

การพัฒนามาตรวัดของการทดสอบและเครื่องมือสำหรับทดสอบทักษะการคิด

นาย ประภาส ศรีชัยวัฒน์



สถาบันวิทยบริการ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย


ปีการศึกษา 2542

ISBN 974-333-504-8

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A DEVELOPMENT OF TEST-RELATED METRICS AND A TOOL
FOR TESTING MATURITY MODEL

MR.PRAPASS SRICHAIVATTANA



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering


Chulalongkorn University

Academic Year 1999

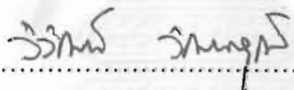
ISBN 974-333-504-8


หัวข้อวิทยานิพนธ์ การพัฒนามาตรวัดของการทดสอบและเครื่องมือสำหรับทดสอบมาทัวริตี
โมเดล
โดย นาย ประภาส ศรีชัยวัฒน์
ภาควิชา วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา อาจารย์ ดร. ชาราทิพย์ สุวรรณศาสตร์

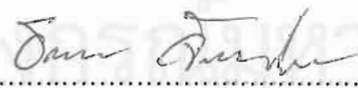
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาโท


..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. ชัชชัย สุมิตร)

คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(อาจารย์ วิวัฒน์ วัฒนาวุฒิ)


..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร. ชาราทิพย์ สุวรรณศาสตร์)


..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ชนาวรรณ จันทร์ตนไพบูลย์)


..... กรรมการ
(อาจารย์ ดร. อาทิตย์ ทองทักษ์)

ประกาศ ศรีชัยวัฒน์: การพัฒนามาตรวัดของการทดสอบและเครื่องมือสำหรับทดสอบ
มาทัวริตีโมเดล (A DEVELOPMENT OF TEST-RELATED METRICS AND A TOOL
FOR TESTING MATURITY MODEL) อาจารย์ที่ปรึกษา: ดร. ชาราทิพย์ สุวรรณศาสตร์,
143 หน้า, ISBN 974-333-504-8.

วิทยานิพนธ์นี้นำเสนอมาตรวัดของการทดสอบสำหรับทดสอบมาทัวริตีโมเดล (ทีเอ็มเอ็ม)
ในแต่ละระดับตั้งแต่ระดับ 2 ถึง 5 จุดประสงค์ของมาตรวัด คือ ใช้เป็นข้อมูลในการติดตามและควบคุม
กระบวนการทดสอบ ซึ่งเป็นประโยชน์ต่อองค์กรพัฒนาซอฟต์แวร์ในการปรับปรุงกระบวนการ
ทดสอบ มาตรวัดที่นำเสนอได้มาจากวิธีเป้าหมาย/คำถาม/มาตรวัด หรือจิกิวเอ็ม โดยเป้าหมายจะมาจาก
เป้าหมายในแต่ละระดับของทีเอ็มเอ็ม งานวิจัยได้ตั้งคำถามเพื่อให้ได้มาตรวัดที่สะท้อนให้เห็น
ถึงเป้าหมายนั้นๆ ซึ่งมาตรวัดที่ได้แบ่งเป็น 2 ประเภทคือ มาตรวัดขั้นต่ำที่จำเป็นต้องมีและมาตรวัด
ที่จะมีหรือไม่มีก็ได้ งานวิจัยได้เสนอเพียงมาตรวัดที่องค์กรควรจะมี แต่ไม่ได้กำหนดถึงวิธีการที่
จะเก็บข้อมูลและความเหมาะสมของค่าของมาตรวัด

งานวิจัยได้ออกแบบและพัฒนาเครื่องมือเพื่อเก็บข้อมูลมาตรวัดในแต่ละระดับตามที่ได้
กำหนดขึ้นมา และนำข้อมูลที่เก็บไว้มาออกรายงานเพื่อใช้ในการวิเคราะห์และติดตามกระบวนการ
ทดสอบ การออกแบบเครื่องมือประกอบด้วยการออกแบบฐานข้อมูล การออกแบบอินพุตและการ
ออกแบบเอาต์พุต จากนั้นทำการพัฒนาเครื่องมือตามที่ได้ออกแบบและทดสอบเครื่องมือโดยการ
ป้อนข้อมูลเข้าไปในหน้าจอและออกรายงานต่างๆ ซึ่งผลการทดสอบเป็นไปตามที่ออกแบบไว้อย่าง
ถูกต้อง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์
ปีการศึกษา 2542

ลายมือชื่อนิสิต รุ่งทรง ทรัพย์รุ่ง
ลายมือชื่ออาจารย์ที่ปรึกษา จ.พงษ์ สุพรรณิการ์
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม _____

3970972121 : MAJOR COMPUTER SCIENCE

KEY WORD : SOFTWARE TESTING/TMM/MEASUREMENT/METRICS

PRAPASS SRICHAIVATTANA : A DEVELOPMENT OF TEST-RELATED
METRICS AND A TOOL FOR TESTING MATURITY MODEL. THESIS

ADVISOR: TARATIP SUWANNASART, Ph.D. 143 PP., ISBN 974-333-504-8.

This thesis presents a set of measurements related to test-related data for each level of the Testing Maturity Model (TMM), from level 2 to 5. The objective of measurements is to track and control testing practices status that will help software development organizations improve their testing processes. The set of measurements is derived from the Goal/Question/Metric (GQM) paradigm. Goals come from the goals in each level of the TMM. The thesis sets the questions and derives the metrics to reflect the goals. The set of measurements is classified into two types, minimum and optional metric set. The thesis defines only what to be collected, not how to collect and does not indicate appropriate values of measurements.

A tool is designed and developed to collect the measurement data. The tool uses these data to provide reports in analyzing and monitoring testing processes. The design of the tool consists of database, input and output design. The tool is developed and tested by inputting data through screens and outputting reports. The test results are satisfied as the design.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์

ปีการศึกษา 2542

ลายมือชื่อนิสิต ประภากร พิริทธิ์ พิริทธิ์

ลายมือชื่ออาจารย์ที่ปรึกษา จ.เทพย์ สุคนธ์จิตร

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม _____



กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ด้วยความช่วยเหลืออย่างดียิ่งของ ดร. ธาราทิพย์ สุวรรณศาสตร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ ในการวิจัยมาด้วยดีตลอด

ขอขอบคุณท่านอาจารย์และเจ้าหน้าที่จากภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำและช่วยเหลือ ตลอดจนเพื่อนและน้องๆ ที่คอยช่วยเหลือและให้กำลังใจมา โดยตลอด

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา-มารดา และบุคคลในครอบครัวของผู้วิจัยซึ่งคอยให้กำลังใจแก่ผู้วิจัยเสมอจนสำเร็จการศึกษา

ประภาส ศรีชัยวัฒน์

มกราคม 2543

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฌ
สารบัญภาพ	ญ
บทที่	
1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตการวิจัย	2
1.4 ขั้นตอนและวิธีการดำเนินการวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 โครงร่างของวิทยานิพนธ์	3
1.7 ผลงานตีพิมพ์	3
2 งานวิจัย แนวคิดและทฤษฎีที่เกี่ยวข้อง	
2.1 แนวคิดเกี่ยวกับการทดสอบ	4
2.2 ทีเอ็มเอ็ม	5
2.3 แนวคิดเกี่ยวกับการวัดซอฟต์แวร์	8
2.4 วิธีการที่ใช้ในการได้มาตรวัดสำหรับทีเอ็มเอ็ม	10
3 มาตรวัดในแต่ละระดับของทีเอ็มเอ็ม	
3.1 มาตรวัดในระดับ 2 การกำหนดเป็นขั้นตอน	12
3.2 มาตรวัดในระดับ 3 การรวม	16
3.3 มาตรวัดในระดับ 4 การจัดการและการวัด	23
3.4 มาตรวัดในระดับ 5 การทำให้มีผลดีที่สุด การป้องกัน	30
ข้อบกพร่อง และ การควบคุมคุณภาพ	

สารบัญ (ต่อ)

บทที่	หน้า
4 การออกแบบและพัฒนาเครื่องมือในการเก็บมาตรวัด	
4.1 การออกแบบฐานข้อมูลเพื่อเก็บมาตรวัด	35
4.2 การออกแบบอินพุตและเอาต์พุต	40
4.2.1 เมนูจัดเตรียม	40
4.2.2 หน้าจอและรายงานของมาตรวัดในระดับ 2	45
4.2.3 หน้าจอและรายงานของมาตรวัดในระดับ 3	60
4.2.4 หน้าจอและรายงานของมาตรวัดในระดับ 4	82
4.2.5 หน้าจอและรายงานของมาตรวัดในระดับ 5	96
4.3 การทดสอบเครื่องมือ	110
5 สรุปผลการวิจัยและข้อเสนอแนะ	111
รายการอ้างอิง	113
ภาคผนวก	
ก บททความวิชาการที่ได้รับการตีพิมพ์	115
ข ตารางต่างๆ ในฐานข้อมูล	120
ประวัติผู้เขียน	143

สารบัญตาราง

	หน้า
ตาราง 3.1 มาตรฐานขั้นต่ำสำหรับทีเอ็มเอ็มในระดับ 2	16
ตาราง 3.2 มาตรฐานขั้นต่ำสำหรับทีเอ็มเอ็มในระดับ 3	22
ตาราง 3.3 มาตรฐานสำหรับคุณสมบัติด้านคุณภาพของผลิตภัณฑ์ซอฟต์แวร์	28
ตาราง 3.4 มาตรฐานขั้นต่ำสำหรับทีเอ็มเอ็มในระดับ 4	29
ตาราง 3.5 มาตรฐานขั้นต่ำสำหรับทีเอ็มเอ็มในระดับ 5	34



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญญภาพ

	หน้า
รูปที่ 2.1 โมเดลเอ็กซ์เทนค์ / โมดิฟายส์ วี	4
รูปที่ 2.2 โครงสร้างที่เอ็มเอ็มแต่ละระดับ	5
รูปที่ 2.3 เป้าหมายที่เอ็มเอ็มในแต่ละระดับ	6
รูปที่ 2.4 สถาปัตยกรรมของกระบวนการการวัดซอฟต์แวร์	9
รูปที่ 4.1 แสดงความสัมพันธ์ของตารางต่างๆ ในฐานข้อมูลสำหรับเก็บ ข้อมูลมาตรวัดของทีเอ็มเอ็ม	36
รูปที่ 4.2 ตัวอย่างหน้าจอรายละเอียดซอฟต์แวร์	40
รูปที่ 4.3 ตัวอย่างหน้าจอบุคลากรในกระบวนการทดสอบ	40
รูปที่ 4.4 โครงสร้างเมนูในระดับ 2	41
รูปที่ 4.5 โครงสร้างเมนูในระดับ 3	42
รูปที่ 4.6 โครงสร้างเมนูในระดับ 4	42
รูปที่ 4.7 โครงสร้างเมนูในระดับ 5	42
รูปที่ 4.8 โครงสร้างเมนูรายงาน	43
รูปที่ 4.9 ตัวอย่างหน้าจอการทดสอบโมดูลในระดับ 2	45
รูปที่ 4.10 ตัวอย่างหน้าจอรายละเอียดกรณีทดสอบในระดับ 2	46
รูปที่ 4.11 ตัวอย่างหน้าจอขั้นตอนปฏิบัติการทดสอบในระดับ 2	46
รูปที่ 4.12 ตัวอย่างหน้าจอข้อบกพร่องในระดับ 2	47
รูปที่ 4.13 ตัวอย่างหน้าจอความพยายามในการพัฒนาแผนงานการทดสอบโมดูลในระดับ 2 ...	47
รูปที่ 4.14 ตัวอย่างหน้าจอแผนงานการทดสอบโมดูลในระดับ 2	48
รูปที่ 4.15 ตัวอย่างหน้าจอแผนงานการทดสอบเบ็ดเสร็จในระดับ 2	49
รูปที่ 4.16 ตัวอย่างหน้าจอแผนงานการทดสอบระบบในระดับ 2	49
รูปที่ 4.17 ตัวอย่างหน้าจอแผนงานการทดสอบเพื่อยอมรับในระดับ 2	50
รูปที่ 4.18 ตัวอย่างหน้าจอเทคนิคการทดสอบในระดับ 2	50
รูปที่ 4.19 ตัวอย่างรายงานค่าใช้จ่ายการทดสอบ โมดูล	51
รูปที่ 4.20 ตัวอย่างรายงานค่าใช้จ่ายในการพัฒนากรณีทดสอบ	52
รูปที่ 4.21 ตัวอย่างรายงานค่าใช้จ่ายในการพัฒนาแผนงานการทดสอบ	52

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.22 ตัวอย่างรายงานสรุปค่าใช้จ่ายในการทดสอบของโครงการ	53
รูปที่ 4.23 ตัวอย่างรายงานสรุปค่าใช้จ่ายในการทดสอบในระดับองค์กร	54
รูปที่ 4.24 ตัวอย่างรายงานค่าใช้จ่ายในการแก้ไขข้อบกพร่อง	55
รูปที่ 4.25 ตัวอย่างรายงานสรุปค่าใช้จ่ายในการแก้ไขข้อบกพร่องในระดับองค์กร	55
รูปที่ 4.26 ตัวอย่างรายงานกิจกรรมของแผนงานการทดสอบโมดูล	56
รูปที่ 4.27 ตัวอย่างรายงานกิจกรรมของแผนงานการทดสอบเบ็ดเสร็จ	56
รูปที่ 4.28 ตัวอย่างรายงานกิจกรรมของแผนงานการทดสอบระบบ	57
รูปที่ 4.29 ตัวอย่างรายงานสรุปแผนงานการทดสอบ	57
รูปที่ 4.30 ตัวอย่างรายงานการใช้เทคนิคในการพัฒนากรณีทดสอบ	58
รูปที่ 4.31 ตัวอย่างรายงานการใช้เทคนิคการทดสอบในการตรวจพบข้อบกพร่อง	59
รูปที่ 4.32 ตัวอย่างหน้าจอบุคลากรในหน่วยงานการทดสอบซอฟต์แวร์	60
รูปที่ 4.33 ตัวอย่างหน้าจอค่าใช้จ่ายของหน่วยงานการทดสอบ	60
รูปที่ 4.34 ตัวอย่างหน้าจอรายละเอียดการฝึกอบรมด้านการทดสอบ	61
รูปที่ 4.35 ตัวอย่างหน้าจอรายละเอียดข้อบกพร่องในระดับ 3	61
รูปที่ 4.36 ตัวอย่างหน้าจอการทดสอบ โมดูลในระดับ 3	62
รูปที่ 4.37 ตัวอย่างหน้าจอสถานะกรณีทดสอบ	62
รูปที่ 4.38 ตัวอย่างหน้าจอการทดสอบเบ็ดเสร็จในระดับ 3	63
รูปที่ 4.39 ตัวอย่างหน้าจอผลลัพธ์ของแต่ละ โมดูลที่ทดสอบร่วมกัน	63
รูปที่ 4.40 ตัวอย่างหน้าจอการทดสอบระบบในระดับ 3	64
รูปที่ 4.41 ตัวอย่างหน้าจอผลลัพธ์การทดสอบความต้องการในการทดสอบระบบ	64
รูปที่ 4.42 ตัวอย่างหน้าจอการทดสอบเพื่อยอมรับในระดับ 3	65
รูปที่ 4.43 ตัวอย่างรายงานจำนวนผู้ทดสอบและค่าใช้จ่ายของหน่วยงานการทดสอบ	66
รูปที่ 4.44 ตัวอย่างรายงานสถานะของการฝึกอบรมด้านการทดสอบ	67
รูปที่ 4.45 ตัวอย่างรายงานสรุปการฝึกอบรมประจำปี	67
รูปที่ 4.46 ตัวอย่างรายงานรายละเอียดของข้อบกพร่องในแต่ละขั้นตอน	68
รูปที่ 4.47 ตัวอย่างรายงานสรุปข้อบกพร่องในแต่ละขั้นตอน	68

