบ (The Quality System Audit (QSA))

จุดประสงค์ของการตรวจสอบคุณภาพของระบบ หรือ การประเมินคุณภาพของโปรแกรมก็เพื่อประเมินผลว่าซอฟต์แวร์นั้นเป็นไปตามแผนงานการรับประกันคุณภาพของซอฟต์แวร์ (a software quality assurance plan)

8.1 จุดประสงค์หลักของการตรวจสอบคุณภาพของระบบก็เพื่อ ทำการตรวจสอบทบทวนหลักฐานต่าง ๆ ซึ่งได้แก่

8.1.1 เอกสารเกี่ยวกับคุณภาพของโปรแกรม ที่พัฒนาขึ้นโดยองค์กรที่พัฒนา ซึ่งอย่างน้อยควรจะมีองค์ประกอบตาม ANSI/IEE Std 730-1984 [A1] หรือมาตรฐานอื่น ๆ ที่เกี่ยวข้อง

8.1.2 องค์กรที่พัฒนาซอฟต์แวร์ได้ดำเนินการตามเอกสารเกี่ยวกับคุณภาพของโปรแกรมที่ได้จัดทำไว้ในแผนงานการตรวจสอบคุณภาพของซอฟต์แวร์ นั่นคือเป็นไปตามเกณฑ์ในด้านประสิทธิภาพตามที่องค์กรวางไว้ เช่น มาตรฐาน และ ระเบียบวิธีการต่าง ๆ ที่ใช้นั้น ถูกต้องตามกฎเกณฑ์ที่ได้วางไว้ เป็นไปตามหนังสือสัญญา และตรงกับมาตรฐาน ANSI/IEEE Std 730-1984 [A1] หรือมาตรฐานเกี่ยวกับการรับประกันคุณภาพอื่น ๆ ที่เกี่ยวข้อง

8.2 สิ่งสำคัญในการตรวจสอบคุณภาพของซอฟต์แวร์ คือการระบุถึงเอกสารเกี่ยวกับคุณภาพของโปรแกรม และการนำมาใช้ในทีมงานตรวจสอบโดยดำเนินการตามกิจกรรมต่าง ๆ ดังนี้

8.2.1 ตรวจสอบเอกสารคุณภาพโปรแกรม และ รายการอื่น ๆที่เกี่ยวข้องเพื่อความครบถ้วนสมบูรณ์ เพื่อการติดตามตรวจสอบ และ ความสมเหตุสมผล

8.2.2 สัมภาษณ์พนักงาน

8.2.3 ทำการตรวจสอบในช่วงการดำเนินการที่เกี่ยวข้องเช่น การสังเกตการณ์และหลักฐานต่าง ๆ

8.2.4 ตรวจสอบรายงานจากการตรวจสอบหน้าที่ที่ผ่านมา

รายการตรวจสอบคุณภาพของซอฟต์แวร์ควรจะมีการประเมินผลของคุณภาพของโปรแกรม และ การประเมินผลเทียบกับรายงานการตรวจสอบมาตรฐาน ส่วนการประเมินผลควรใช้รูปแบบ เช่นเดียวกับการตรวจสอบหน้าที่

ขั้นตอนการทำงานในการพัฒนาซอฟต์แวร์ประยุกต์ (Software development life cycle)

ขั้นตอนการพัฒนาซอฟต์แวร์ตามวงจรชีวิตในการพัฒนาซอฟต์รูปแบบ Waterfall แบ่งออกเป็น

- **ขั้นตอนแนวคิดเบื้องต้น (The Concept Phase)**

หรือถูกเรียกโดย IEEE ว่าขั้นตอน Concept Exploration Phase จะเป็นการจัดเตรียมข้อมูลพื้นฐานเกี่ยวกับ

1. การจัดเตรียมข้อเสนอสิ่งที่ต้องการ (Request for proposal - RFP) ซึ่งจะใช้ประโยชน์เมื่อโครงการติดต่อกับผู้พัฒนาภายนอก ซึ่งประกอบด้วยรายละเอียดปัญหา และวัตถุประสงค์ของโครงการ ความต้องการทางด้านเทคนิค ต้นทุน รายละเอียดสิ่งที่ต้องการให้จัดส่ง (เอกสาร ซอฟต์แวร์ การฝึกอบรม ฮาร์ดแวร์ และอุปกรณ์ที่เกี่ยวข้อง เครื่องมือในการพัฒนาและทดสอบ) ขอบเขตของงานที่ต้องดำเนินการและตารางการยื่นข้อเสนอ และตารางการตัดสินใจเลือกผู้พัฒนา รวมทั้งตารางเวลาในการพัฒนางานสำเร็จ
2. การกำหนดขอบเขตความต้องการซอฟต์แวร์ (ขั้นตอนต่อไป)
3. แผนงานเริ่มต้น และการจัดเตรียมการประเมินงาน

