

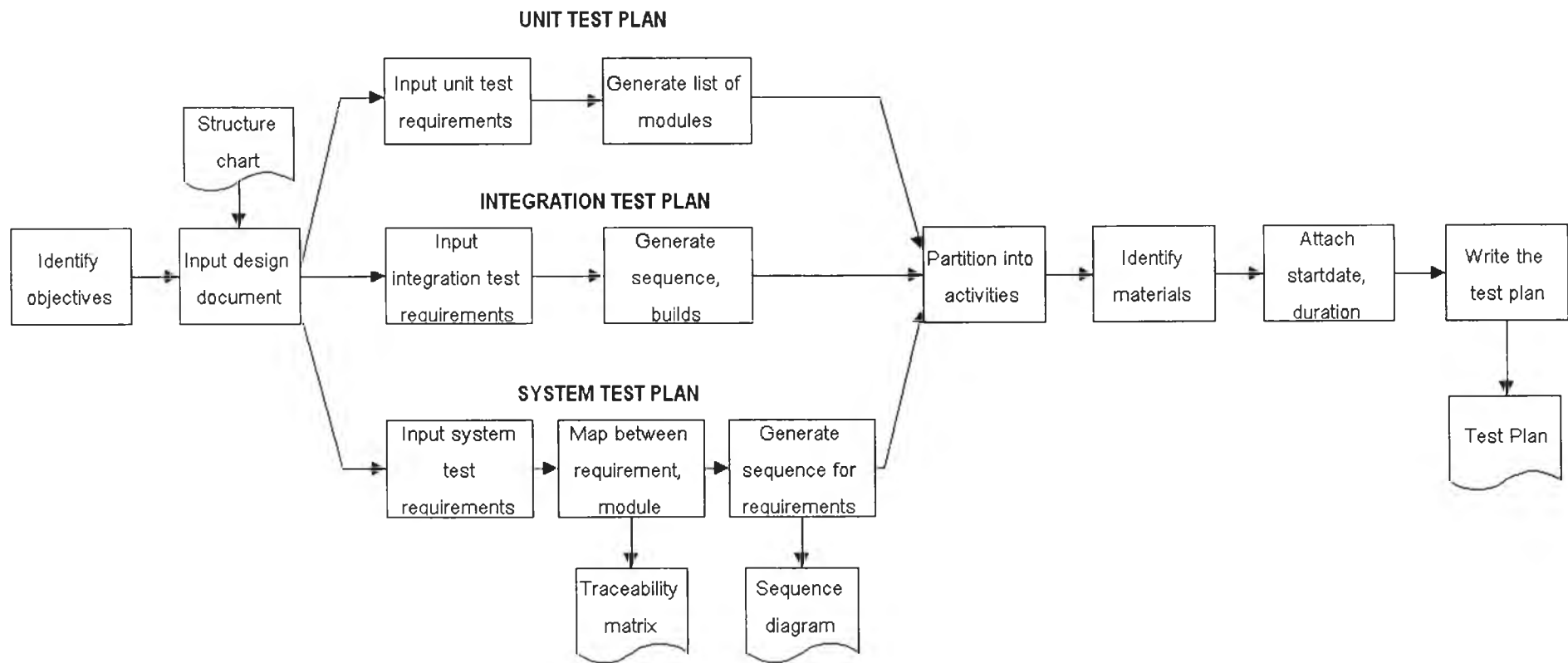
บทที่ 3

แนวความคิดในการพัฒนาเครื่องมือเพื่อจัดการการวางแผนการทดสอบซอฟต์แวร์

3.1 การกำหนดลำดับขั้นตอนของการวางแผนการทดสอบในแต่ละระดับ

ในกระบวนการวางแผนการทดสอบซอฟต์แวร์ขององค์กร ประกอบด้วยกิจกรรมต่างๆ ที่ต้องทำหลายขั้นตอน[3] เพื่อให้สามารถจัดการทดสอบให้ดำเนินไปได้ตามจุดมุ่งหมายและเพื่อให้มีการนำทรัพยากรมาใช้อย่างคุ้มค่า อีกทั้งการทดสอบในแต่ละระดับมีจุดมุ่งหมายที่แตกต่างกัน ดังนั้นจึงจำเป็นต้องมีการกำหนดลำดับขั้นตอนของการวางแผนการทดสอบในแต่ละระดับ เพื่อให้สามารถติดตามและควบคุมได้ ซึ่งผู้วิจัยได้เสนอลำดับขั้นตอนของการวางแผนการทดสอบในระดับต่าง ๆ ประกอบด้วย การทดสอบระดับหน่วย การทดสอบการรวม และการทดสอบระบบ ซึ่งสามารถแสดงได้ดังรูปที่ 3.1 โดยมีรายละเอียดดังนี้