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.48 ตัวอย่างกราฟในรายงานสรุปข้อบกพร่องในแต่ละขั้นตอน	69
รูปที่ 4.49 ตัวอย่างรายงานกิจกรรมการทดสอบโมดูล	70
รูปที่ 4.50 ตัวอย่างรายงานความพยายามในการทดสอบของโครงการตามระยะเวลา	71
รูปที่ 4.51 ตัวอย่างรายงานสถานะในการพัฒนากรณีทดสอบ	72
รูปที่ 4.52 ตัวอย่างรายงานการใช้กรณีทดสอบในแต่ละกิจกรรมการทดสอบ	72
รูปที่ 4.53 ตัวอย่างรายงานสรุปสถานะของการใช้กรณีทดสอบในแต่ละขั้นตอนการทดสอบ ...	73
รูปที่ 4.54 ตัวอย่างรายงานความก้าวหน้าของกรณีทดสอบในการทดสอบตามระยะเวลา	73
รูปที่ 4.55 ตัวอย่างกราฟความก้าวหน้าของกรณีทดสอบในการทดสอบตามระยะเวลา	74
รูปที่ 4.56 ตัวอย่างรายงานความก้าวหน้าของการทดสอบโมดูลตามระยะเวลา	74
รูปที่ 4.57 ตัวอย่างกราฟความก้าวหน้าของการทดสอบโมดูลตามระยะเวลา	75
รูปที่ 4.58 ตัวอย่างรายงานความครอบคลุมการทดสอบของการทดสอบโมดูล	75
รูปที่ 4.59 ตัวอย่างรายงานแสดงจำนวนโมดูลในช่วงเปอร์เซ็นต์ความครอบคลุมต่างๆ	76
รูปที่ 4.60 ตัวอย่างรายงานความหนาแน่นของข้อบกพร่องของโมดูล	77
รูปที่ 4.61 ตัวอย่างรายงานสถานะของโมดูลตามกิจกรรมการทดสอบเบ็ดเสร็จ	77
รูปที่ 4.62 ตัวอย่างรายงานความก้าวหน้าการทดสอบเบ็ดเสร็จตามระยะเวลา	78
รูปที่ 4.63 ตัวอย่างรายงานสถานะของการทดสอบความต้องการ	79
รูปที่ 4.64 ตัวอย่างรายงานความก้าวหน้าการทดสอบความต้องการตามระยะเวลา	80
รูปที่ 4.65 ตัวอย่างรายงานความหนาแน่นของข้อบกพร่องตามระยะเวลา	81
รูปที่ 4.66 ตัวอย่างหน้าจอบ่งชี้ผลการทบทวนในระดับ 4	82
รูปที่ 4.67 ตัวอย่างหน้าจอรายละเอียดการทบทวนในระดับ 4	83
รูปที่ 4.68 ตัวอย่างหน้าจอรายละเอียดข้อบกพร่องในระดับ 4	83
รูปที่ 4.69 ตัวอย่างหน้าจอกิจกรรมโปรแกรมการวัดการทดสอบในระดับ 4	84
รูปที่ 4.70 ตัวอย่างหน้าจอประเภทข้อบกพร่องในระดับ 4	84
รูปที่ 4.71 ตัวอย่างหน้าจอความรุนแรงข้อบกพร่องในระดับ 4	85
รูปที่ 4.72 ตัวอย่างหน้าจอคุณสมบัติด้านคุณภาพของซอฟต์แวร์ในระดับ 4	86

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.73 ตัวอย่างรายงานประสิทธิผลของการทบทวน	87
รูปที่ 4.74 ตัวอย่างรายงานรายละเอียดการทบทวน	88
รูปที่ 4.75 ตัวอย่างรายงานสรุปการทบทวน	89
รูปที่ 4.76 ตัวอย่างรายงานสถานะโปรแกรมการวัดการทดสอบ	90
รูปที่ 4.77 ตัวอย่างรายงานจำนวนข้อบกพร่องตามประเภทของข้อบกพร่อง	91
รูปที่ 4.78 ตัวอย่างรายงานสรุปจำนวนข้อบกพร่องตามระดับความรุนแรง	92
รูปที่ 4.79 ตัวอย่างรายงานรายละเอียดผลผลิตของผู้ทดสอบ	93
รูปที่ 4.80 ตัวอย่างรายงานสรุปผลผลิตของผู้ทดสอบ	93
รูปที่ 4.81 ตัวอย่างรายงานความน่าเชื่อถือของซอฟต์แวร์	94
รูปที่ 4.82 ตัวอย่างรายงานคุณสมบัติด้านคุณภาพของซอฟต์แวร์	95
รูปที่ 4.83 ตัวอย่างหน้าจอการประชุมหาสาเหตุรากของข้อบกพร่องในระดับ 5	96
รูปที่ 4.84 ตัวอย่างหน้าจอประเภทสาเหตุของข้อบกพร่องในระดับ 5	96
รูปที่ 4.85 ตัวอย่างหน้าจอรายละเอียดข้อบกพร่องในระดับ 5	97
รูปที่ 4.86 ตัวอย่างหน้าจอกิจกรรมป้องกันข้อบกพร่องในระดับ 5	97
รูปที่ 4.87 ตัวอย่างหน้าจอการควบคุมคุณภาพในระดับ 5	98
รูปที่ 4.88 ตัวอย่างหน้าจอการประเมินเทคโนโลยีหรือเครื่องมือการทดสอบ	98
รูปที่ 4.89 ตัวอย่างหน้าจอความพยายามที่ใช้ในการกำหนดเมื่อใดจะหยุดทดสอบ	99
รูปที่ 4.90 ตัวอย่างหน้าจอเกณฑ์ที่ใช้ในการกำหนดเมื่อใดจะหยุดทดสอบ	99
รูปที่ 4.91 ตัวอย่างรายงานสถานะในการหาสาเหตุรากของข้อบกพร่อง	100
รูปที่ 4.92 ตัวอย่างกราฟรายงานสรุปความพยายามในการหาสาเหตุรากของข้อบกพร่อง	100
รูปที่ 4.93 ตัวอย่างรายงานข้อบกพร่องตามประเภทของสาเหตุ	101
รูปที่ 4.94 ตัวอย่างรายงานรายละเอียดสถานะกิจกรรมการป้องกันข้อบกพร่อง	102
รูปที่ 4.95 ตัวอย่างรายงานสรุปสถานะกิจกรรมการป้องกันข้อบกพร่อง	102
รูปที่ 4.96 ตัวอย่างรายงานเปรียบเทียบความหนาแน่นข้อบกพร่องระหว่างโครงการ	103
รูปที่ 4.97 ตัวอย่างรายงานแผนผังควบคุมความครอบคลุมทางเดินพื้นฐาน	104
รูปที่ 4.98 ตัวอย่างรายงานแผนผังควบคุมความหนาแน่นข้อบกพร่องของโมดูล	105

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.99 ตัวอย่างรายงานสถานะของกิจกรรมในการปรับปรุงกระบวนการทดสอบ	106
รูปที่ 4.100 ตัวอย่างรายงานความพยายามในการทดสอบในแต่ละเดือน	107
รูปที่ 4.101 ตัวอย่างรายงานประสิทธิผลในการจัดซื้ออุปกรณ์ระหว่างโครงการ	107
รูปที่ 4.102 ตัวอย่างรายงานความพยายามที่ใช้ในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ ..	108
รูปที่ 4.103 ตัวอย่างรายงานความหนาแน่นข้อบกพร่องตั้งแต่การทดสอบระบบ	109



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



1.1 ความเป็นมาและความสำคัญของปัญหา

การทดสอบเป็นกระบวนการ (Process) ที่สำคัญในการพัฒนาซอฟต์แวร์ เพื่อที่จะค้นหาข้อผิดพลาดของซอฟต์แวร์แล้วทำการแก้ไขก่อนส่งไปยังผู้ใช้ ทำให้ซอฟต์แวร์มีคุณภาพที่ดี กระบวนการทดสอบที่ดีจะต้องค้นหาข้อผิดพลาดออกมาให้ได้มากที่สุด มิฉะนั้นข้อผิดพลาดจะยังคงอยู่ในซอฟต์แวร์ ทำให้เกิดปัญหาต่างๆ เมื่อนำไปใช้งานซึ่งจะทำให้เกิดความเสียหายและเสียค่าใช้จ่ายที่สูงในการแก้ไข ในแต่ละขั้นตอนของการพัฒนาซอฟต์แวร์ยังค้นหาข้อผิดพลาดได้ช้าเท่าใดจากจุดที่ผิดพลาดเกิดขึ้น ข้อผิดพลาดนั้นก็จะขยายต่อไปในขั้นตอนอื่น ซึ่งจะทำให้เสียเวลาและค่าใช้จ่ายที่สูงมากยิ่งขึ้นในการแก้ไข [1] ดังนั้นจะต้องมีการทดสอบในทุกๆ ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์ เพื่อที่จะค้นหาข้อผิดพลาดให้ได้เร็วที่สุด และทำการแก้ไขทันที

ทีเอ็มเอ็ม (Testing Maturity Model : TMM) [2] เป็นโมเดลที่มุ่งเน้นเฉพาะกระบวนการทดสอบ ที่จะทำให้องค์กรสามารถประเมินความสามารถและปรับปรุงกระบวนการทดสอบ ทีเอ็มเอ็มระบุถึงกิจกรรม (Activities) ของการทดสอบต่างๆ ที่ต้องทำในกระบวนการพัฒนาซอฟต์แวร์ในทุกๆ ขั้นตอน เพื่อบรรลุเป้าหมายที่ระบุไว้ในทีเอ็มเอ็ม ทีเอ็มเอ็มแบ่งออกเป็นระดับต่างๆ เมื่อกระบวนการทดสอบขององค์กรบรรลุเป้าหมายในระดับใดก็แสดงให้เห็นว่า องค์กรนั้นๆ มีความสามารถในการทดสอบอยู่ที่ระดับนั้นๆ ของทีเอ็มเอ็ม ยิ่งระดับสูงขึ้นก็就会有ความสามารถสูงขึ้น โดยต้องมีความสามารถที่ระดับล่างเป็นฐาน

การที่จะรู้ว่ากระบวนการใดดีหรือไม่ดีนั้น จะต้องทำการวัด (Measurement) กระบวนการทดสอบก็เช่นเดียวกัน การวัดจะทำให้เข้าใจกระบวนการทดสอบในปัจจุบัน ประเมินผลว่าเป็นไปตามเป้าหมายหรือไม่ เห็นจุดบกพร่องในกระบวนการซึ่งจะต้องทำการปรับปรุง และควบคุมกระบวนการให้ได้ตามมาตรฐานที่ตั้งไว้

ปัจจุบันทีเอ็มเอ็มยังไม่มีการระบุถึงมาตรวัด (Metrics) ในแต่ละระดับที่องค์กรควรเก็บ เพื่อใช้เป็นข้อมูลในการควบคุมและติดตามในการปรับปรุงกระบวนการทดสอบ ดังนั้นงานวิจัยนี้จะเสนอมาตรวัดของการทดสอบในแต่ละระดับของทีเอ็มเอ็ม

1.2 วัตถุประสงค์

- 1) เพื่อพัฒนามาตรวัดของการทดสอบแต่ละระดับของทีเอ็มเอ็ม (ยกเว้นระดับ 1)
- 2) เพื่อพัฒนาเครื่องมือในการเก็บค่าที่ได้จากการวัดตามมาตรวัดของการทดสอบ แล้วนำข้อมูลที่เก็บมานำเสนอในรูปแบบที่จะเป็นประโยชน์ต่อการวิเคราะห์กระบวนการทดสอบและกระบวนการพัฒนาซอฟต์แวร์

1.3 ขอบเขตการวิจัย

- 1) พัฒนามาตรวัดของการทดสอบตามแนวคิดของทีเอ็มเอ็ม ในแต่ละระดับตั้งแต่ระดับที่ 2 ถึง 5
- 2) มาตรวัดจะมีค่าเป็นตัวเลข ซึ่งอาจจะคำนวณได้จากสมการคณิตศาสตร์ หรือวิธีการประเมินค่าแบบการให้ค่าเรตติ้ง (Rating)
- 3) งานวิจัยจะให้แนวทางในการหาค่าของมาตรวัด แต่จะไม่บอกถึงความเหมาะสมของค่าที่วัดออกมาได้
- 4) เครื่องมือที่จะพัฒนาจะถูกออกแบบเพื่อเก็บข้อมูลในแต่ละโครงการ (Project) ไว้ในฐานข้อมูลซึ่งจะนำข้อมูลมาตรวัดของโครงการต่างๆ มาสรุปรวมกัน เพื่อดูแนวโน้มของค่ามาตรวัดที่วัดได้
- 5) เครื่องมือที่จะพัฒนาสามารถทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) ที่ใช้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 95 (Microsoft Windows 95) เป็นอย่างต่ำ