ในขั้นตอนนี้จะเกิดเอกสารสำหรับโครงการขึ้น 2 ชนิด คือ

1. คำอธิบายรายละเอียดผลิตภัณฑ์ (The Product Description) เป็นเอกสารเกี่ยวกับทางด้านการตลาดใช้ในการแสดงรายละเอียดคุณสมบัติโดยทั่วไปของผลิตภัณฑ์
2. เอกสารรายละเอียดแนวคิดเบื้องต้น (A Concept Document) เป็นเอกสารทางด้านเทคนิค และแบบฟอร์มที่เป็นกิจกรรมหลักของขั้นตอนนี้ ได้แก่ Request for proposal -RFP และแผนงานเบื้องต้น

- **ขั้นตอนการกำหนดรายละเอียดความต้องการซอฟต์แวร์**
  (The Software Requirements Phase)

หรือเรียกว่าขั้นตอนการกำหนดขอบเขต (The Definition Phase) เป็นขั้นตอนในการพัฒนาซอฟต์แวร์อย่างเป็นทางการขั้นแรก มีเป้าหมายเพื่อให้ได้รายละเอียดของปัญหาที่ต้องการแก้ไขอย่างสมบูรณ์ครบถ้วนและกำหนดความต้องการต่าง ๆ ได้แก่ ความต้องการของผู้ใช้ ซึ่งต้องมีการกำหนดอย่างระมัดระวังรวมทั้งในเรื่องการจัดทำเอกสารทั้งในแง่ของหน้าที่ต่าง ๆ และความต้องการอื่น ๆ ที่เสริมเข้ามา เช่น ความต้องการในเรื่องความมีประสิทธิภาพ ความเชื่อถือได้ เอกสารประกอบของผู้ใช้ การฝึกอบรมผู้ใช้ และต้นทุน เป็นต้น ความต้องการในเรื่องของสิ่งแวดล้อมที่เกี่ยวข้องในแง่ของซอฟต์แวร์และซอฟต์แวร์ที่ต้องใช้ ในช่วงของการวิเคราะห์ความต้องการนี้เรายังไม่ต้องคำนึงถึงในเรื่องจะทำอย่างไรจึงจะได้ความต้องการเหล่านี้จนกว่าจะถึงขั้นตอนของการออกแบบ ผลลัพธ์ที่ได้จะประกอบด้วยการกำหนดรายละเอียดของระบบซอฟต์แวร์ที่จะพัฒนาซึ่งเป็นเอกสารที่ให้รายละเอียดเกี่ยวกับปัญหาต่าง ๆที่จะแก้ไขเพื่อช่วยในการทำความเข้าใจของทั้งนักวิเคราะห์ระบบ และ ผู้ใช้ และเป็นพื้นฐานสำหรับการติดต่อระหว่างทั้ง 2 ฝ่าย ซึ่งอาจจะเป็นทางการหรือไม่เป็นทางการก็ได้ อีกทั้งยังใช้ในการประเมินผลการตรวจสอบระบบในขั้นสุดท้ายว่าระบบที่ได้เป็นไปตามข้อกำหนดรายละเอียดหรือไม่

ก่อนที่การออกแบบ (Design) และการนำไปใช้งาน (Implementation) จริง ๆ จะเริ่มขึ้นนั้นควรมีการประเมินความเป็นไปได้ ของการพัฒนาซอฟต์แวร์นั้นเสียก่อน ซึ่งการประเมินความเป็นไปได้นี้จะทำการประเมินความเป็นไปได้ทางเทคนิคในแง่ของฮาร์ดแวร์และ ซอฟต์แวร์ที่เกี่ยวข้องว่าสามารถจะนำมาใช้กับระบบที่จะพัฒนาได้ดีแค่ไหนจำเป็นต้องมีการเปลี่ยนแปลงหรือไม่ และการประเมินความเป็นไปได้ในแง่ของเศรษฐศาสตร์ซึ่งเป็นการประเมินความเป็นไปได้ในแง่ของความคุ้มค่าต่อการลงทุนว่าผลประโยชน์ที่ได้รับมีมากกว่าต้นทุนที่เกิดขึ้นจากการพัฒนาซอฟต์แวร์หรือไม่ โดยที่ต้นทุนในการพัฒนาซอฟต์แวร์จะถูกกำหนดจากเวลาและค่าจ้างที่จำเป็นในการพัฒนาซอฟต์แวร์