1. การทดสอบระดับหน่วย มีจุดมุ่งหมายหลักเพื่อพยายามทดสอบทุก ๆ โมดูลที่มีทั้งหมดในโปรแกรม มีขั้นตอนดังนี้
 - 1.1 กำหนดวัตถุประสงค์ของการทดสอบในระดับหน่วย
 - 1.2 ระบุโมดูลที่มีทั้งหมดใน โปรแกรมเพื่อทำการทดสอบ โดยอาจนำข้อมูลมาจากเอกสารที่เกี่ยวข้องกับการออกแบบระบบในขั้นตอนการพัฒนา อันได้แก่ แผนภูมิโครงสร้าง เป็นต้น
 - 1.3 สร้างลำดับการทดสอบของโมดูลที่รับเข้ามา ซึ่งจะให้เป็นกิจกรรมที่ต้องทำในการทดสอบ
 - 1.4 ระบุความต้องการอื่น ๆ ที่มีในการทดสอบระดับหน่วย และระบุกิจกรรมเพื่อใช้ทดสอบความต้องการนั้น
 - 1.5 แบ่งกิจกรรมที่ได้ทั้งหมดเป็นกิจกรรมย่อย ๆ โดยพยายามให้ 1 กิจกรรมเป็น 1 การทดสอบโมดูล เพื่อให้สามารถจัดการและควบคุมได้ง่าย
 - 1.6 กำหนดทรัพยากรที่ต้องใช้สำหรับแต่ละกิจกรรมนั้น อันได้แก่ ฮาร์ดแวร์ ซอฟต์แวร์(ตัวขับและตัวดำเนินการ) และผู้ทดสอบ
 - 1.7 กำหนดวันที่เริ่มต้น และระยะเวลาที่ใช้สำหรับแต่ละกิจกรรม
 - 1.8 จัดทำเอกสารการทดสอบระดับหน่วย
2. การทดสอบการรวม มีจุดมุ่งหมายเพื่อพยายามทดสอบการเชื่อมต่อและการทำงานร่วมกันระหว่างโมดูล มีขั้นตอนดังนี้
 - 2.1 กำหนดวัตถุประสงค์ของการทดสอบการรวม
 - 2.2 ระบุโมดูลที่มีทั้งหมดใน โปรแกรมเพื่อทำการทดสอบ
 - 2.3 สร้างลำดับการรวมโมดูล โดยเลือกใช้วิธีการรวมแบบต่าง ๆ เช่น การรวมจากบนลงล่าง การรวมจากล่างขึ้นบน เป็นต้น ซึ่งจะให้เป็นกิจกรรมที่ต้องทำในการทดสอบ
 - 2.4 ระบุความต้องการอื่น ๆ ที่มีในการทดสอบการรวม และระบุกิจกรรมเพื่อใช้ทดสอบความต้องการนั้น
 - 2.5 แบ่งกิจกรรมที่ได้ทั้งหมดเป็นกิจกรรมย่อย ๆ เพื่อให้สามารถจัดการและควบคุมได้ง่าย