1.4 ขั้นตอนและวิธีการดำเนินการวิจัย

- 1) ศึกษาทีเอ็มเอ็ม
- 2) ศึกษาวิธีการ เทคนิค และเครื่องมือแบบต่างๆ ที่ใช้ในการทดสอบ
- 3) ศึกษาทฤษฎีและหลักการในการวัดซอฟต์แวร์ รวมถึงมาตรวัดแบบต่างๆ ที่จะปรับนำมาใช้สำหรับการพัฒนามาตรวัดของทีเอ็มเอ็ม
- 4) พัฒนามาตรวัดสำหรับแต่ละระดับของทีเอ็มเอ็ม ตามคุณสมบัติและเป้าหมายในแต่ละระดับของทีเอ็มเอ็ม
- 5) ออกแบบและพัฒนาเครื่องมือที่จะใช้เก็บและวิเคราะห์มาตรวัด
- 6) ทดสอบเครื่องมือที่พัฒนาขึ้น
- 7) สรุปผลการวิจัยและข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) องค์กรพัฒนาซอฟต์แวร์สามารถนำมาตรวัดที่ได้ไปใช้เป็นตัวชี้ให้เห็นถึงคุณภาพกระบวนการทดสอบ อันจะเป็นประโยชน์ในการปรับปรุงกระบวนการทดสอบและการพัฒนาซอฟต์แวร์ ซึ่งจะทำได้ซอฟต์แวร์ที่มีคุณภาพดี
- 2) มาตรวัดของการทดสอบที่ได้สามารถนำไปใช้ร่วมกับมาตรวัดซอฟต์แวร์ด้านอื่น ซึ่งจะทำให้องค์กรมีมาตรวัดได้สมบูรณ์มากยิ่งขึ้น สามารถนำโปรแกรมไปใช้เป็นเครื่องมือในการเก็บและช่วยในการวิเคราะห์ข้อมูลมาตรวัดของการทดสอบ

1.6 โครงร่างของวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์แบ่งเป็นบทดังนี้

- บทที่ 1 เป็นบทนำ กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตการวิจัย ขั้นตอนและวิธีการดำเนินการวิจัย และประโยชน์ที่คาดว่าจะได้รับจากการวิจัย
- บทที่ 2 เป็นบทเกี่ยวกับงานวิจัย แนวคิดและทฤษฎีที่เกี่ยวข้อง กล่าวถึงแนวคิดเกี่ยวกับการทดสอบ ทีเอ็มเอ็ม การวัดซอฟต์แวร์ และวิธีการที่ใช้ในการได้มาตรวัดสำหรับทีเอ็มเอ็ม
- บทที่ 3 เป็นบทเกี่ยวกับมาตรวัดในแต่ละระดับของทีเอ็มเอ็ม กล่าวถึงรายละเอียดการได้มาของมาตรวัดต่างๆ
- บทที่ 4 เป็นบทเกี่ยวกับการออกแบบและพัฒนาเครื่องมือ กล่าวถึงรายละเอียดในการออกแบบเครื่องมือ ซึ่งประกอบด้วยการออกแบบฐานข้อมูล การออกแบบอินพุต และการออกแบบเอาต์พุต รวมทั้งมีตัวอย่างหน้าจอและรายงานของเครื่องมือ และกล่าวถึงการทดสอบเครื่องมือ และผลการทดสอบ
- บทที่ 5 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ
- ภาคผนวก เป็นบทความวิชาการซึ่งเป็นส่วนหนึ่งของวิทยานิพนธ์ที่ได้ตีพิมพ์ นอกจากนี้ยังให้รายละเอียดตารางและเขตข้อมูลต่างๆ ในฐานข้อมูลของเครื่องมือ

1.7 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์ได้ตีพิมพ์เป็นบทความวิชาการในหัวข้อ “A Set of Measurements to Improve Software Testing Process” โดย ดร.ธาราทิพย์ สุวรรณศาสตร์ และประกาศ ศรีชัยวัฒน์ ในการประชุมวิชาการ “The 1999 National Computer Science and Engineering Conference (NCSEC’ 99)” ซึ่งจัดโดย คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัย อัสสัมชัญ ณ โรงแรมแลนด์มาร์ค ถนนสุขุมวิท ในวันที่ 16-17 ธันวาคม พ.ศ. 2542 รายละเอียดบทความ ดูได้จากภาคผนวก ก

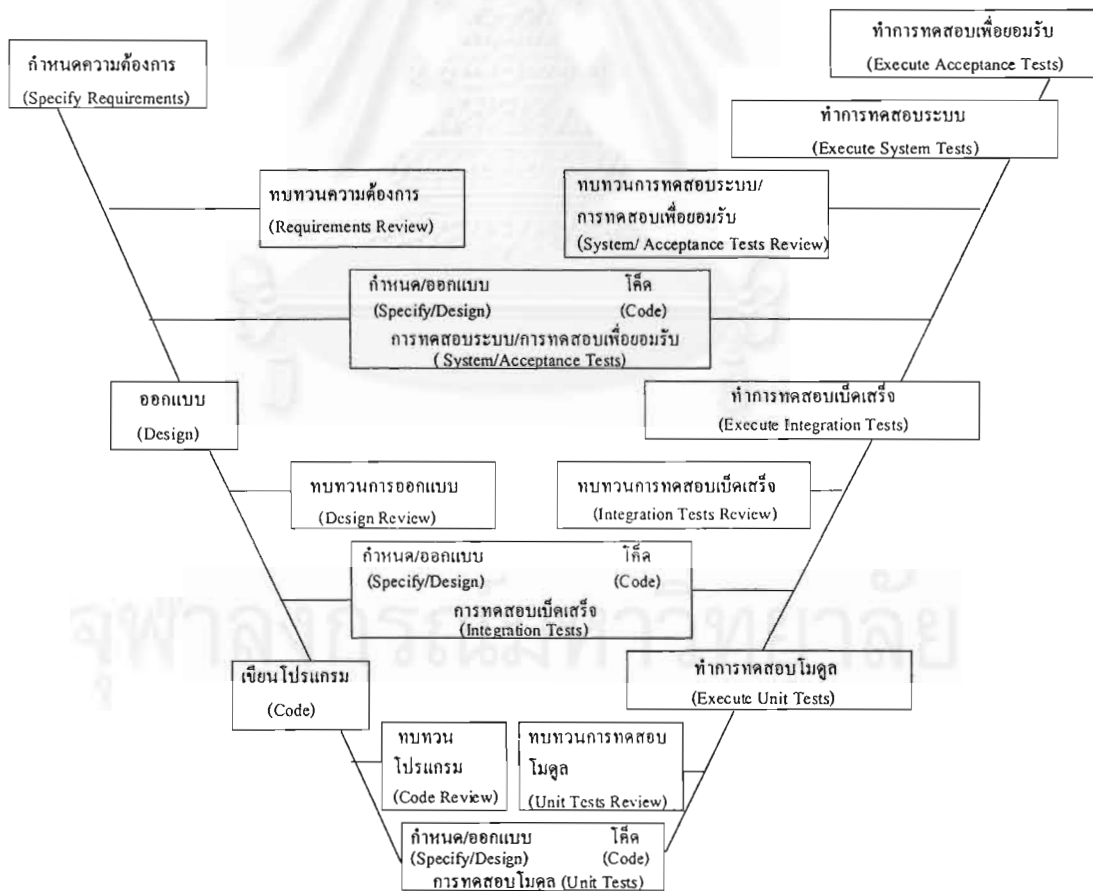
บทที่ 2

งานวิจัย แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 แนวคิดเกี่ยวกับการทดสอบ

ในกระบวนการพัฒนาซอฟต์แวร์ การทดสอบจะเป็นขั้นตอนๆ หนึ่งหลังจากขั้นตอนการเขียนโปรแกรม (Coding) ซึ่งประกอบไปด้วยการทดสอบโมดูล (Unit testing) การทดสอบเบ็ดเสร็จ (Integration testing) การทดสอบระบบ (System testing) และการทดสอบเพื่อยอมรับ (Acceptance testing) แนวคิดนี้การทดสอบจะเริ่มก็ต่อเมื่อทำการเขียนโปรแกรมเสร็จ เป็นการทดสอบโดยการปฏิบัติตามคำสั่งในโปรแกรม (Execution-based testing) เท่านั้น ดังนั้นข้อผิดพลาดที่เกิดขึ้นจากขั้นตอนการกำหนดความต้องการ และการออกแบบจะไม่ถูกพบและแก้ไขอย่างทันที่ทันใด จึงมีแนวคิดในการขยายการทดสอบไปยังทุกๆ ขั้นตอนของการพัฒนาซอฟต์แวร์

ชราทิพย์ [2] ได้เสนอโมเดลเอ็กซ์เทนด / โมดิฟายด์ วี (Extended / Modified V-Model) ซึ่งปรับปรุงมาจากโมเดลโมดิฟายด์วี (Modified V- Model)[3] โมเดลนี้ได้รวมการทดสอบเข้ากับกระบวนการพัฒนาซอฟต์แวร์ตลอดกระบวนการ ดังรูปที่ 2.1



รูปที่ 2.1 โมเดลเอ็กซ์เทนด / โมดิฟายด์ วี [2]

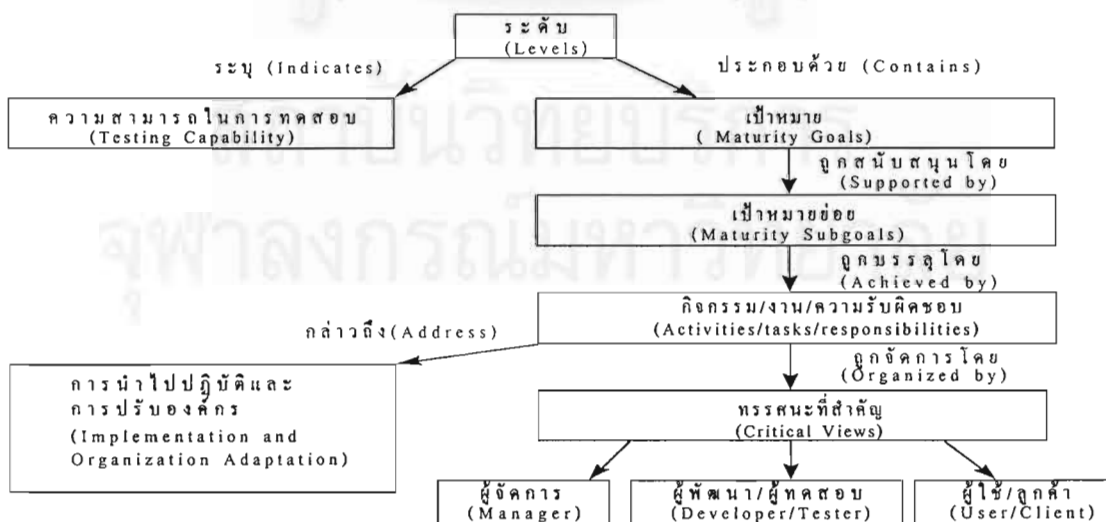
จากรูปที่ 2.1 การทดสอบไม่ได้หมายถึงแค่การทดสอบโดยการปฏิบัติตามคำสั่งในโปรแกรมเท่านั้น ซึ่งจะทำให้การทดสอบหลังจากเขียนโปรแกรมเสร็จ แต่ยังคงขยายการทดสอบเข้าไปยังกระบวนการพัฒนาซอฟต์แวร์ในทุกๆ ขั้นตอน เช่น หลังจากที่มีระบุความต้องการ (Requirements) แล้ว ก็จะต้องดำเนินการกิจกรรมการทดสอบโดยการทบทวน (Reviews) เพื่อหาข้อบกพร่องหรือข้อผิดพลาดของความต้องการ ซึ่งจะช่วยให้ข้อผิดพลาดไม่ขยายต่อไปในขั้นตอนถัดไป อันจะทำให้เกิดปัญหาดังกล่าวตามมา

2.2 ทีเอ็มเอ็ม

ทีเอ็มเอ็ม ถูกเสนอโดยธาราทิพย์ [2] เป็นโมเดลที่มุ่งเน้นเฉพาะกระบวนการทดสอบ จุดประสงค์หลักที่ทีเอ็มเอ็มถูกพัฒนาขึ้นมานั้นเพราะว่าในปัจจุบันไม่มีโมเดลใดเลยที่มุ่งเน้นถึงความสามารถในการทดสอบ และการปรับปรุงกระบวนการทดสอบในองค์กร ซึ่งองค์กรพัฒนาซอฟต์แวร์สามารถนำทีเอ็มเอ็มมาใช้ร่วมกับซีเอ็มเอ็ม (Capability Maturity Model: CMM) [4] เพื่อเป็นแนวทางในการพัฒนาซอฟต์แวร์ให้ได้คุณภาพ

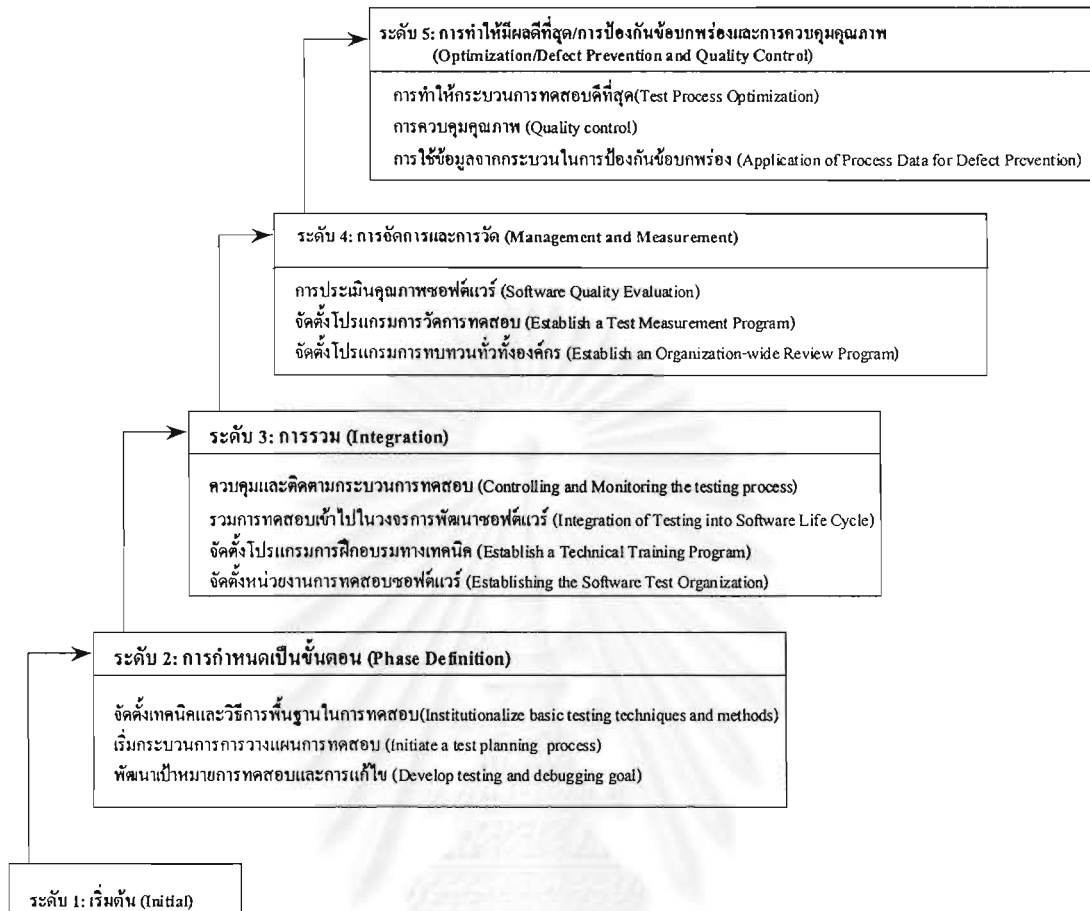
องค์ประกอบของทีเอ็มเอ็มมีอยู่ 2 ส่วน คือ ระดับ (Level) และ โมเดลการประเมินผล (Assessment model)