ในขั้นตอนนี้จะเกิดเอกสารขึ้น 3 ชนิด คือ

1. เอกสารรายละเอียดความต้องการระบบงานซอฟต์แวร์ (Software requirements specification document)
2. แผนงานการพัฒนาโครงการ (Project development plan)
3. แผนงานการทดสอบซอฟต์แวร์ (Software test plan)

- **ขั้นตอนการออกแบบ (The Design phase)**

มักจะถูกแบ่งออกเป็น 2 ขั้น คือ ขั้นการออกแบบขั้นต้น (Top level design) และขั้นการออกแบบรายละเอียด (Detailed design) ซึ่งการแบ่งขั้นตอนการออกแบบเป็น 2 ระดับนี้มีประโยชน์กว่าการใช้ระดับเดียว ซึ่งจะมีความสำคัญมากในโครงการขนาดกลาง และขนาดใหญ่ เพราะข้อผิดพลาดในการออกแบบหลักจะถูกพบในช่วงต้นเร็วที่สุดเท่าที่จะเป็นได้ เมื่อข้อผิดพลาดหลัก ๆ ในการออกแบบถูกพบในช่วงสุดท้ายของการออกแบบขั้นต้นจะง่ายในการแก้ไขมากกว่า ถ้าพบหลังจากการออกแบบรายละเอียดทั้งหมดเสร็จแล้ว

วิธีการที่ใช้ในการออกแบบ (Design Methods)

ในการออกแบบนี้เรามีวิธีการต่าง ๆ ที่ใช้อยู่หลายวิธี ดังแสดงในตารางที่ 2.1 วิธีการเหล่านี้จะเป็นให้แนวทางในการใช้งาน และ วิธีการปฏิบัติในการออกแบบระบบ

ตารางที่ 2.1 แสดงตัวอย่างของวิธีการที่ใช้ออกแบบ

Name	Description
Decision tables	Matrix representation of complex decision logic at the detail design level.
E-R	Entity-Relationship Model.Family of graphical techniques for expressing data-relationships [Chen76].
JSP	Jackson Structure Programming. Data structure oriented method.
OOD	Object oriented design; exists in many flavors.
SA/SD	Structure Analysis/Structure Design. Data flow design technique.
SA/WM	Ward-Mellor extension to Structured Analysis so that real-time aspects can be described.
SADT	Structure Analysis and Design Technique. Graphical language emphasizing hierarchical relations.
SSADM	Structured System Analysis and Design Method.A highly prescriptive method for performing the analysis and design stages;UK standard[Downs88]

แนวทางในการเลือกใช้วิธีการที่ใช้ในการออกแบบ

ในการเลือกใช้วิธีการต่าง ๆ ในการออกแบบนั้นไม่ใช่เรื่องง่ายเนื่องจากว่าในแต่ละวิธีนั้นต่างก็มีทั้งข้อดีและข้อเสีย แต่เราพอที่จะสรุปปัจจัยที่มีผลต่อการเลือกวิธีการที่ใช้ในการออกแบบให้ประสบผลสำเร็จได้ดังนี้

- ความคุ้นเคยกับปัญหาที่จะแก้ไข ถ้าผู้ออกแบบมีความคุ้นเคยกับปัญหาที่จะแก้ไขเป็นอย่างดีการใช้เทคนิคจากบนลงล่าง(top-down approach) หรือเทคนิคที่อาศัยพื้นฐานที่เกี่ยวกับโครงสร้างของข้อมูลจะใช้ได้ดี
- ประสบการณ์ของผู้ออกแบบ ถ้าผู้ออกแบบมีประสบการณ์เกี่ยวกับวิธีการที่ใช้ในการออกแบบมากก็สามารถที่จะพิจารณาจากกฎเกณฑ์ข้อบังคับและข้อจำกัดของแต่ละวิธีได้อย่างเหมาะสม
- เครื่องมือที่มีใช้ เครื่องมือที่ใช้สนับสนุนวิธีการในการออกแบบนั้นส่วนใหญ่จะสนับสนุนวิธีการใดวิธีการหนึ่ง ซึ่งขึ้นอยู่กับว่าองค์กรได้เลือกใช้วิธีการใดในการออกแบบ

เอกสารประกอบในการออกแบบ (Design Documentation)