รูปที่ 3.1 ลำดับขั้นตอนของการวางแผนการทดสอบในระดับต่าง ๆ

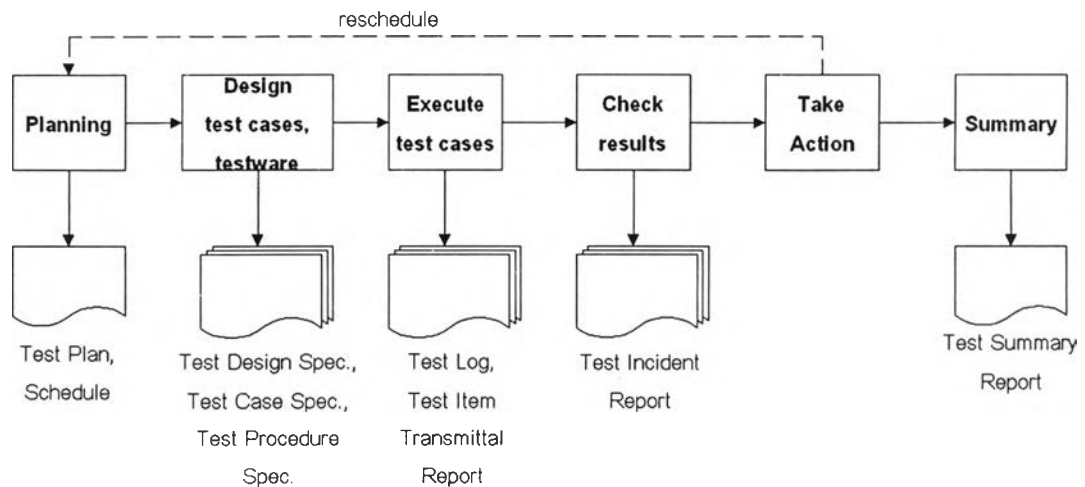
- 2.6 กำหนดทรัพยากรที่ต้องใช้สำหรับแต่ละกิจกรรมนั้น
- 2.7 กำหนดวันที่เริ่มต้น และระยะเวลาที่ใช้สำหรับแต่ละกิจกรรม
- 2.8 จัดทำเอกสารการทดสอบการรวม
3. การทดสอบระบบ มีจุดมุ่งหมายเพื่อพยายามทดสอบระบบให้ตรงกับความต้องการของผู้ใช้ มีขั้นตอนดังนี้
 - 3.1 กำหนดวัตถุประสงค์ของการทดสอบระบบ
 - 3.2 ระบุโมดูลที่มีทั้งหมดในโปรแกรมเพื่อทำการทดสอบ
 - 3.3 ระบุความต้องการของผู้ใช้ในการทดสอบระบบ ซึ่งจะอยู่ในรูปของข้อความที่อธิบายความสามารถของโปรแกรม
 - 3.4 กำหนดความสัมพันธ์ของความต้องการในแต่ละข้อกับโมดูลที่มีทั้งหมด อาจได้มาจากขั้นตอนการเก็บรวบรวมความต้องการของผู้ใช้หรือขั้นตอนการออกแบบระบบได้ เพื่อให้ทราบว่า หากต้องการทดสอบความต้องการนี้ จะต้องพิจารณาในโมดูลใด เมื่อกำหนดความสัมพันธ์แล้วจะได้เป็น เมตริกซ์ที่ติดตามได้ (Traceability Matrix)
 - 3.5 กำหนดลำดับการทดสอบความต้องการแต่ละข้อ เนื่องจากอาจมีการขึ้นต่อกันในระหว่างการทดสอบ ซึ่งจะได้ผลลัพธ์เป็นแผนผังลำดับ (Sequence Diagram)
 - 3.6 จากความต้องการที่ได้ พยายามสร้างเป็นกิจกรรมย่อย ๆ เพื่อรองรับความต้องการนั้น เพื่อให้สามารถจัดการและควบคุมได้
 - 3.7 กำหนดทรัพยากรที่ต้องใช้สำหรับแต่ละกิจกรรม
 - 3.8 กำหนดวันที่เริ่มต้น และระยะเวลาที่ใช้สำหรับแต่ละกิจกรรม
 - 3.9 จัดทำเอกสารการทดสอบระบบ

เมื่อองค์กรได้แผนการทดสอบซอฟต์แวร์ที่ระดับต่าง ๆ แล้ว องค์กรจำเป็นต้องทำกิจกรรมเพื่อให้บรรลุตามแผนการทดสอบที่ได้กำหนดไว้ ซึ่งในแต่ละกิจกรรมจะเกิดเอกสารต่าง ๆ ขึ้นดังแสดงในรูปที่