2.2.1 ระดับ ทีเอ็มเอ็มได้ถูกแบ่งออกเป็นระดับ ซึ่งจะระบุความสามารถในการทดสอบ (Testing capability) ที่ระดับนั้น โดยที่โครงสร้างของทีเอ็มเอ็มเป็นไปตามรูปที่ 2.2 ในแต่ละระดับจะประกอบไปด้วยเป้าหมาย (Maturity goals) แต่ละเป้าหมายประกอบไปด้วยเป้าหมายย่อย (Maturity subgoals) ซึ่งจะบรรลุได้ด้วยการดำเนินงานและความรับผิดชอบ (Activities/Tasks/Responsibilities) โดยกลุ่มบุคคลที่เกี่ยวข้องกับกระบวนการทดสอบ 3 กลุ่มคือ ผู้จัดการ (Manager) ผู้พัฒนา / ผู้ทดสอบ (Developer / Tester) และผู้ใช้ / ลูกค้า (User / Client)



รูปที่ 2.2 โครงสร้างทีเอ็มเอ็มแต่ละระดับ [2]

รูปที่ 2.3 แสดงให้เห็นถึงระดับ 5 ระดับของ ทีเอ็มเอ็ม ในแต่ละระดับแสดงให้เห็นถึงเป้าหมายที่ต้องบรรลุ การจะได้ระดับที่สูงขึ้นจะต้องได้ระดับล่างเป็นฐาน



รูปที่ 2.3 เป้าหมายทีเอ็มเอ็มในแต่ละระดับ [2]

รายละเอียดของแต่ละระดับเป็นดังต่อไปนี้

ระดับ 1 : เริ่มต้น จุดมุ่งหมายของการทดสอบ เพื่อแสดงว่าซอฟต์แวร์ทำงานโดยไม่ได้คำนึงถึงความต้องการของผู้ใช้ / ลูกค้า ระดับนี้การทดสอบเป็นไปอย่างไม่มีระเบียบกฎเกณฑ์ (Chaotic) และเป็นไปตามความพอใจ (Ad hoc) โดยจะทำการทดสอบหลังจากเขียนโปรแกรมเสร็จ ไม่มีการจัดการการทดสอบ ไม่มีวัตถุประสงค์ของการทดสอบ (Test objectives) ไม่มีการวางแผนการทดสอบ (Test planning) ไม่มีการวิเคราะห์และออกแบบการทดสอบ (Test analysis and design) และไม่มีการรับประกันคุณภาพ (Quality assurance) ในระดับนี้องค์กรไม่มีทรัพยากร เครื่องมือ และบุคลากรที่เกี่ยวข้องกับการทดสอบอย่างเพียงพอ

ระดับ 2 : การกำหนดเป็นขั้นตอน ที่ระดับนี้กระบวนการทดสอบเริ่มมีระบบมากขึ้น โดยถูกกำหนดขึ้นให้เป็นขั้นตอนๆ หนึ่งหลังจากขั้นตอนการเขียนโปรแกรมในวงจรของการพัฒนาซอฟต์แวร์ การทดสอบยังเป็นแนวคิดแบบการทดสอบโดยการปฏิบัติตามคำสั่งในโปรแกรม เป้าหมายหลักของการทดสอบคือการทดสอบว่าซอฟต์แวร์เป็นไปตามข้อกำหนดที่

ต้องการหรือไม่ ในระดับนี้มีการวางแผนการทดสอบ แต่ทำหลังจากขั้นตอนการเขียนโปรแกรม มีเทคนิคและวิธีการการทดสอบเบื้องต้น มีการกำหนดวัตถุประสงค์ของการทดสอบแต่ทำหลังจากขั้นตอนการเขียนโปรแกรม

ระดับ 3 : การรวม การทดสอบไม่เป็นแค่ขั้นตอนหลังจากขั้นตอนการเขียนโปรแกรม แต่จะเริ่มเข้าไปรวมกับวงจรของการพัฒนาซอฟต์แวร์ เป้าหมายระดับนี้แสดงให้เห็นว่าซอฟต์แวร์ไม่มีความผิดพลาด (Error-free) และเป็นไปตามความต้องการของผู้ใช้ / ลูกค้า การวางแผนการทดสอบจะเริ่มตั้งแต่ขั้นตอนระบุความต้องการแทนที่จะระบุหลังจากขั้นตอนการเขียนโปรแกรม ในองค์กรมีการจัดตั้งหน่วยงานที่จะรับผิดชอบด้านการทดสอบโดยเฉพาะ เช่นกำหนดวัตถุประสงค์การทดสอบให้เป็นไปตามความต้องการของผู้ใช้ / ลูกค้า กำหนดมาตรฐานการทดสอบ มีโปรแกรมการฝึกอบรมด้านเทคนิคการทดสอบ มีการควบคุมและติดตามกระบวนการทดสอบ แต่ยังไม่มีการทบทวนตลอดวงจรของการพัฒนาซอฟต์แวร์ ดังนั้นระดับนี้การเข้าไปรวมของกระบวนการทดสอบในวงจรของการพัฒนาซอฟต์แวร์ไม่สมบูรณ์

ระดับ 4 : การจัดการและการวัด ในระดับนี้กระบวนการทดสอบรวมกับวงจรของการพัฒนาซอฟต์แวร์อย่างสมบูรณ์ ดังรูปที่ 2.1 ทุกๆ ขั้นตอนของการพัฒนาซอฟต์แวร์มีการทดสอบ มีการทดสอบผลิตภัณฑ์ซอฟต์แวร์ในคุณสมบัติเชิงคุณภาพ (Quality attributes) เช่นประสิทธิภาพ (Efficiency) ความสามารถในการบำรุงรักษา (Maintainability) มีการทบทวนซึ่งถือเป็นนโยบายขององค์กร มีการเก็บกรณีทดสอบ (Test cases) และข้อบกพร่อง (Defects) จากโครงการต่างๆ ขององค์กรไว้ในฐานข้อมูล มีนโยบายและโปรแกรมในการวัดกระบวนการทดสอบ (Test measurement program) แต่ในระดับนี้ยังไม่คำนึงถึงการป้องกันข้อบกพร่อง (Defect prevention)

ระดับ 5: การทำให้มีผลดีที่สุด, การป้องกันข้อบกพร่อง และ การควบคุมคุณภาพ (Optimization, Defect Prevention and Quality control) จุดมุ่งหมายหลักจะเป็นการป้องกันข้อบกพร่องในทุกขั้นตอนของวงจรพัฒนาซอฟต์แวร์ มีการวิเคราะห์ข้อบกพร่องที่เกิดขึ้น แล้วหาทางป้องกันไม่ให้ข้อบกพร่องนั้นเกิดขึ้นอีกโดยใช้วิธีการทางสถิติในการจัดการและควบคุมกระบวนการทดสอบ มีการใช้เครื่องมือการทดสอบแบบอัตโนมัติในการใช้และใช้ซ้ำกรณีการทดสอบได้อย่างอัตโนมัติ มีกลไกในการปรับแต่ง (Fine-tune) และปรับปรุงการทดสอบอย่างต่อเนื่อง มุ่งที่จะผลิตซอฟต์แวร์ที่มีข้อบกพร่องเป็นศูนย์ (Zero defect software)

2.2.2 โมเดลการประเมินผล [5] องค์กรสามารถใช้โมเดลการประเมินผลในการประเมินและปรับปรุงกระบวนการทดสอบ โมเดลการประเมินผลมีองค์ประกอบ 3 ส่วนคือ

1) ขั้นตอนการประเมินผล (Assessment procedure) เป็นลำดับขั้นตอนที่องค์กรต้องทำในการประเมินกระบวนการทดสอบ ซึ่งประกอบไปด้วย การเตรียมการ การทำการประเมิน การรายงานผลการประเมิน การวิเคราะห์ผลการประเมิน การวางแผนการปรับปรุงและทำการปรับปรุง

2) แบบสอบถาม (Questionnaire) ใช้ในการเก็บและบันทึกข้อมูลจากการประเมินผล คำถามในแบบสอบถามจะสัมพันธ์กับเป้าหมาย เป้าหมายย่อยและกิจกรรม/งาน/ความรับผิดชอบในแต่ละระดับของทีเอ็มเอ็ม

3) การฝึกอบรมและเกณฑ์ในการเลือกทีมงานที่ทำการประเมิน ทีมงานที่ทำการประเมินได้รับการฝึกอบรม ในการเตรียมการก่อนทำการประเมินกระบวนการทดสอบ เกณฑ์ในการเลือกสมาชิกในทีมงานคือ ความเข้าใจในเป้าหมายของการประเมิน ความรู้ ประสบการณ์ ทักษะและคำสัญญา (Commitment) ของสมาชิกในการปรับปรุงกระบวนการทดสอบ

เมื่อองค์กรได้ทำการประเมินตามขั้นตอนการประเมินผลแล้ว องค์กรได้ผลจากการประเมินดังต่อไปนี้

- ระดับของทีเอ็มเอ็ม (ระดับ 1 ถึง 5)
- สภาพสถานะกระบวนการทดสอบขององค์กร
- จุดแข็งและจุดอ่อนของกระบวนการทดสอบ
- แผนงานการปรับปรุงกระบวนการทดสอบ
- บันทึกการประเมิน

2.3 แนวคิดเกี่ยวกับการวัดซอฟต์แวร์

2.3.1 มาตรวัดซอฟต์แวร์ (Software Metrics)

ในงานวิศวกรรมต่างๆ การวัดถือเป็นเรื่องปกติ เช่นการวัดอุณหภูมิ การวัดน้ำหนัก การวัดความกว้างและความยาว เป็นต้น เราจะต้องระบุสิ่งที่ต้องการวัด (Entities) และคุณสมบัติต่างๆ (Attributes) ของสิ่งที่ต้องการวัด พร้อมกับหน่วยที่จะใช้ในการวัด (Unit) เช่นการวัดอุณหภูมิจะมีหน่วยเป็นองศาเซลเซียส การวัดน้ำหนักจะมีหน่วยเป็นกิโลกรัม การวัดความกว้างและความยาวจะมีหน่วยเป็นเมตร เป็นต้น การวัดทางวิศวกรรมซอฟต์แวร์ก็เช่นเดียวกัน โดยมีวัตถุประสงค์ในการวัดซอฟต์แวร์ดังต่อไปนี้ [1]

- เพื่อชี้ถึงคุณภาพของซอฟต์แวร์
- เพื่อประเมินความสามารถในการผลิต (Productivity) ของพนักงานที่ผลิตซอฟต์แวร์
- เพื่อประเมินผลประโยชน์ที่ได้รับจากการใช้วิธีการทางวิศวกรรมซอฟต์แวร์และเครื่องมือ
- เพื่อใช้เป็นพื้นฐาน (Baseline) ในการประมาณค่า (Estimation) ของโครงการต่างๆ เช่น ความพยายาม (Effort) และค่าใช้จ่าย (Cost)

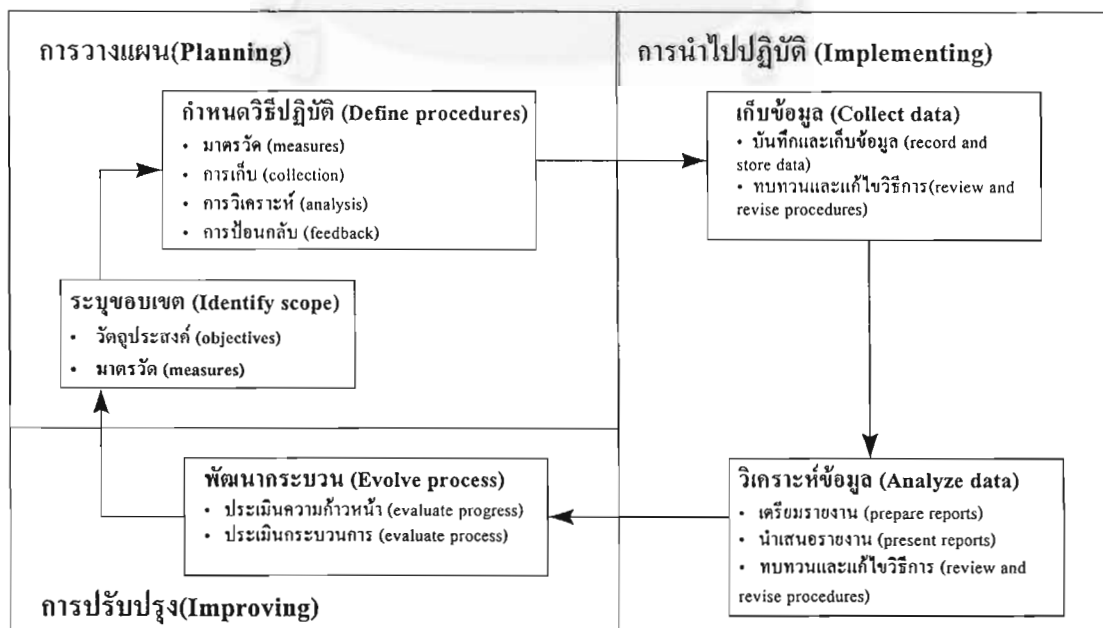
- เพื่อช่วยให้เห็นถึงความต้องการในการปรับปรุงกระบวนการพัฒนาซอฟต์แวร์ ความต้องการเครื่องมือในการพัฒนาและเพิ่มการอบรมแก่บุคลากร

วิธีการวัดซอฟต์แวร์แบ่งออกเป็น 2 ลักษณะคือการวัดโดยตรง (Direct measures) และการวัดโดยทางอ้อม (Indirect measures) คุณสมบัติบางอย่างของซอฟต์แวร์หรือ กระบวนการการพัฒนาซอฟต์แวร์ ก็สามารถวัดได้โดยตรง เช่น ค่าใช้จ่าย ความพยายาม จำนวนบรรทัดของคำสั่ง (Lines of Code) จำนวนข้อบกพร่อง เป็นต้น แต่ก็มีหลายอย่างไม่สามารถวัดได้โดยตรงซึ่งต้องวัดโดยอ้อม เช่น ความซับซ้อน (Complexity) ความน่าเชื่อถือ (Reliability) ความสามารถในการบำรุงรักษา เป็นต้น