เช่นเดียวกันกับในช่วงการวิเคราะห์ความต้องการ ในช่วงของการออกแบบก็ต้องมีเอกสารต่าง ๆ ที่ใช้ในการอธิบายรายละเอียดต่าง ๆ ในการออกแบบ IEEE 1016 ได้มีการจัดทำแนวทางเกี่ยวกับเอกสารประกอบในการออกแบบไว้โดยกล่าวถึงข้อมูลหลัก ๆ และคุณสมบัติของเอกสารที่จำเป็นอยู่ 10 อย่างคือ

1. การแสดงชื่อ (Identification) แสดงชื่อของส่วนประกอบต่าง ๆ เพื่อใช้ในการอ้างอิงซึ่งชื่อนี้ต้องไม่ซ้ำกัน

2. ชนิด (Type) ชนิดของส่วนประกอบ เช่น เป็นระบบย่อย กระบวนการ โมดูล หรือ แฟ้มข้อมูล

3. ความมุ่งหมาย (Purpose) ความมุ่งหมายของแต่ละส่วนประกอบคืออะไรในส่วนนี้จะมีการอ้างถึงข้อกำหนดคุณลักษณะความต้องการที่ได้กล่าวมาแล้ว

4. ฟังก์ชัน (Function) ส่วนประกอบต่าง ๆ นั้นต้องทำอะไร ซึ่งจำนวนของส่วนประกอบนั้นจะถูกระบุไว้ในข้อกำหนดความต้องการ

5. ส่วนประกอบย่อย (Subordinates) อธิบายว่าส่วนประกอบย่อยนี้เป็นส่วนประกอบของส่วนประกอบใหญ่อันไหนโดยระบุถึงความสัมพันธ์ระหว่างส่วนประกอบนั้น ๆ

6. ความเกี่ยวข้อง (Dependencies) แสดงรายละเอียดเกี่ยวกับความสัมพันธ์กับส่วนประกอบอื่น

7. การติดต่อ (Interface) อธิบายเกี่ยวกับการติดต่อกับส่วนประกอบอื่น ซึ่งจะเกี่ยวกับวิธีการที่ใช้ในการติดต่อ เช่น ต้องติดต่อโดยใช้พารามิเตอร์อะไรและต้องทำอย่างไร

8. ทรัพยากร (Resources) ทรัพยากรที่ต้องการ เช่น หน่วยความจำ เครื่องพิมพ์ รวมทั้งการแก้ปัญหาการแย่งใช้ทรัพยากรหรือการป้องกันการติดตาย (dead lock)

9. การประมวลผล (Processing) คำอธิบายรายละเอียดของอัลกอริธึมที่ใช้ การกำหนดค่าเริ่มต้น และการจัดการเกี่ยวกับข้อยกเว้นต่าง ๆ

10. ข้อมูล (Data) เป็นการอธิบายเกี่ยวกับการใช้ข้อมูล รูปแบบของข้อมูลและความหมายของข้อมูลภายใน [Barnard83] ได้แยกลักษณะของผู้ใช้เอกสารประกอบการออกแบบออกเป็น 7 พวก ตามบทบาทหน้าที่ของแต่ละคนคือ

10.1 ผู้จัดการโครงการ (Project manager) ต้องการข้อมูลที่ใช้ในการวางแผนควบคุม และ จัดการโครงการผู้จัดการโครงการนี้ต้องสามารถที่จะระบุถึงส่วนประกอบต่าง ๆ ของระบบ และ ต้องเข้าใจถึงเป้าหมายตลอดจนหน้าที่ของส่วนประกอบนั้น ๆ

10.2 ผู้จัดการโครงสร้าง (Configuration manager) ต้องการข้อมูลที่ใช้ในการประกอบหรือรวบรวมส่วนประกอบต่าง ๆ ของระบบให้เป็นระบบใหญ่ และใช้ในการควบคุมการเปลี่ยนแปลงได้

10.3 ผู้ออกแบบ (Designer) ต้องการข้อมูลเกี่ยวกับหน้าที่ (function) ในแต่ละส่วนประกอบรวมทั้งการติดต่อระหว่างส่วนประกอบเหล่านั้น

10.4 โปรแกรมเมอร์ (Programmer) ต้องรู้อัลกอริธึมที่จะใช้ โครงสร้างของข้อมูล และชนิดของผลกระทบต่าง ๆระหว่างแต่ละส่วนประกอบ

10.5 ผู้ทดสอบระดับหน่วย (Unit tester) ต้องมีรายละเอียดของข้อมูลที่เกี่ยวกับส่วนประกอบต่าง ๆ เช่น อัลกอริธึมที่ใช้ ค่าเริ่มต้น และข้อมูลที่ต้องการ