3.2 โดยอิงกับมาตรฐาน IEEE Std.829-1998 [4] ประกอบด้วยกิจกรรมต่าง ๆ ดังนี้

1. การวางแผนการทดสอบ (Test Planning) โดยทำตามขั้นตอนในรูปที่ 3.1 จะได้แผนการทดสอบ (Test Plan) และกำหนดการทดสอบ (Schedule)
2. การออกแบบกรณีทดสอบ (Design test cases and testware) โดยเลือกวิธีการทดสอบที่ต้องการแล้วนำมาสร้างเป็นชั้นสมมูลให้ครอบคลุมทุกกรณี จากนั้นจึงนำชั้นสมมูลที่ได้มาออกแบบเป็นกรณีทดสอบ จะได้เอกสารการออกแบบการทดสอบ (Test Design Specification) เอกสารเกี่ยวกับกรณีทดสอบ (Test Case Specification) และเอกสารเกี่ยวกับลำดับการทดสอบ (Test Procedure Specification)
3. การดำเนินการทดสอบ (Execute test cases) โดยผู้ทดสอบนำกรณีทดสอบที่ได้ไปทดสอบและบันทึกผลการทดสอบไว้ จะได้บันทึกการทดสอบ (Test Log) และรายงานส่วนของโปรแกรมที่ถูกส่งไปทดสอบ (Test Item Transmittal Report)

4. การตรวจสอบผลลัพธ์ (Check results) โดยตรวจสอบผลลัพธ์ที่ได้จากการทดสอบว่าตรงกับผลลัพธ์ที่คาดว่าจะได้รับหรือไม่ จะได้รายงานการทดสอบที่ต้องดำเนินการต่อ (Test Incident Report)
5. การดำเนินการแก้ไข (Take action) โดยการวิเคราะห์ข้อผิดพลาดที่เกิดจากการทดสอบ แล้วหาแนวทางแก้ไข เช่น ส่งกลับไปให้กลุ่มผู้พัฒนาทำการแก้ไข โปรแกรม หรือปรับกำหนดการใหม่ เป็นต้น
6. การสรุปผลการทดสอบ (Summary) โดยตรวจสอบประสิทธิภาพของการทดสอบ ซึ่งพิจารณาจากกรณีทดสอบทั้งหมด กรณีทดสอบที่เกิดข้อผิดพลาด และกรณีทดสอบที่ได้รับการแก้ไข จะได้รายงานสรุปผลการทดสอบ (Test Summary Report)