มาตรวัดซอฟต์แวร์ควรมีคุณสมบัติดังต่อไปนี้ [6]

- มาตรวัดควรจะคงทน (Robust) คือให้ผลการวัดเหมือนเดิมเมื่อวัดซ้ำ (Repeatable) และเที่ยงตรง (Precise)
- มาตรวัดควรจะสัมพันธ์กับคุณสมบัติของผลิตภัณฑ์และกระบวนการที่ทำการวัด
- มาตรวัดควรจะแนะนำให้เกิดการปรับปรุง เช่น การวัดความซับซ้อนของซอฟต์แวร์จะช่วยแนะนำให้ลดความซับซ้อนลง
- มาตรวัดควรจะง่าย (Simple) มาตรวัดที่ซับซ้อนจะยากในการวิเคราะห์หรืออธิบาย และการแปลความหมาย

2.3.2 กระบวนการการวัดซอฟต์แวร์ (Software measurement process) มีวงจรดังรูปที่ 2.4 [7] ซึ่งประกอบไปด้วยขั้นตอนดังต่อไปนี้



รูปที่ 2.4 สถาปัตยกรรมของกระบวนการการวัดซอฟต์แวร์ [7]

1) การวางแผน ประกอบด้วย 2 ขั้นตอนย่อยคือ

1.1) ระบุขอบเขตของการวัด เริ่มต้นจากองค์การกำหนดวัตถุประสงค์ที่จำเป็น ต้องทำการวัด ตั้งเป้าหมายของการวัดที่สอดคล้องกับวัตถุประสงค์นั้น จากนั้นตั้งคำถามสำหรับแต่ละเป้าหมายซึ่งคำตอบจะบอกได้ว่าบรรลุเป้าหมายไปมากน้อยแค่ไหน โดยที่คำตอบจะได้มาจากสิ่งที่ต้องการวัดหรือมาตรวัดที่ต้องกำหนดลงไป

1.2) กำหนดวิธีการปฏิบัติ ขั้นตอนนี้จะให้รายละเอียดมาตรวัดที่ได้จากขั้นตอนแรกอย่างชัดเจน (Clear) ไม่คลุมเครือ (Unambiguous) และนำไปปฏิบัติงานได้ (Operational) จากนั้นจะระบุถึงวิธีการเก็บข้อมูลเช่น ใครเป็นคนเก็บข้อมูล เริ่มต้นเก็บเมื่อใด ความถี่ในการเก็บ เป็นต้น แล้วให้แนวทางในการวิเคราะห์ข้อมูล วิธีการทำรายงาน เช่น ใช้กราฟเส้น กราฟแท่ง เป็นต้น และกลไกในการป้อนกลับเพื่อจะแก้ไขวิธีการปฏิบัติคือเมื่อกำหนดวิธีการปฏิบัติไปแล้ว จะต้องมีการตรวจสอบว่าวิธีการดังกล่าวเหมาะสมหรือไม่ จำเป็นต้องเปลี่ยนแปลงแก้ไขอย่างไร เช่น จัดประชุมเพื่อคุยว่าวิธีการเก็บข้อมูลยากในการปฏิบัติหรือไม่ ถ้ายากจะปรับปรุงวิธีการปฏิบัติในการเก็บข้อมูลให้ง่ายขึ้นได้อย่างไร

2) การนำไปปฏิบัติ ประกอบด้วย 2 ขั้นตอนย่อยคือ

2.1) เก็บข้อมูล ทำการบันทึกและเก็บข้อมูลตามวิธีการที่ได้กำหนดไว้ และต้องทบทวนดูว่าวิธีการเก็บข้อมูลเหมาะสมหรือไม่ จำเป็นต้องแก้ไขอย่างไร

2.2) วิเคราะห์ข้อมูล นำข้อมูลมาทำเป็นรายงานในรูปแบบที่กำหนดไว้แล้ว นำเสนอรายงานแก่ผู้เกี่ยวข้อง วิเคราะห์ข้อมูลให้สัมพันธ์กับเป้าหมายที่ตั้งไว้ ข้อมูลชี้ให้เห็นถึงสิ่งที่ทำการวัดว่าเป็นอย่างไร หาสาเหตุว่าทำไมถึงเป็นเช่นนั้น ทำการแก้ไข และทำการตัดสินใจว่าต้องปรับปรุงกระบวนการหรือไม่ ถ้าต้อง ทำที่ไหน อย่างไร นอกจากนี้ต้องทบทวนวิธีการวิเคราะห์ข้อมูล ว่าต้องมีการแก้ไขหรือไม่ เช่น กราฟดูยากทำให้ยากในการวิเคราะห์ ต้องเปลี่ยนรูปแบบรายงานให้ดูง่ายขึ้น เป็นต้น

3) การปรับปรุงกระบวนการวัด มาถึงขั้นตอนนี้ ผู้เกี่ยวข้องได้รับรายงานการวัดซึ่งจะทบทวนรายงานว่าเพียงพอต่อความต้องการหรือไม่ เช่นขาดข้อมูลบางอย่างในการวิเคราะห์หรือไม่ ผู้เกี่ยวข้องจะระบุประเด็นปัญหาต่างๆในการวัด ซึ่งต้องปรับปรุงแก้ไขในรายงานรอบถัดไป ปัญหาต่างๆที่ได้รับจากขั้นตอนการวัดทุกขั้นตอนจะถูกแก้ไขและใช้ในการปรับปรุงกระบวนการวัดรอบถัดไป เช่น ปรับแผนงานการวัด ปรับวิธีปฏิบัติ เป็นต้น

2.4 วิธีการที่ใช้ในการได้มาตรวัดสำหรับทีเอ็มเอ็ม

ในงานวิจัยนี้ ได้ใช้วิธีการ "เป้าหมาย/คำถาม/มาตรวัด" หรือจีคิวเอ็ม (Goal/Question/Metric : GQM) [8] เพื่อให้ได้รับมาตรวัดสำหรับกระบวนการทดสอบ วิธีจีคิวเอ็ม ได้รับการพัฒนาขึ้นมาเพื่อจะขจัดปัญหาที่ว่าโครงการวัดซอฟต์แวร์ส่วนใหญ่จะล้มเหลวเพราะไม่

ได้กำหนดมาตรวัดที่ชัดเจน อีกทั้งไม่มีจุดประสงค์ในการวัด เริ่มต้นวิธีจิกวเอ็มถูกใช้ในโครงการต่างๆ ขององค์การนาซา (NASA/GSFC) ซึ่งเป็นไปได้ด้วยดี อีกทั้งวิธีการนี้เป็นวิธีการที่ไม่ซับซ้อน ดังนั้นจึงถูกใช้อย่างแพร่หลายในองค์กรต่างๆ ขั้นตอนในการใช้วิธีจิกวเอ็มเป็นดังต่อไปนี้

- 2.4.1 ตั้งเป้าหมายตามความต้องการซึ่งจะบอกถึงสิ่งที่ต้องการรู้เพื่อบรรลุเป้าหมาย เช่น ต้องการปรับปรุงกระบวนการจัดการคำร้องขอแก้ไขของผู้ใช้
- 2.4.2 ตั้งคำถามสำหรับเป้าหมายที่ตั้งไว้ซึ่งเป้าหมายหนึ่งอาจจะมีหลายคำถามได้ เช่น อะไรเป็นขนาดของคำร้องขอแก้ไขที่ยังไม่ได้ดำเนินการ? เวลาที่ใช้ในการแก้ไขอยู่ภายในเวลาที่ผู้ใช้ต้องการหรือไม่?
- 2.4.3 กำหนดมาตรวัดที่จะตอบคำถามที่ตั้งไว้ เช่น เวลาเฉลี่ยที่ใช้ในการแก้ไขในขั้นตอนต่างๆ

ในงานวิจัยนี้ เป้าหมายจะเป็นเป้าหมายในแต่ละระดับของทีเอ็มเอ็ม คำถามจะถูกตั้งขึ้นเพื่อบรรลุเป้าหมายนั้น และกำหนดมาตรวัดที่จะตอบคำถามที่ตั้งไว้



บทที่ 3

มาตรวัดในแต่ละระดับของทีเอ็มเอ็ม

ในงานวิจัยนี้จะแบ่งมาตรวัดเป็น 2 ประเภทคือมาตรวัดขั้นต่ำที่จำเป็นสำหรับแต่ละระดับ (Minimum metric set) และมาตรวัดที่จะมีหรือไม่มีก็ได้ (Optional metric set) ซึ่งมาตรวัดขั้นต่ำจะมีเครื่องหมาย * ระบุอยู่ที่มาตรวัด ทั้งนี้เนื่องจากมาตรวัดที่งานวิจัยเสนอมีอยู่จำนวนมาก ถ้าองค์กรเริ่มต้นเก็บข้อมูล อาจจะเป็นภาระที่หนักในการเก็บและวิเคราะห์ข้อมูล ดังนั้นงานวิจัยจึงเสนอให้มีมาตรวัดขั้นต่ำที่จำเป็นสำหรับแต่ละระดับ ซึ่งจะเพียงพอในการติดตามและวิเคราะห์กระบวนการทดสอบได้ในระดับหนึ่ง

3.1 มาตรวัดในระดับ 2 การกำหนดเป็นขั้นตอน

ระดับ 2 ของทีเอ็มเอ็มประกอบไปด้วยเป้าหมายทั้งหมด 3 เป้าหมาย ซึ่งในแต่ละเป้าหมายมีรายละเอียดต่างๆ แสดงถึงกิจกรรมที่จะต้องกระทำเพื่อให้บรรลุเป้าหมาย โดยเป้าหมายในระดับ 2 มีดังต่อไปนี้

3.1.1 พัฒนาเป้าหมายการทดสอบและการแก้ไข

จุดมุ่งหมายสำหรับเป้าหมายนี้คือจำแนกความแตกต่างระหว่างการทดสอบและการแก้ไขข้อบกพร่อง เป้าหมายสำหรับการทดสอบคือการแสดงว่ามีข้อผิดพลาดอยู่ แต่เป้าหมายสำหรับการแก้ไขข้อบกพร่องคือการหาสาเหตุที่ทำให้เกิดข้อผิดพลาดนั้นแล้วดำเนินการแก้ไข ดังนั้นการแก้ไขข้อบกพร่องจะเป็นกิจกรรมที่ตามหลังการทดสอบ

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) จัดตั้งคณะกรรมการเกี่ยวกับการทดสอบและการแก้ไขข้อบกพร่อง
- 2) จัดตั้งเป้าหมายและนโยบายสำหรับการทดสอบทั่วทั้งองค์กรและบันทึก
- 3) จัดตั้งเป้าหมายและนโยบายสำหรับการแก้ไขข้อบกพร่องทั่วทั้งองค์กรและบันทึก
- 4) กระจายเอกสารเกี่ยวกับการทดสอบและแก้ไขข้อบกพร่องให้แก่ผู้จัดการ โครงการและผู้พัฒนา/ผู้ทดสอบ

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า เป็นดังต่อไปนี้

- ผู้จัดการจัดสรรทรัพยากร ไม่ว่าจะเป็นบุคลากรหรือเงินทุนและให้นโยบายในการตั้งเป้าหมายการทดสอบและการแก้ไขข้อบกพร่อง
- ผู้พัฒนา/ผู้ทดสอบ ทำงานกับผู้จัดการในการตั้งเป้าหมายการทดสอบในแต่ละระดับได้แก่ การทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ รวมทั้งการตั้งเป้าหมายการแก้ไขข้อบกพร่อง

- ผู้ใช้/ลูกค้า ทำงานกับผู้พัฒนา/ผู้ทดสอบ ในการกำหนดเป้าหมายการทดสอบและการแก้ไขข้อบกพร่องในขั้นตอนการกำหนดความต้องการ และการทดสอบเพื่อยอมรับ

เพื่อที่จะจำแนกความแตกต่างระหว่างกระบวนการทดสอบกับการแก้ไขข้อบกพร่อง องค์กรควรจะแยกค่าใช้จ่ายระหว่างการทดสอบกับการแก้ไขข้อบกพร่อง คำถามและมาตรวัดคือ

คำถาม 3.1.1.1 อะไรคือค่าใช้จ่ายของการทดสอบสำหรับแต่ละโครงการและระดับองค์กร ?

$$\text{มาตรวัด 3.1.1.1 } E_{\text{test-total}} = E_{\text{procedures}} + E_{\text{test-cases}} + E_{\text{test-debugging}} + E_{\text{running}}$$

โดยที่

$E_{\text{test-total}}$ คือ ค่าใช้จ่ายในการทดสอบในรูปของความพยายาม มีหน่วยเป็น คน.ชั่วโมง

$E_{\text{procedures}}$ คือ ความพยายามในการพัฒนาขั้นตอนปฏิบัติการทดสอบ (Test procedures) มีหน่วยเป็น คน.ชั่วโมง

$E_{\text{test-cases}}$ คือ ความพยายามในการพัฒนากรณีทดสอบ มีหน่วยเป็น คน.ชั่วโมง

$E_{\text{test-debugging}}$ คือ ความพยายามในการแก้ไขการทดสอบ มีหน่วยเป็น คน.ชั่วโมง

E_{running} คือ ความพยายามในการดำเนินการทดสอบ มีหน่วยเป็น คน.ชั่วโมง

คำถาม 3.1.1.2 อะไรคือค่าใช้จ่ายของการแก้ไขข้อบกพร่อง สำหรับแต่ละโครงการและระดับองค์กร ?

$$\text{มาตรวัด 3.1.1.2 } E_{\text{debug-total}} = E_{\text{investigating}} + E_{\text{correcting}} + E_{\text{retesting}}$$

โดยที่

$E_{\text{debug-total}}$ คือ ค่าใช้จ่ายในการแก้ไขข้อบกพร่องในรูปของความพยายามมีหน่วยเป็น คน.ชั่วโมง

$E_{\text{investigating}}$ คือ ความพยายามในการค้นหาสาเหตุของปัญหา มีหน่วยเป็น คน.ชั่วโมง

$E_{\text{correcting}}$ คือ ความพยายามในการแก้ไขปัญหา มีหน่วยเป็น คน.ชั่วโมง

$E_{\text{retesting}}$ คือ ความพยายามในการทดสอบซอฟต์แวร์ส่วนที่ถูกแก้ไข มีหน่วยเป็น คน.ชั่วโมง

3.1.2 เป้าหมายเริ่มกระบวนการวางแผนการทดสอบ

จุดประสงค์สำหรับเป้าหมายนี้คือ มีแผนงานการทดสอบที่จะทำในกระบวนการทดสอบซึ่งประกอบไปด้วย แผนงานการทดสอบสำหรับการทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ

เป้าหมายย่อยสำหรับเป้าหมายการเริ่มกระบวนการวางแผนการทดสอบ มีดังต่อไปนี้

- 1) จัดตั้งคณะกรรมการการวางแผนการทดสอบ

- 2) พัฒนาแผ่นแบบ (Template) สำหรับแผนงานการทดสอบ แล้วกระจายให้แก่ผู้จัดการโครงการและวิศวกรซอฟต์แวร์
- 3) พัฒนากลไกที่จะทำให้ความต้องการของผู้ใช้เป็นสิ่งที่นำเข้าสู่สำหรับแผนงานการทดสอบ
- 4) ใช้เป้าหมายการทดสอบเป็นฐานในการวางแผนการทดสอบ
- 5) ฝึกอบรมผู้จัดการโครงการในการใช้แผ่นแบบแผนงานการทดสอบขององค์กร
- 6) ฝึกอบรมวิศวกรซอฟต์แวร์ที่จะพัฒนาการออกแบบการทดสอบและกรณีทดสอบ

กิจกรรมที่จะต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า มีดังต่อไปนี้

- ผู้จัดการจัดสรรทรัพยากร มอบหมายความรับผิดชอบให้บุคคลที่จะพัฒนาแผนงานการทดสอบซึ่งประกอบไปด้วย แผนงานการทดสอบสำหรับการทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบและการทดสอบเพื่อยอมรับ นอกจากนี้ผู้จัดการจะต้องประเมินความพยายาม ค่าใช้จ่ายและกำหนดการในการทดสอบ
- ผู้พัฒนา/ผู้ทดสอบพัฒนาข้อกำหนดการทดสอบ การออกแบบการทดสอบและออกแบบกรณีทดสอบซึ่งเป็นส่วนหนึ่งของแผนงานการทดสอบ นอกจากนี้ยังเข้าร่วมการวางแผนการทดสอบในการกำหนดขั้นตอน เครื่องมือ กำหนดการและงบประมาณ
- ผู้ใช้/ลูกค้าร่วมในการพัฒนาแผนงานการทดสอบเพื่อยอมรับ

องค์กรควรจะติดตามสถานะของการพัฒนาแผ่นแบบของแผนงานการทดสอบ ที่จะถูกใช้เป็นนโยบายในการวางแผนการทดสอบสำหรับองค์กร คำถามและมาตรวัดคือ

คำถาม 3.1.2.1 อะไรแสดงสถานะของการพัฒนาแผ่นแบบของแผนงานการทดสอบ?

มาตรวัด 3.1.2.1 ค่าใช้จ่ายในรูปของความพยายามในการพัฒนาแผ่นแบบของแผนงานการทดสอบ

ในแต่ละโครงการ ผู้จัดการควรจะติดตามสถานะของการพัฒนาแผนงานการทดสอบ และกำหนดกิจกรรมต่างๆ ในแผนงาน คำถามและมาตรวัดคือ

คำถาม 3.1.2.2 อะไรแสดงสถานะของการพัฒนาแผนงานการทดสอบในแต่ละโครงการ ?

มาตรวัด 3.1.2.2 ค่าใช้จ่ายในรูปของความพยายามในการพัฒนาแผนงานการทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ

คำถาม 3.1.2.3 อะไรใช้เป็นตัวติดตามสถานะของกิจกรรมในแผนงานการทดสอบ?

มาตรวัด 3.1.2.3 - *วันเริ่มต้นของกิจกรรมตามแผนงาน

- *วันสิ้นสุดของกิจกรรมตามแผนงาน

- ค่าใช้จ่ายในรูปของตัวเงินหรือความพยายามที่ใช้ไปในกิจกรรมตามแผนงาน

- จำนวนบุคลากรที่ทำในกิจกรรมตามแผนงาน

3.1.3 เป้าหมายจัดตั้งเทคนิคและวิธีการพื้นฐานในการทดสอบ

จุดประสงค์ของเป้าหมายนี้ คือระบุว่าเทคนิคพื้นฐานจะถูกใช้เมื่อใดและอย่างไร ระบุความต้องการในการฝึกอบรมเทคนิคพื้นฐานและระบุทรัพยากรที่ต้องการในการทำให้เกิดเทคนิคพื้นฐานเช่น เทคนิคอีควิวเลนซ์ พาร์ทิชันนิ่ง (Equivalence partitioning) เทคนิคการวิเคราะห์ค่าที่ขอบ (Boundary value analysis) และเทคนิคการทดสอบแบบไวท์บ็อกซ์ (Whitebox testing)

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) จัดตั้งเทคนิคและวิธีการพื้นฐานในการทดสอบ
- 2) ระบุและจัดหาเครื่องมือในการทดสอบแบบพื้นฐาน

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และ ผู้ใช้/ลูกค้า มีดังต่อไปนี้

- ผู้จัดการจัดสรรทรัพยากรเงินและให้ นโยบายในการจัดตั้งเทคนิค วิธีการและเครื่องมือพื้นฐานในการทดสอบ
- ผู้พัฒนา/ผู้ทดสอบใช้เทคนิคพื้นฐานในการทดสอบ
- ผู้ใช้/ลูกค้าระบุเงื่อนไขอินพุตและเอาต์พุตของซอฟต์แวร์ และความต้องการทางฟังก์ชันสำหรับการทดสอบแบบแบล็คบ็อกซ์ (Blackbox testing)

เพื่อที่จะติดตามการใช้เทคนิคและวิธีการพื้นฐานในการทดสอบ องค์กรควรติดตามผลผลิตที่เกิดจากการใช้ เช่น กรณีทดสอบ และข้อบกพร่อง คำถามและมาตรวัดคือ

คำถาม 3.1.3.1 อะไรแสดงว่าองค์กรใช้เทคนิคและวิธีการพื้นฐานในการทดสอบ ?

- มาตรวัด 3.1.3.1 - จำนวนกรณีทดสอบที่ถูกพัฒนาจากเทคนิคการทดสอบแบบแบล็คบ็อกซ์
- จำนวนกรณีทดสอบที่ถูกพัฒนาจากเทคนิคการทดสอบแบบไวท์บ็อกซ์
 - จำนวนข้อบกพร่องที่ถูกตรวจพบ โดยเทคนิคการทดสอบแบบแบล็คบ็อกซ์
 - จำนวนข้อบกพร่องที่ถูกตรวจพบ โดยเทคนิคการทดสอบแบบไวท์บ็อกซ์

หมายเหตุ ในงานวิจัยนี้ ไม่ได้ให้นิยามของข้อบกพร่อง องค์กรจะต้องกำหนดเองว่าอะไรควรจะเป็นข้อบกพร่อง

ที่เอ็มเอ็มกล่าวว่าผู้จัดการต้องระบุและจัดหาเครื่องมือการทดสอบพื้นฐานที่องค์กรต้องมี คำถามและมาตรวัดคือ

คำถาม 3.1.3.2 องค์กรมีเครื่องมือทดสอบอะไรบ้าง เช่น โปรแกรมตรวจสอบข้อบกพร่อง สเตติกอนาไลเซอร์ (Static analyzers) ?

มาตรวัด 3.1.3.2 - จำนวนเครื่องมือการทดสอบพื้นฐาน

- มูลค่าในรูปตัวเงินของเครื่องมือการทดสอบ

โดยสรุปมาตรวัดขั้นต่ำสำหรับระดับ 2 เป็นไปตามตารางที่ 3.1

เป้าหมาย	มุมมอง	มาตรวัด	หน่วย
3.1.1	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ	- ความพยายามในการทดสอบ - ความพยายามในการแก้จุดบกพร่อง	คน. ชั่วโมง คน. ชั่วโมง
3.1.2	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- ความพยายามในการพัฒนาแผนงานการทดสอบ - กำหนดการของกิจกรรม	คน. ชั่วโมง วัน
3.1.3	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ	- จำนวนกรณีทดสอบที่ได้มาจากเทคนิคการทดสอบแบบเบสิคบล็อกซ์และไวท์บล็อกซ์	กรณีทดสอบ

ตาราง 3.1 มาตรวัดขั้นต่ำสำหรับทีเอ็มเอ็มในระดั 2

3.2 มาตรวัดในระดับ 3 การรวม

ระดับ 3 ของทีเอ็มเอ็มประกอบไปด้วยเป้าหมายทั้งหมด 4 เป้าหมาย ซึ่งในแต่ละเป้าหมายมีรายละเอียดต่างๆ แสดงถึงกิจกรรมที่จะต้องกระทำเพื่อให้บรรลุเป้าหมาย โดยเป้าหมายในระดับ 3 มีดังต่อไปนี้

3.2.1 เป้าหมายจัดตั้งหน่วยงานการทดสอบซอฟต์แวร์

จุดประสงค์สำหรับเป้าหมายนี้คือ เพื่อมีบุคลากรที่จะทำหน้าที่ด้านการทดสอบโดยเฉพาะ ซึ่งจะทำงานร่วมกับหน่วยงานรับประกันคุณภาพซอฟต์แวร์ (Software Quality Assurance) ในการกำหนดแผนงานและมาตรฐานการทดสอบ เป้าหมายในการทดสอบ และควบคุมและติดตามการทดสอบขององค์กรให้เป็นไปตามแผน

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) จัดตั้งหน่วยงานการทดสอบซอฟต์แวร์ขององค์กร
- 2) จัดตั้งบทบาทและความรับผิดชอบของหน่วยงานการทดสอบซอฟต์แวร์
- 3) จัดสรรบุคลากรให้แก่หน่วยงานการทดสอบซอฟต์แวร์
- 4) ช่วยหน่วยงานรับประกันคุณภาพซอฟต์แวร์ ในการเพิ่มประสิทธิผลการทดสอบและปรับปรุงคุณภาพซอฟต์แวร์

กิจกรรมที่จะต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และ ผู้ใช้/ลูกค้า มีดังต่อไปนี้

- ผู้จัดการจัดสรรทรัพยากรและเงิน จัดหาบุคลากรให้ นโยบายและมอบหมายความรับผิดชอบแก่หน่วยงานการทดสอบซอฟต์แวร์
- ผู้พัฒนา/ผู้ทดสอบรับบทบาทและความรับผิดชอบในการทดสอบ
- ผู้ใช้/ลูกค้าติดต่อกับหน่วยงานการทดสอบในการตั้งกฎเกณฑ์ในการทดสอบเพื่อยอมรับเข้าร่วมในการทบทวน และให้ข้อมูลในการใช้ซอฟต์แวร์

เพื่อที่จะทราบว่ามีการจัดตั้งหน่วยงานการทดสอบ องค์กรควรจะติดตามขนาดและค่าใช้จ่ายของหน่วยงาน คำถามและมาตรวัดคือ

คำถาม 3.2.1.1 ขนาดของหน่วยงานการทดสอบซอฟต์แวร์วัดได้จากอะไร?

มาตรวัด 3.2.1.1 จำนวนของผู้ทดสอบ

คำถาม 3.2.1.2 ผู้จัดการของหน่วยงานการทดสอบควบคุมหน่วยงานได้ดีเพียงใด?

มาตรวัด 3.2.1.2

$$\% \text{ ความแตกต่างของค่าใช้จ่ายของหน่วยงาน} = \frac{\text{ค่าใช้จ่ายจริง} - \text{ค่าใช้จ่ายตามงบประมาณ}}{\text{ค่าใช้จ่ายตามงบประมาณ}} * 100$$

3.2.2 เป้าหมายจัดตั้งโปรแกรมการฝึกอบรมทางเทคนิค

จุดประสงค์สำหรับเป้าหมายนี้คือ การเพิ่มทักษะทางเทคนิคของบุคลากรในการทดสอบและการรับประกันคุณภาพ โปรแกรมการฝึกอบรมจะสอนให้บุคลากรใช้เครื่องมือการทดสอบ ช่วยให้เข้าใจกระบวนการทดสอบขององค์กร ช่วยปรับปรุงการทดสอบและคุณภาพของผลิตภัณฑ์ซอฟต์แวร์ และช่วยให้เข้าใจกระบวนการวางแผนการทดสอบ

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) จัดตั้งหน่วยงานการฝึกอบรมขององค์กร
- 2) จัดตั้งแผนงานการฝึกอบรม
- 3) ปรับปรุงทักษะและเทคนิคในการทดสอบ

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า มีดังต่อไปนี้

- ผู้จัดการจัดสรรทรัพยากรและเงินทุนในการจัดตั้งหน่วยงานการฝึกอบรม ควบคุมและติดตามว่าโปรแกรมการฝึกอบรมเป็นไปตามแผนหรือไม่
- ผู้พัฒนา/ผู้ทดสอบ ระบุความต้องการเนื้อหาในการอบรมและใช้ทักษะที่ได้รับในการฝึกอบรมในงานต่างๆ
- ผู้ใช้/ลูกค้าเข้ารับการฝึกอบรมเทคนิคในการทบทวน

องค์กรควรจะให้ปริมาณการฝึกอบรมทางเทคนิคเกี่ยวกับการทดสอบในระดับที่เหมาะสมแก่บุคลากร คำถามและมาตรวัดคือ

คำถาม 3.2.2.1 บุคลากรได้รับการฝึกอบรมเกี่ยวกับการทดสอบมากน้อยเพียงใด?

มาตรวัด 3.2.2.1 - จำนวนหลักสูตรเกี่ยวกับการทดสอบต่อปี
- จำนวนชั่วโมงที่บุคลากรได้เข้ารับการอบรมต่อปี

องค์กรควรจะติดตามสถานะของโปรแกรมการฝึกอบรม คำถามและมาตรวัดคือ

คำถาม 3.2.2.2 โปรแกรมการฝึกอบรมเป็นไปตามที่ได้วางแผนหรือไม่?