10.6 ผู้ทดสอบระดับรวม (Integration tester) ต้องรู้ถึงความสัมพันธ์ระหว่างแต่ละส่วนประกอบ รวมทั้งหน้าที่ของแต่ละส่วนประกอบนั้นตลอดจนการใช้ส่วนประกอบอื่นที่ต้องนำมาเกี่ยวข้อง

10.7 โปรแกรมเมอร์บำรุงรักษา (Maintenance programmer) ต้องมีภาพรวมของความสัมพันธ์ระหว่างแต่ละส่วนประกอบ และ ต้องรู้วิธีในการที่จะทำให้ได้ผลลัพธ์ซึ่งเป็นความต้องการของผู้ใช้

ในขั้นตอนการออกแบบซอฟต์แวร์นี้จะเกิดเอกสารขึ้นดังนี้ คือ

1. เอกสารรายละเอียดความต้องการ (Design specification) สำหรับโครงการใหญ่จะมีทั้งรายละเอียดการออกแบบขั้นต้น (Top level design specification) และรายละเอียดการออกแบบขั้นละเอียด (Detailed design specification)

2. แผนงานการรวมกันของซอฟต์แวร์ (Integration plan)

3. รายละเอียดการทดสอบในกรณีต่าง ๆ ซึ่งจะอธิบายในรายละเอียดแต่ละจุดที่จะใช้การทดสอบระดับต่ำ (low level test)

- **ขั้นตอนการนำซอฟต์แวร์ไปใช้งาน (The Implementation Phase)**

ในขั้นตอนนี้ระบบย่อยต่าง ๆ ของซอฟต์แวร์จะถูกเขียนเป็นโปรแกรม (Coding) และเริ่มต้นทดสอบในแต่ละส่วนย่อย ๆ (Unit Test) ซึ่งการทดสอบในแต่ละส่วนย่อย ๆ จะรับผิดชอบโดยโปรแกรมเมอร์ในแต่ละระบบย่อย ซึ่งจะทำการทดสอบทันทีหลังจากที่เขียนโปรแกรมเสร็จ

วงจรชีวิตในการพัฒนาซอฟต์แวร์ ใช้อัตราส่วนของความพยายาม และเวลาในการพัฒนาเป็น 40-20-40 ซึ่งหมายความว่า จะใช้เวลา 40% ในการกำหนดรายละเอียดความต้องการ และการออกแบบ, เวลา 20% ใช้ในขั้นตอนการนำซอฟต์แวร์ไปใช้งาน นั่นคือ การเขียนโปรแกรม และการทดสอบโปรแกรมส่วนย่อย และเวลา 40% ใช้ในในการรวบรวมซอฟต์แวร์และการทดสอบระบบ แต่แนวโน้มในปัจจุบันมีการลดเวลาในขั้นตอนการนำซอฟต์แวร์ไปใช้ และเพิ่มเวลาเข้าไปในขั้นตอนการกำหนดรายละเอียดความต้องการแทน เพราะว่าขั้นตอนในการพัฒนาซอฟต์แวร์ขั้นตอนแรก ๆ จะใช้ต้นทุนน้อยกว่าขั้นต่อ ๆ มา เพราะฉะนั้นถ้าความพยายามถูกทุ่มเทไปในขั้นตอนการกำหนดรายละเอียดความต้องการ และการออกแบบเมื่อถึงขั้นตอนการรวบรวมซอฟต์แวร์ และการทดสอบระบบจะดำเนินการได้ง่าย และมีประสิทธิภาพมากขึ้น

ขั้นตอนการนำซอฟต์แวร์ไปใช้งานจะรวมกิจกรรมหลักต่าง ๆ ดังนี้ คือ

1. การพัฒนาโปรแกรมซอฟต์แวร์
2. การจัดเตรียมการรวบรวมซอฟต์แวร์ และการทดสอบระบบ (ขั้นตอนต่อไป)
3. การพัฒนาแผนงานการบำรุงรักษาระบบงานซอฟต์แวร์ (Maintenance Plan)

ในขั้นตอนนี้จะเกิดเอกสารสำหรับโครงการขึ้น 3 ชนิด คือ

1. เอกสารบันทึกรายละเอียดต่าง ๆ ของโปรแกรมเมอร์, เอกสารที่เป็นเงื่อนไขในการเขียนโปรแกรม, เอกสารการทดสอบระบบย่อย และเอกสารการแก้ปัญหาที่เกิดขึ้นในขั้นตอนการนำซอฟต์แวร์ไปใช้งาน

2. แผนงานการบำรุงรักษาระบบงานซอฟต์แวร์ (Maintenance Plan) และเอกสารรายละเอียดต่าง ๆ ที่จำเป็นในการบำรุงรักษาระบบงานซอฟต์แวร์