รูปที่ 3.2 เอกสารที่เกิดขึ้นจากการทดสอบ

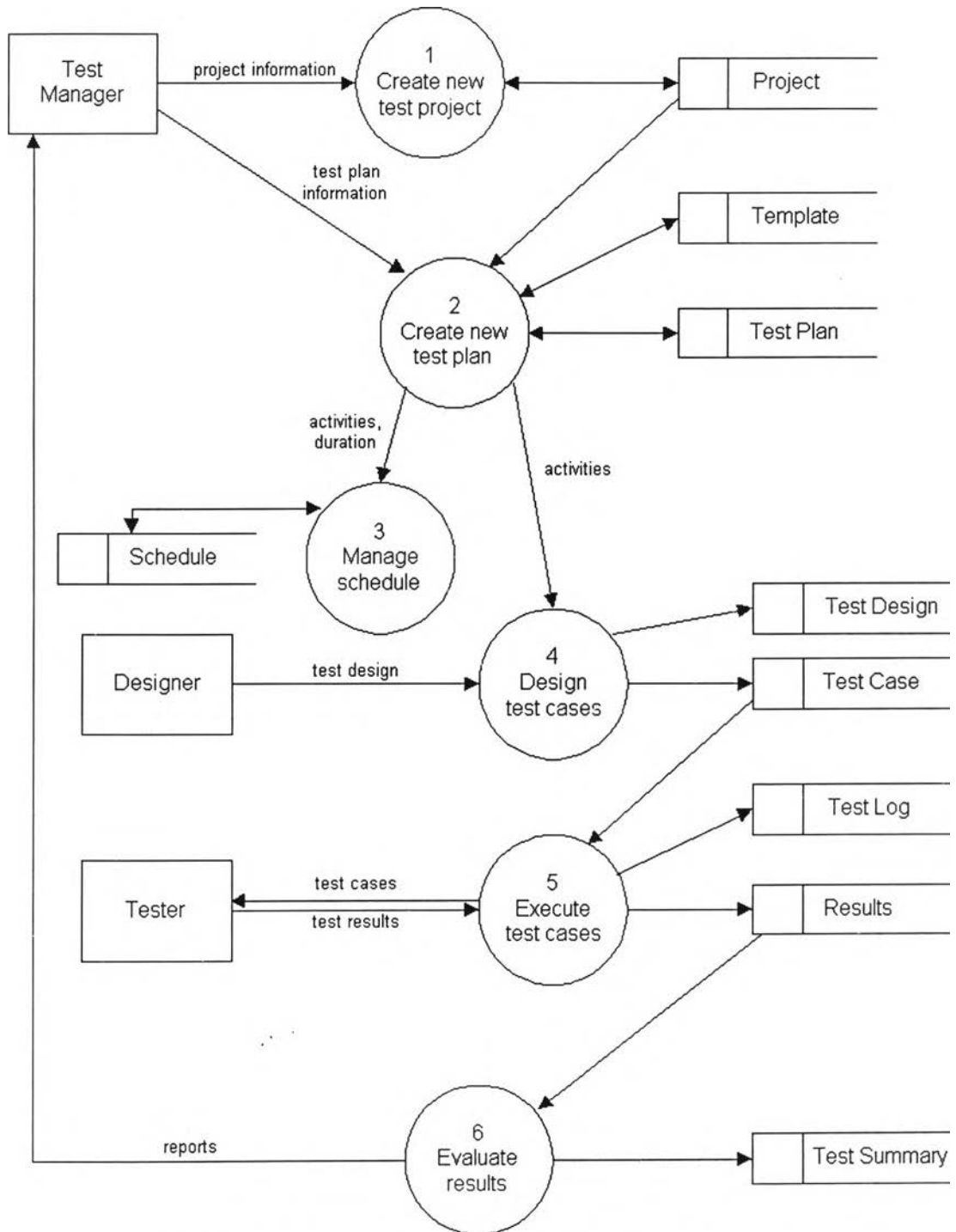
3.2 การออกแบบระบบฐานข้อมูลสำหรับเครื่องมือจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์

เมื่อออกแบบขั้นตอนการทดสอบในแต่ละระดับแล้ว ขั้นตอนต่อไปจะเป็นการออกแบบระบบฐานข้อมูลเพื่อนำมาบันทึกข้อมูลต่าง ๆ ระหว่างการทดสอบ ซึ่งสามารถแสดงการไหลของข้อมูลได้ดังแผนภาพกระแสข้อมูลระดับที่ 0 ในรูปที่ 3.3 โครงการทดสอบหนึ่ง ๆ สามารถถูกจัดตั้งขึ้นโดยผู้จัดการทดสอบหนึ่งหรือหลายคนได้ เมื่อสร้างโครงการแล้วจึงดำเนินการสร้างแผนการทดสอบในแต่ละระดับเพื่อให้ได้เป็นกิจกรรมต่าง ๆ ที่ต้องทำ จากนั้นจึงมอบหมายแต่ละกิจกรรมให้กับผู้ออกแบบกรณีทดสอบและผู้ทดสอบเพื่อดำเนินการทดสอบจนเสร็จสิ้น แล้วจัดทำเป็นสรุปผลการทดสอบกลับไปยังผู้จัดการทดสอบ