มาตรวัด 3.2.2.2 - ความแตกต่างของกำหนดการ = วันที่ฝึกอบรมจริง - วันที่ฝึกอบรมตามแผนงาน
- % ความแตกต่างของค่าใช้จ่ายในการฝึกอบรม

$$= \frac{\text{ค่าใช้จ่ายจริง} - \text{ค่าใช้จ่ายตามแผนงาน}}{\text{ค่าใช้จ่ายตามแผนงาน}} * 100$$

- % ความแตกต่างของจำนวนผู้เข้ารับการอบรม

$$= \frac{\text{จำนวนผู้เข้ารับการอบรมจริง} - \text{ตามแผนงาน}}{\text{จำนวนผู้เข้ารับการอบรมตามแผนงาน}} * 100$$

3.2.3 เป้าหมายรวมการทดสอบเข้าไปในวงจรพัฒนาซอฟต์แวร์

จุดประสงค์สำหรับเป้าหมายนี้คือการทำกิจกรรมการทดสอบขนานกับวงจรพัฒนาซอฟต์แวร์ตามวีโมเดล ทุกๆ ขั้นตอนมีการระบุความต้องการในการทดสอบ สิ่งที่จะทดสอบและใครเป็นผู้รับผิดชอบ

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) แบ่งกิจกรรมการทดสอบเข้ากับทุกๆ ขั้นตอนในวงจรพัฒนาซอฟต์แวร์
- 2) เริ่มกิจกรรมการทบทวนเพื่อสนับสนุนเป้าหมายย่อยในข้อ 1
- 3) เริ่มกำหนดมาตรฐานการทดสอบ
- 4) ปรับปรุงคุณภาพของผลิตภัณฑ์

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า เป็นดังต่อไปนี้

- ผู้จัดการให้การสนับสนุนและมอบหมายความรับผิดชอบให้แก่หน่วยงานการทดสอบซอฟต์แวร์ ในการรวมการทดสอบเข้าไปในวงจรพัฒนาซอฟต์แวร์ตามวีโมเดล
- ผู้พัฒนา/ผู้ทดสอบกำหนดตัวโมเดลและดำเนินกิจกรรมการทดสอบ รวมถึงได้รับการฝึกอบรมเพื่อให้ตระหนักว่าการทดสอบไม่เป็นเพียงแค่ขั้นตอนหนึ่งหลังจากขั้นตอนการเขียนโปรแกรม

- ผู้ใช้/ลูกค้าเข้าร่วมกิจกรรมการทดสอบตามวีโมเดล

องค์กรควรจะติดตามผลลัพธ์ของการรวมการทดสอบเข้าไปในวงจรพัฒนาซอฟต์แวร์ ซึ่งอาจจะติดตามจำนวนข้อบกพร่องที่ถูกรับในแต่ละขั้นตอน คำถามและมาตรวัดคือ

คำถาม 3.2.3.1 ประสิทธิภาพของการรวมการทดสอบเข้าไปในวงจรพัฒนาซอฟต์แวร์เป็นอย่างไร?

มาตรวัด 3.2.3.1 - จำนวนข้อบกพร่องที่ถูกรับในแต่ละขั้นตอนของการทดสอบในวงจรพัฒนาซอฟต์แวร์

- ประสิทธิภาพของการจัดข้อบกพร่องในขั้นตอน_i (%) [9]

$$= \frac{\text{จำนวนข้อบกพร่องที่ถูกจัดในขั้นตอน}_i}{\text{จำนวนข้อบกพร่องที่มาจากขั้นตอน}_{i-1} + \text{จำนวนข้อบกพร่องที่เกิดจากขั้นตอน}_i} * 100$$

โดยที่ขั้นตอน_i คือ การวิเคราะห์ความต้องการ การออกแบบ การเขียนโปรแกรม การทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ

ขั้นตอน_{i-1} คือ ขั้นตอนก่อนขั้นตอน_i

3.2.4 เป้าหมายควบคุมและติดตามกระบวนการทดสอบ

จุดประสงค์สำหรับเป้าหมายนี้คือ ทำให้เห็นและติดตามกิจกรรมการทดสอบ และความก้าวหน้าในการทดสอบว่าเป็นไปตามแผนหรือไม่ ถ้าไม่จะได้แก้ไขอย่างทันท่วงที

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) พัฒนากลไกและนโยบายในการควบคุม และติดตามกระบวนการทดสอบโดยเปรียบเทียบกับแผนงานการทดสอบ
- 2) วัดกิจกรรมและคุณภาพในการทดสอบ
- 3) จัดตั้งวิธีการแก้ไขปัญหาและแผนงานเฉพาะเมื่อกระบวนการทดสอบ และความก้าวหน้าในการทดสอบไม่เป็นไปตามแผนงานที่วางไว้

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า เป็นดังต่อไปนี้

- ผู้จัดการด้านการทดสอบ ติดตามสถานะการทดสอบ จัดตั้งมาตรฐานสำหรับผลผลิตที่เกิดจากการทดสอบ ตั้งมาตรฐานมาตรวัดในการติดตามความก้าวหน้าของการทดสอบ ดำเนินการแก้ไขเมื่อการทดสอบไม่เป็นไปตามแผน กำหนดรายงานความก้าวหน้าของการทดสอบ และเปรียบเทียบค่าใช้จ่าย เวลา ความพยายามและจำนวนบุคลากรที่เกิดขึ้นจริงกับแผนงาน

- ผู้พัฒนา/ผู้ทดสอบ หน่วยงานการทดสอบซอฟต์แวร์กำหนดและเก็บข้อมูลมาตรวัดสำหรับความก้าวหน้าของการทดสอบ เช่นความครอบคลุมความต้องการ ความครอบคลุมทางเดิน (Path coverage) และรายงานสถานะของการทดสอบ เช่น สถานะของการออกแบบการทดสอบ สถานะของกรณีทดสอบ
- ผู้ใช้/ลูกค้า เข้าร่วมในการทดสอบความสามารถในการใช้งาน (Usability) ของซอฟต์แวร์ที่ขั้นตอนต่างๆ ของวงจรพัฒนาซอฟต์แวร์

องค์กรจะต้องติดตามและควบคุมกิจกรรมต่างๆที่วางไว้ในแผนงานการทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.2.4.1 กิจกรรมที่เกี่ยวข้องกับการทดสอบเป็นไปตามแผนการทดสอบหรือไม่?

มาตรวัด 3.2.4.1 - ความแตกต่างของวันเริ่มกิจกรรม = $\frac{\text{วันเริ่มกิจกรรมจริง} - \text{ตามแผนงาน}}$

- ความแตกต่างของวันสิ้นสุดกิจกรรม = $\frac{\text{วันสิ้นสุดกิจกรรมจริง} - \text{ตามแผนงาน}}$

- %ความแตกต่างของค่าใช้จ่ายในรูปตัวเงินหรือความพยายาม

$$= \frac{\text{ค่าใช้จ่ายจริง} - \text{ตามแผนงาน}}{\text{ค่าใช้จ่ายตามแผนงาน}} * 100$$

ค่าใช้จ่ายตามแผนงาน

- %ความแตกต่างของจำนวนบุคลากร = $\frac{\text{จำนวนบุคลากรจริง} - \text{ตามแผนงาน}}{\text{จำนวนบุคลากรตามแผนงาน}} * 100$

จำนวนบุคลากรตามแผนงาน

องค์กรจะต้องติดตามสถานะกรณีทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.2.4.2 สถานะของกรณีทดสอบในแต่ละขั้นตอนของการทดสอบเป็นอย่างไร?

มาตรวัด 3.2.4.2 - %กรณีทดสอบที่ทดสอบผ่าน = $\frac{\text{จำนวนกรณีทดสอบที่ผ่าน}}{\text{จำนวนกรณีทดสอบทั้งหมด}} * 100$

จำนวนกรณีทดสอบทั้งหมด

- %กรณีทดสอบที่ทดสอบไม่ผ่าน = $\frac{\text{จำนวนกรณีทดสอบที่ไม่ผ่าน}}{\text{จำนวนกรณีทดสอบทั้งหมด}} * 100$

จำนวนกรณีทดสอบทั้งหมด

โดยที่ขั้นตอนของการทดสอบคือ การทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ

องค์กรควรจะต้องติดตามสถานะของโมดูลที่นำมาทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.2.4.3 สถานะของของโมดูลที่นำมาทดสอบเป็นอย่างไร ?

มาตรวัด 3.2.4.3 - %โมดูลที่ทดสอบผ่าน = $\frac{\text{จำนวนโมดูลที่ทดสอบผ่าน}}{\text{จำนวนโมดูลทั้งหมด}} * 100$

จำนวนโมดูลทั้งหมด

$$- \% \text{โมดูลที่ทดสอบไม่ผ่าน} = \frac{\text{จำนวนโมดูลที่ทดสอบไม่ผ่าน}}{\text{จำนวนโมดูลทั้งหมด}} * 100$$

องค์กรควรจะติดตามคุณภาพของการทดสอบโมดูลซึ่งความครอบคลุมการทดสอบเป็นดัชนีหนึ่งซึ่งถึงคุณภาพ คำถามและมาตรวัดคือ

คำถาม 3.2.4.4 วัดความครอบคลุมการทดสอบของการทดสอบโมดูลได้อย่างไร?

มาตรวัด 3.2.4.4 - %ความครอบคลุมข้อความสั่ง (Statement coverage)

$$= \frac{\text{ข้อความสั่งที่ถูกทดสอบ}}{\text{ข้อความสั่งทั้งหมด}} * 100$$

- %ความครอบคลุมกิ่ง (Branch coverage)

$$= \frac{\text{จำนวนกิ่งที่ถูกทดสอบ}}{\text{จำนวนกิ่งทั้งหมด}} * 100$$

- %ความครอบคลุมทางเดินพื้นฐาน (Basis path coverage)

$$= \frac{\text{จำนวนทางเดินพื้นฐานที่ถูกทดสอบ}}{\text{จำนวนทางเดินพื้นฐานทั้งหมด}} * 100$$

องค์กรควรจะติดตามและควบคุมคุณภาพของโมดูลก่อนนำไปทำการทดสอบเบ็ดเสร็จ ซึ่งจำนวนข้อบกพร่องของโมดูลเป็นดัชนีซึ่งถึงคุณภาพ คำถามและมาตรวัดคือ

คำถาม 3.2.4.5 วัดคุณภาพของโมดูลได้อย่างไร?

มาตรวัด 3.2.4.5 ความหนาแน่นของข้อบกพร่อง (Defect density) = $\frac{\text{จำนวนข้อบกพร่องของโมดูล}}{\text{จำนวนบรรทัดคำสั่งของโมดูล}}$

องค์กรควรจะติดตามสถานะของการทดสอบเบ็ดเสร็จ คำถามและมาตรวัดคือ

คำถาม 3.2.4.6 วัดความก้าวหน้าของการทดสอบเบ็ดเสร็จได้อย่างไร?

มาตรวัด 3.2.4.6 - %โมดูลที่นำมารวม = $\frac{\text{จำนวนโมดูลที่นำมารวม}}{\text{จำนวนโมดูลทั้งหมด}} * 100$

$$- \% \text{โมดูลที่นำมารวมสำเร็จ} = \frac{\text{จำนวนโมดูลที่นำมารวมสำเร็จ}}{\text{จำนวนโมดูลทั้งหมด}} * 100$$

องค์กรควรจะติดตามและควบคุมคุณภาพของโมดูลที่นำมารวม ซึ่งจำนวนข้อบกพร่องของโมดูลที่นำมารวม เป็นดัชนีซึ่งถึงคุณภาพ คำถามและมาตรวัดคือ

คำถาม 3.2.4.7 วัดคุณภาพของโมดูลที่นำมารวมระหว่างการทดสอบเบ็ดเสร็จได้อย่างไร?

มาตรวัด 3.2.4.7 ความหนาแน่นของข้อบกพร่อง = $\frac{\text{จำนวนข้อบกพร่องของโมดูลที่นำมารวม}}{\text{จำนวนบรรทัดคำสั่งของโมดูลที่นำมารวม}}$

องค์กรควรจะติดตามสถานะของการทดสอบความต้องการระหว่างการทดสอบระบบ และการทดสอบเพื่อยอมรับ คำถามและมาตรวัดคือ

คำถาม 3.2.4.8 วัดความก้าวหน้าของการทดสอบความต้องการได้อย่างไร?

มาตรวัด 3.2.4.8 - %ความต้องการที่ทดสอบ = $\frac{\text{จำนวนความต้องการที่ทดสอบ}}{\text{จำนวนความต้องการทั้งหมด}} * 100$

- %ความต้องการที่ทดสอบผ่าน = $\frac{\text{จำนวนความต้องการที่ทดสอบผ่าน}}{\text{จำนวนความต้องการทั้งหมด}} * 100$

องค์กรควรจะติดตามและควบคุมคุณภาพของซอฟต์แวร์ระหว่างการทดสอบระบบ และการทดสอบเพื่อยอมรับ ซึ่งข้อบกพร่องของระบบเป็นดัชนีชี้ถึงคุณภาพ คำถามและมาตรวัดคือ

คำถาม 3.2.4.9 คุณภาพของซอฟต์แวร์ระหว่างการทดสอบระบบและการทดสอบเพื่อยอมรับเป็นอย่างไร?

มาตรวัด 3.2.4.9 ความหนาแน่นของข้อบกพร่อง_t = $\frac{\text{จำนวนข้อบกพร่องของซอฟต์แวร์}}{\text{จำนวนบรรทัดคำสั่งของซอฟต์แวร์}}$

โดยที่ t คือช่วงระยะเวลาที่ทำการทดสอบ

มาตรวัดนี้หาค่าความหนาแน่นของข้อบกพร่องเป็นช่วงระยะเวลาเช่น 1 สัปดาห์ ตั้งแต่การทดสอบระบบว่ามีจำนวนข้อบกพร่องเท่าใด หาค่าด้วยจำนวนบรรทัดคำสั่งของซอฟต์แวร์ แล้วดูว่าระยะเวลาผ่านไป ความหนาแน่นของข้อบกพร่องเป็นอย่างไร

โดยสรุป มาตรวัดขั้นต่ำสำหรับระดับ 3 เป็นไปตามตารางที่ 3.2

เป้าหมาย	มุมมอง	มาตรวัด	หน่วย
3.2.1	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ	- จำนวนของผู้ทดสอบ - ค่าใช้จ่ายของหน่วยงานการทดสอบ	คน ฿
3.2.2	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ	- จำนวนหลักสูตรเกี่ยวกับการทดสอบ	หลักสูตรต่อปี
3.2.3	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- จำนวนข้อบกพร่องในแต่ละขั้นตอน	ข้อบกพร่อง

ตาราง 3.2 มาตรวัดขั้นต่ำสำหรับที่เอ็มเอ็มในระดับ 3

เป้าหมาย	มุมมอง	มาตรวัด	หน่วย
3.2.4	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- ความแตกต่างของกำหนดการ	วัน
		- จำนวนกรณีทดสอบที่ผ่านและไม่ผ่าน	กรณีทดสอบ
		- จำนวนโมดูลที่ทดสอบผ่านและไม่ผ่าน	โมดูล
		- ความครอบคลุมข้อความสั่ง	%
		- ความครอบคลุมกิ่ง	%
		- ความครอบคลุมทางเดินพื้นฐาน	%
		- จำนวนโมดูลที่นำมารวม	โมดูล

ตาราง 3.2 มาตรวัดขั้นต่ำสำหรับทีเอ็มเอ็มในระดับ 3 (ต่อ)

3.3 มาตรวัดในระดับ 4 การจัดการและการวัด

ระดับ 4 ของทีเอ็มเอ็มประกอบไปด้วยเป้าหมายทั้งหมด 3 เป้าหมาย ซึ่งในแต่ละเป้าหมายมีรายละเอียดต่างๆ แสดงถึงกิจกรรมที่จะต้องกระทำเพื่อให้บรรลุเป้าหมาย โดยเป้าหมายในระดับ 4 มีดังต่อไปนี้