3. เอกสารสำหรับผู้ใช้งานชุดแรก ได้แก่ คู่มือที่เกี่ยวข้องกับการใช้งาน (Reference Manuals) และคู่มือแนะนำการปฏิบัติงาน (Operator Guides)

- **ขั้นตอนการรวบรวมระบบงานซอฟต์แวร์ และการทดสอบ**
  (The Integration and Test Phase)

ในระหว่างขั้นตอนนี้จะมีการรวมระบบย่อย ๆ ของซอฟต์แวร์เข้าเป็นระบบเดี่ยว และทำการทดสอบคุณสมบัติของระบบว่าตรงตามที่ต้องการหรือไม่ และจะมีการประเมินเพื่อแจกแจงปัญหาที่ต้องแก้ไขก่อนที่ระบบจะเสร็จสมบูรณ์

ในขั้นตอนนี้จะจัดเตรียมพื้นฐานสำหรับสิ่งต่าง ๆ ได้แก่

1. การสร้างระบบงานซอฟต์แวร์จากองค์ประกอบของซอฟต์แวร์ต่าง ๆ
2. การรวบรวมระบบงานซอฟต์แวร์เข้ากับอุปกรณ์ฮาร์ดแวร์
3. การตัดสินว่าระบบงานที่พัฒนาขึ้นตรงตามขอบเขตรายละเอียดความต้องการที่ได้กำหนดไว้หรือไม่
4. การสร้างคุณภาพให้กับระบบงาน

ในขั้นตอนนี้มักจะแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ ขั้นตอนการรวบรวมระบบงาน (Integration Phase) และขั้นตอนการทดสอบระบบงาน (Testing Phase) ในโครงการใหญ่ ๆ การแบ่งส่วนเช่นนี้เป็นสิ่งที่จำเป็น โดยเฉพาะเมื่อการทดสอบถูกจัดทำโดยกลุ่มอิสระที่แยกออกต่างหาก (The independent test team) ซึ่งจะทำงานควบคู่ไปกับผู้พัฒนาซอฟต์แวร์ แต่ในโครงการขนาดเล็ก ๆ เหตุผลในการรวมกิจกรรม 2 อย่างนี้เป็นขั้นตอนเดียวก็เนื่องจากว่า การรวบรวมระบบงานซอฟต์แวร์จะดำเนินการให้สำเร็จโดยปราศจากการทดสอบที่ควรจะกระทำควบคู่ไปด้วยกันไม่ได้ เพราะฉะนั้นถ้ากิจกรรม 2 อย่างนี้ถูกจัดทำโดยทีมงานทีมเดียวกันก็ควรจะรวมขั้นตอนให้อยู่ในขั้นตอนเดียวกัน

เหตุผลที่ต้องมีการทดสอบก็เพื่อการทวนสอบ และการตรวจสอบความถูกต้อง (Verification and Validation)

- การทวนสอบ เป็นกระบวนการในการกำหนดหรือยอมรับผลิตภัณฑ์ว่าเป็นไปตามความต้องการที่ได้ระบุไว้ในขั้นตอนก่อน ๆ หรือไม่ กล่าวคือเป็นการหาคำตอบที่ว่า เราได้สร้างความถูกต้องของระบบหรือไม่

- การตรวจสอบความถูกต้อง เป็นกระบวนการในช่วงสุดท้ายของแต่ละขั้นตอนของวงจรการพัฒนาซอฟต์แวร์เพื่อตรวจสอบหรือยอมรับซอฟต์แวร์ว่าเป็นไปตามข้อกำหนดรายละเอียดความต้องการหรือไม่ กล่าวคือเป็นการหาคำตอบที่ว่าเราได้สร้างระบบที่ถูกต้องหรือไม่

<u>เทคนิค และวิธีการในการรวบรวมระบบงาน</u> (Techniques and method of integration) มีดังนี้ คือ

1. เทคนิคจากบนลงล่าง จะเริ่มต้นจากการเตรียมนำระบบงานหลักที่สำคัญ ๆ ไปใช้งานก่อน แล้วจึงค่อย ๆ รวบรวมระบบย่อย ๆ เล็กเข้าไป