ผู้ใช้เครื่องมือสามารถแบ่งออกเป็น 3 กลุ่มได้แก่

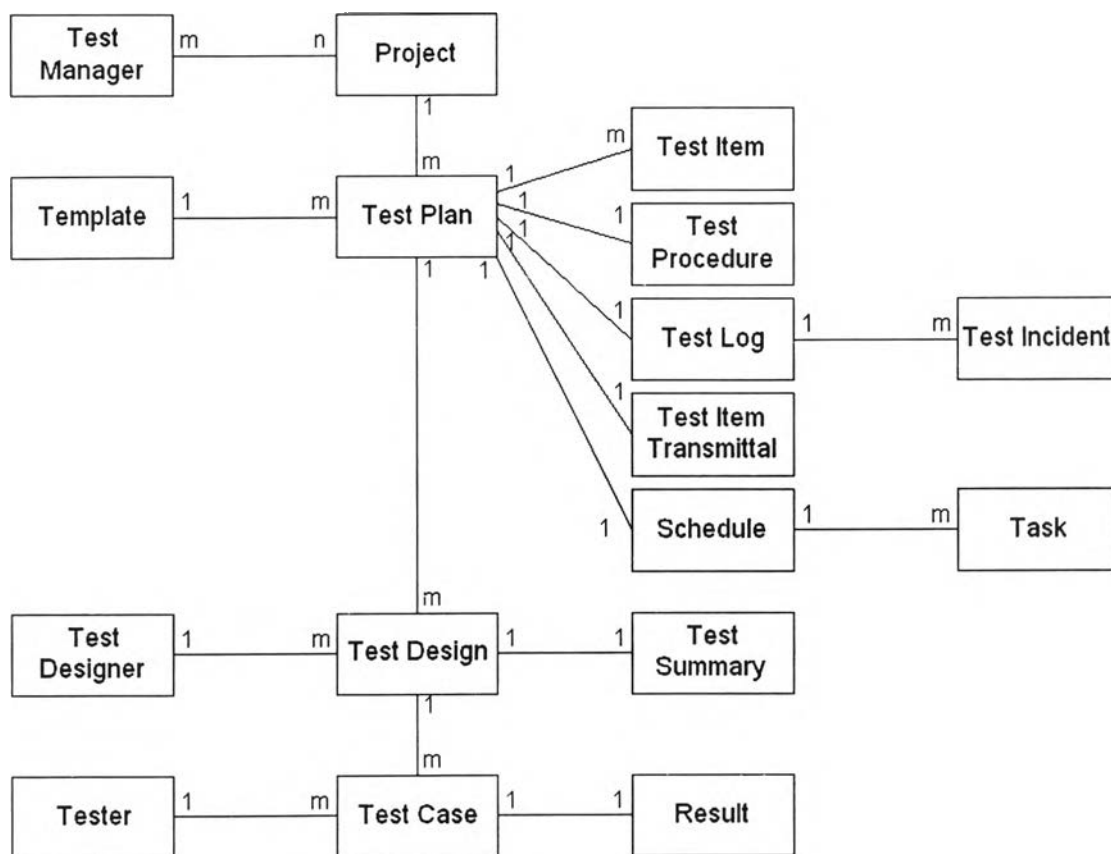
- 1) ผู้จัดการทดสอบ (Test Manager) เป็นผู้ริเริ่มโครงการทดสอบ กำหนดวัตถุประสงค์และเป้าหมายของโครงการ กำหนดสภาพแวดล้อมที่ใช้ สร้างกิจกรรมรวมทั้งจัดกำหนดการเพื่อให้การทดสอบนั้นบรรลุตามวัตถุประสงค์

- 2) ผู้ออกแบบการทดสอบ (Test Designer) เป็นผู้ออกแบบกรณีทดสอบสำหรับแต่ละกิจกรรมที่ถูกสร้างโดยผู้จัดการ เพื่อนำไปให้ผู้ทดสอบดำเนินการทดสอบต่อไป
- 3) ผู้ทดสอบ (Tester) เป็นผู้ดำเนินการทดสอบกรณีทดสอบที่สร้างขึ้น และบันทึกผลที่ได้เพื่อนำไปแก้ไขต่อไป



รูปที่ 3.3 แผนภาพกระแสข้อมูลระดับที่ 0 ของเครื่องมือจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์

เมื่อได้แผนภาพกระแสข้อมูลของเครื่องมือแล้ว จากนั้นจะทำการวิเคราะห์ความสัมพันธ์ระหว่างข้อมูล ซึ่งแสดงได้ดังแผนภาพเอนทิตีและความสัมพันธ์ในรูปที่ 3.4



รูปที่ 3.4 แผนภาพเอนทิตีและความสัมพันธ์ของเครื่องมือจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์