2. เทคนิคจากล่างขึ้นบน (bottom up) จะเริ่มต้นจากระบบย่อยเล็ก ๆ ในระบบล่างสุดแล้วค่อย ๆ รวมกันเพิ่มขึ้นเรื่อย ๆ เป็นกลุ่มขนาดใหญ่ขึ้นจนประกอบเป็นระบบทั้งหมดที่สมบูรณ์ เทคนิคจากล่างขึ้นบนมักจะไม่ค่อยแนะนำให้ใช้เป็นกลยุทธ์ในการรวบรวมระบบงานซอฟต์แวร์ ส่วนมากจะแนะนำให้ใช้เทคนิคจากบนลงล่าง เพราะง่ายกว่า และใกล้เคียงกับสภาพความเป็นจริงมากกว่า แต่ในความเป็นจริงการรวบรวมระบบงานซอฟต์แวร์ที่ประสบความสำเร็จมักจะใช้ 2 เทคนิคนี้ร่วมกัน

3. เทคนิคจากภายในสู่ภายนอก (inside out) เป็นเทคนิคที่ถูกนำมาใช้ในการพัฒนาระบบฐานข้อมูลขนาดใหญ่ เริ่มต้นด้วยการสร้างโครงสร้างแฟ้มข้อมูลภายใน (Internal File Structure) ก่อน แล้วตามด้วยการเพิ่มเติมส่วนตรรกศาสตร์ของการประมวลผลข้อมูล (data processing logic) เข้าไปแล้วสุดท้ายก็เพิ่มเติมส่วนการติดต่อใช้งาน (Human interface) เข้าไป เทคนิคนี้จะใช้ได้ดีกับระบบที่ประกอบด้วยส่วนประกอบแยกตามหน้าที่เป็นขั้น ๆ ตามลำดับ แต่ก็มีข้อเสียคือ การนำส่วนการติดต่อใช้งานเข้าไปรวมเป็นลำดับสุดท้าย เลยอาจทำให้ต้องเขียนโปรแกรมชั่วคราวขึ้นมาใช้ทำการทดสอบเพื่อให้สามารถแสดงผลลัพธ์ได้ ทำให้การทดสอบช้าและยาก ขั้นตอนการทดสอบจะเริ่มต้นพร้อมกับการรวบรวมระบบงานซอฟต์แวร์ และดำเนินการไปเรื่อย ๆ จนกระทั่งขั้นตอนสุดท้าย คือ จัดส่งระบบงานให้กับผู้ใช้งาน หรือลูกค้า

ประเภทของการทดสอบมีแบบต่าง ๆ ดังนี้ คือ

1. การทดสอบการรวมกันของระบบ (Integration Testing) ซึ่งดำเนินการโดยผู้รวมรวมระบบ (System Integrators)

2. การทดสอบแบบอิสระ (Independent Testing) ซึ่งดำเนินการโดยกลุ่มผู้ทดสอบภายนอก เพื่อให้มั่นใจว่าการทดสอบระบบนั้นไม่มีอคติ (Unbiased Testing)

3. การทดสอบการติดตั้งระบบ (Installation Testing) รวมถึงการทดสอบผลการปฏิบัติงานโดยทั่วไป เมื่อระบบงานชุดแรกถูกติดตั้งในสภาพแวดล้อมที่ปฏิบัติงานจริง เพื่อให้มั่นใจว่าระบบถูกติดตั้งอย่างสมบูรณ์

4. การทดสอบแบบอัลฟ่า และเบต้า (Alpha and beta testing) การทดสอบนี้จะกระทำภายใต้สภาพแวดล้อมระบบงานจริง ซึ่งการทดสอบแบบอัลฟ่าจะทำสอบระบบโดยไม่มีข้อมูลจริง แต่การทดสอบแบบเบต้าจะทดสอบโดยใช้ข้อมูลจริงตามข้อจำกัดต่าง ๆ เพื่อที่จะได้แก้ไขปัญหาต่าง ๆ ที่อาจเกิดขึ้น

5. การทดสอบการยอมรับระบบงาน (Acceptance Testing) ซึ่งเป็นขั้นตอนขั้นสุดท้ายของโครงการ ซึ่งโครงการจะสำเร็จสมบูรณ์จะสังเกตได้จาก การที่ลูกค้า หรือผู้ใช้ยอมรับผลิตภัณฑ์งานที่พัฒนาขึ้นมา

ในขั้นตอนนี้เอกสารต่าง ๆ ทั้งหมดจะต้องเสร็จสมบูรณ์ และพร้อมที่ที่จะจัดส่ง ซึ่งได้แก่

1. เอกสารการบำรุงรักษาระบบงาน (Maintenance Documentation)
2. เอกสารสำหรับผู้ใช้ที่เสร็จสมบูรณ์ (Final user Documentation)
3. เอกสารการพัฒนาระบบงานทั้งหมดที่ปรับปรุงล่าสุด (All Updated Development Documentation)
4. เอกสารการทดสอบระบบงาน และรายงานการทดสอบระบบ (Test Documentation and Test report)

- **ขั้นตอนการบำรุงรักษาระบบงาน (The Maintenance Phase)**

ขั้นตอนการบำรุงรักษาระบบงานถือเป็นขั้นตอนสุดท้ายของวงจรชีวิตการพัฒนาซอฟต์แวร์ และเป็นจุดเชื่อมต่อระหว่างผลิตภัณฑ์ซอฟต์แวร์ที่เสร็จสมบูรณ์กับการพัฒนาผลิตภัณฑ์ใหม่ คำว่า "การบำรุงรักษาซอฟต์แวร์ (Sostware maintenance) " อาจทำให้เกิดความสับสนเพราะทำให้เข้าใจว่าหมายถึงความต้องการที่จะซ่อมบำรุงผลิตภัณฑ์ที่เสื่อมคุณภาพ หรือต้องการจะเปลี่ยนชิ้นส่วนที่เสีย เช่น ระบบทางด้านไฟฟ้า หรือเครื่องยนต์ แต่ซอฟต์แวร์ไม่เป็นเช่นนั้น เพราะตัวซอฟต์แวร์จะไม่เปลี่ยแปลงหรือเสื่อมสภาพ ถ้าไม่มีคนเข้าเข้าไปเกี่ยวข้อง

IEEE ได้ให้คำจำกัดความของคำว่า "การบำรุงรักษาซอฟต์แวร์" ว่าหมายถึงการแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นในซอฟต์แวร์ก่อนการจัดส่ง ซึ่งรวมถึงการเปลี่ยนแปลงเพื่อปรับปรุงคุณภาพ หรือการปรับสภาพผลิตภัณฑ์ให้เข้ากับสภาพแวดล้อมที่เปลี่ยนแปลงไป

กิจกรรมที่เป็นส่วนหนึ่งของการบำรุงรักษา ได้แก่

1. การปรับปรุงเอกสารต่าง ๆ ให้ทันสมัยอยู่เสมอ
2. การปรับปรุงหลักสูตรการฝึกอบรมให้แก่ผู้ใช้
3. การปรับปรุงแก้ไขซอฟต์แวร์ (Modifying Software)
4. การควบคุมขอบเขตโครงร่างของซอฟต์แวร์ (Configuration Control)

การปรับปรุงแก้ไขซอฟต์แวร์จะรวมกิจกรรมต่าง ๆ เช่นเดี่ยวกับการพัฒนาซอฟต์แวร์ ไม่ว่าจะเป็นการรวบรวมรายละเอียดที่ต้องการอย่างเป็นทางการ การออกแบบ การนำไปใช้งาน การรวบรวม และทดสอบซึ่งต้องใช้งบประมาณในการดำเนินงานด้วย เปรียบเสมือนว่าเป็นโครงการซอฟต์แวร์เล็ก ๆ โครงการหนึ่ง

การควบคุมขอบเขตโครงร่างของซอฟต์แวร์ เป็นงานที่สำคัญในขั้นตอนนี้เพราะจะควบคุมจัดการเกี่ยวกับการเปลี่ยนแปลงต่าง ๆ ของซอฟต์แวร์ และควบคุมเกี่ยวกับชุดของซอฟต์แวร์ (Release and Version of Software) ที่ออกมาเป็นลำดับก่อนหลัง จะได้สะดวกในการสนับสนุนงานต่าง ๆ

ขั้นตอนการบำรุงรักษาระบบงานนี้ต้องการทีมงานขนาดเล็ก ซึ่งอาจจัดตั้งขึ้นมาเพื่อบำรุงรักษาผลิตภัณฑ์ซอฟต์แวร์หลาย ๆ ชนิด โดยจัดการดูแลเกี่ยวกับการควบคุมรายละเอียดขอบเขตโครงร่างของซอฟต์แวร์ (Configuration control) การติดตั้งระบบ (Installation) และงานด้านวิศวกรรมซอฟต์แวร์ รวมถึงการบำรุงรักษาเอกสารต่าง ๆ ของระบบงานด้วย

เอกสารต่าง ๆ ที่ต้องปรับปรุงในระหว่างขั้นตอนนี้ ได้แก่

1. เอกสารรายละเอียดชุดของซอฟต์แวร์ (Version release documentation)
2. รายงานปัญหาต่าง ๆ (Problem reports)
3. เอกสารการพัฒนาระบบงานทั้งหมด (All Development documentation)
4. เอกสารคู่มือการใช้งาน (User documentation)

5. บันทึกรายละเอียดการบำรุงรักษา (Maintenance logs) และรายงานการบริการลูกค้า (Customer service reports)