3.3.1 เป้าหมายจัดตั้งโปรแกรมการทบทวนทั่วทั้งองค์กร

เป้าหมายจัดตั้งโปรแกรมการทบทวนทั่วทั้งองค์กร คือการรวมการทบทวนเข้าไปในทุกๆ ขั้นตอนของวงจรการพัฒนาซอฟต์แวร์ตามรูปที่ 2.1 ในบทที่ 2 เพื่อค้นหาข้อบกพร่องที่ผลิตที่เกิดจากการพัฒนาซอฟต์แวร์ (Software work products) และผลผลิตที่เกิดจากการทดสอบ (Test work products) ได้อย่างมีประสิทธิภาพ

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) พัฒนาแผนงานการทบทวน
- 2) ระบุผลผลิตที่เกิดจากการพัฒนาซอฟต์แวร์ และผลผลิตที่เกิดจากการทดสอบที่จะต้องทบทวน
- 3) ระบุข้อบกพร่องในผลผลิตที่เกิดจากการทำงาน
- 4) จัดตั้งกระบวนการในการติดตามการแก้ไขข้อบกพร่องที่ถูกระบุจากการทบทวน
- 5) ทำรายงานผลลัพธ์การทบทวน

กิจกรรมของการทบทวนตามทีเอ็มเอ็มในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า เป็นดังต่อไปนี้

- ผู้จัดการต้องให้นโยบายการทบทวนจัดสรรทรัพยากรที่จะใช้ในการทบทวนและทำแผนงานการทบทวน (Review plan)
- ผู้พัฒนา/ผู้ทดสอบทำการทบทวนตามแผนงานที่วางไว้ ซึ่งสิ่งที่นำมาทบทวนก็คือ ผลผลิตที่เกิดจากการพัฒนาซอฟต์แวร์และผลผลิตที่เกิดจากการทดสอบ ผลผลิตที่เกิดจาก

การพัฒนาซอฟต์แวร์ได้แก่ เอกสารความต้องการ เอกสารการออกแบบ และโปรแกรม ส่วนผลผลิตที่เกิดจากการทดสอบได้แก่ แผนงานการทดสอบ ข้อกำหนดกรณีทดสอบ ขั้นตอนปฏิบัติการทดสอบ และเอกสารการทดสอบอื่นๆ หลังจากทำการทบทวนเสร็จ จะต้องตัดสินใจว่าจะยอมรับหรือไม่ แล้วทำรายงานผลของการทบทวนให้แก่ผู้ที่เกี่ยวข้อง

- ผู้ใช้/ลูกค้าทำการทบทวนเอกสารความต้องการและเอกสารการทดสอบเพื่อยอมรับว่าเป็นไปตามความต้องการหรือไม่

องค์กรควรจะควบคุมประสิทธิผลของการทบทวนซึ่งจำนวนข้อบกพร่องที่ถูกขจัดโดยการทบทวนเป็นดัชนีชี้ตัวหนึ่ง คำถามและมาตรวัดคือ

คำถาม 3.3.1.1 วัดประสิทธิผลของการทบทวนได้อย่างไร?

มาตรวัด 3.3.1.1 %ประสิทธิผลของการทบทวน

$$= \frac{\text{จำนวนข้อบกพร่องที่ถูกขจัดโดยการทบทวน} * 100}{\text{จำนวนข้อบกพร่องทั้งหมด}}$$

องค์กรควรจะติดตามสถานะของการทบทวน คำถามและมาตรวัดคือ

คำถาม 3.3.1.2 การทบทวนเป็นไปตามแผนงานการทบทวนหรือไม่?

มาตรวัด 3.3.1.2 - ความแตกต่างของกำหนดการ = กำหนดการจริง - ตามแผนงาน

- %ความแตกต่างของเวลาในการเตรียมการ (Preparation Time)

$$= \frac{\text{เวลาในการเตรียมการจริง} - \text{ตามแผนงาน}}{\text{เวลาในการเตรียมการตามแผนงาน}} * 100$$

- %ความแตกต่างของเวลาในการตรวจ (Inspection Time)

$$= \frac{\text{เวลาในการตรวจจริง} - \text{ตามแผนงาน}}{\text{เวลาในการตรวจตามแผนงาน}} * 100$$

- %ความแตกต่างของจำนวนผู้ทบทวน

$$= \frac{\text{จำนวนผู้ทบทวนจริง} - \text{ตามแผนงาน}}{\text{จำนวนผู้ทบทวนตามแผนงาน}} * 100$$

- %ความครอบคลุมการทบทวนความต้องการ

$$= \frac{\text{จำนวนความต้องการที่ถูกทบทวน}}{\text{จำนวนความต้องการทั้งหมด}} * 100$$

- %ความครอบคลุมการทบทวนการออกแบบ

$$= \frac{\text{จำนวนหน้าของเอกสารการออกแบบที่ถูกทบทวน}}{\text{จำนวนหน้าของเอกสารการออกแบบทั้งหมด}} * 100$$

- %ความครอบคลุมการทบทวนโปรแกรม

$$= \frac{\text{จำนวนบรรทัดคำสั่งของโปรแกรมที่ถูกทบทวน}}{\text{จำนวนบรรทัดคำสั่งทั้งหมด}} * 100$$

- อัตราการทบทวนเอกสาร = $\frac{\text{จำนวนหน้าของเอกสาร}}{\text{เวลาในการเตรียมการ+เวลาในการตรวจ}}$ หน้า/ชั่วโมง

โดยที่อัตราการทบทวนเอกสารเป็นอัตราการทบทวนของเอกสารแต่ละชนิดที่นำมาตรวจทั้งเอกสารที่เกิดจากการพัฒนาซอฟต์แวร์และเอกสารที่เกิดจากการทดสอบ

- อัตราการทบทวนโปรแกรม

$$= \frac{\text{จำนวนบรรทัดคำสั่ง}}{\text{เวลาในการเตรียมการ+เวลาในการตรวจ}}$$
 บรรทัดคำสั่ง/ชั่วโมง

3.3.2 เป้าหมายจัดตั้งโปรแกรมการวัดการทดสอบ

จุดประสงค์สำหรับเป้าหมายนี้คือ ให้ข้อมูลที่เป็นประโยชน์สำหรับองค์กรในการประเมินคุณภาพของกระบวนการทดสอบ ประเมินผลผลิตของบุคลากร และใช้ข้อมูลที่ได้จากการวัดในการปรับปรุงกระบวนการทดสอบ

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) จัดตั้งโปรแกรมการวัดการทดสอบขององค์กร
- 2) พัฒนาแผนงานการวัดกระบวนการทดสอบ
- 3) เริ่มกลไกในการกระจายมาตรวัดและการปรับปรุงกระบวนการทดสอบ
- 4) เก็บและบันทึกข้อมูลมาตรวัดการทดสอบ

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า เป็นดังต่อไปนี้

- ผู้จัดการ สนับสนุนและจัดสรรทรัพยากรในการจัดตั้งโปรแกรมการวัดการทดสอบ ผู้จัดการด้านการทดสอบพัฒนาแผนงานการวัดการทดสอบ วิธีการในการตรวจสอบความถูกต้องของข้อมูลและมาตรวัด และใช้ข้อมูลที่ได้จากการวัดในการควบคุมกระบวนการทดสอบ
- ผู้พัฒนา/ผู้ทดสอบ ปฏิบัติตามโปรแกรมการวัดการทดสอบในการเก็บข้อมูลที่เกี่ยวข้องกับความก้าวหน้าในการทดสอบ ข้อมูลเกี่ยวกับข้อบกพร่อง ค่าใช้จ่ายในการทดสอบ และความน่าเชื่อถือ (Reliability) ของซอฟต์แวร์

- ผู้ใช้/ลูกค้า ให้มาตรวัดในมุมมองของเขา เพื่อใช้ในการวัดความพอใจ มีการวัดค่าใช้จ่ายที่ผู้ใช้/ลูกค้าร่วมในการทดสอบเพื่อยอมรับและการทดสอบความสามารถในการใช้งาน

ที่เอ็มเอ็มกล่าวว่าองค์กรจะต้องเก็บรายละเอียดของข้อบกพร่อง คำถามและมาตรวัดคือ

คำถาม 3.3.2.1 รายละเอียดของข้อบกพร่องอะไรบ้างที่ต้องเก็บ?

มาตรวัด 3.3.2.1 - จำนวนข้อบกพร่องในแต่ละระดับความรุนแรง (Severity levels)

- จำนวนข้อบกพร่องในแต่ละประเภทของข้อบกพร่อง (Defect types)

องค์กรควรจะติดตามสถานะของการจัดตั้งโปรแกรมการวัดการทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.3.2.2 การจัดตั้งโปรแกรมการวัดการทดสอบเป็นไปตามแผนหรือไม่?

มาตรวัด 3.3.2.2 - ความแตกต่างของวันเริ่มกิจกรรม = วันเริ่มกิจกรรมจริง - ตามแผนงาน

- ความแตกต่างของวันสิ้นสุดกิจกรรม = วันสิ้นสุดกิจกรรมจริง - ตามแผนงาน

- %ความแตกต่างของค่าใช้จ่ายในรูปตัวเงินหรือความพยายาม

$$= \frac{\text{ค่าใช้จ่ายจริง} - \text{ตามแผนงาน}}{\text{ค่าใช้จ่ายตามแผนงาน}} * 100$$

ค่าใช้จ่ายตามแผนงาน

- %ความแตกต่างของจำนวนบุคลากร = $\frac{\text{จำนวนบุคลากรจริง} - \text{ตามแผนงาน}}{\text{จำนวนบุคลากรตามแผนงาน}} * 100$

จำนวนบุคลากรตามแผนงาน

องค์กรควรจะกำหนดว่าจะประเมินกระบวนการทดสอบอย่างไร คำถามและมาตรวัดคือ

คำถาม 3.3.2.3 อะไรเป็นดัชนีที่จะชี้ให้เห็นถึงคุณภาพกระบวนการทดสอบ?

มาตรวัด 3.3.2.3 - ประสิทธิภาพของการจัดข้อบกพร่องตามมาตรวัด 3.2.3.1

- ความครอบคลุมการทดสอบ ได้แก่ ความครอบคลุมข้อความสั่ง ความครอบคลุมถึง ความครอบคลุมทางเค้นพื้นฐาน ความครอบคลุมการทบทวน

- ค่าใช้จ่ายในการทดสอบ

- ความหนาแน่นข้อบกพร่อง = $\frac{\text{จำนวนข้อบกพร่องที่พบในขั้นตอนการทดสอบ}}{\text{ขนาดของผลผลิตที่ทดสอบ}}$

ขนาดของผลผลิตที่ทดสอบ

โดยที่ ขั้นตอนการทดสอบ คือการทบทวนความต้องการ การทบทวนการออกแบบ การทบทวนโปรแกรม การทบทวนการทดสอบโมดูล การทดสอบโมดูล การทบทวนการ

ทดสอบเบ็ดเสร็จ การทดสอบเบ็ดเสร็จ การทบทวนการทดสอบระบบ การทดสอบระบบ การทบทวนการทดสอบเพื่อยอมรับ การทดสอบเพื่อยอมรับ

ทีเอ็มเอ็มกล่าวไว้ว่าองค์กรควรประเมินผลผลิต (Productivity) ของผู้ทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.3.2.4 วัดผลผลิตของผู้ทดสอบได้อย่างไร?

มาตรวัด 3.3.2.4 - ผลผลิตการพัฒนาระณีทดสอบ = $\frac{\text{จำนวนของกรณีทดสอบ}}{\text{ความพยายามในการพัฒนาระณีทดสอบ}}$

- ผลผลิตในการทดสอบ = $\frac{\text{ขนาดของผลผลิตที่นำมาทดสอบ}}{\text{ความพยายามในการทดสอบ}}$

โดยที่ผลผลิตที่นำมาทดสอบคือ เอกสารความต้องการ เอกสารการออกแบบ โปรแกรม เอกสารการทดสอบต่างๆ เช่น ข้อกำหนดกรณีทดสอบ ขั้นตอนปฏิบัติการทดสอบ

ทีเอ็มเอ็มกล่าวไว้ว่าองค์กรควรวัดความน่าเชื่อถือของซอฟต์แวร์ คำถามและมาตรวัดคือ

คำถาม 3.3.2.5 วัดความน่าเชื่อถือของซอฟต์แวร์ได้อย่างไร?

มาตรวัด 3.3.2.5 - ระยะเวลาเฉลี่ยระหว่างความล้มเหลว (Mean Time Between Failures)

- ข้อบกพร่องที่พบตามเวลา (Defect arrival by time) [9]

3.3.3 เป้าหมายการประเมินคุณภาพซอฟต์แวร์

จุดประสงค์สำหรับเป้าหมายนี้คือ ให้ความสัมพันธ์ระหว่างคุณภาพของซอฟต์แวร์กับความเพียงพอในกระบวนการทดสอบ กำหนดคุณสมบัติด้านคุณภาพที่สามารถวัดได้ และกำหนดเป้าหมายคุณภาพเพื่อทดสอบผลิตภัณฑ์ที่เกิดจากการพัฒนาซอฟต์แวร์ คุณลักษณะทางคุณภาพได้แก่ ความถูกต้อง (Correctness) ความน่าเชื่อถือ ประสิทธิภาพ ความเที่ยงตรง (Integrity) ความสามารถในการใช้งาน ความสามารถในการบำรุงรักษา ความยืดหยุ่น (Flexibility) ความสามารถในการทดสอบ (Testability) ความสามารถในการย้ายไปทำงานกับระบบอื่น (Portability) ความสามารถในการนำกลับมาใช้ใหม่ (Reusability) และความสามารถในการทำงานร่วมกับระบบอื่น (Interoperability)

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) กำหนดคุณสมบัติด้านคุณภาพของผลิตภัณฑ์ซอฟต์แวร์
- 2) ใช้กระบวนการทดสอบเป็นเครื่องมือในการประเมินคุณภาพซอฟต์แวร์
- 3) ใช้ผลลัพธ์ในการปรับปรุงคุณภาพซอฟต์แวร์

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า เป็นดังต่อไปนี้

- ผู้จัดการจัดตั้งเป้าหมายคุณภาพขององค์กร จัดสรรทรัพยากรและเงินทุนเพื่อให้บรรลุเป้าหมายนั้นๆ ผู้จัดการโครงการกำหนดคุณสมบัติด้านคุณภาพสำหรับแต่ละโครงการ และต้องถูกระบุลงในโครงการและแผนงานการทดสอบว่าทำอย่างไรจึงจะบรรลุเป้าหมายคุณภาพที่ตั้งไว้
- ผู้พัฒนา/ผู้ทดสอบดำเนินการทดสอบสำหรับคุณสมบัติด้านคุณภาพตามแผนงานการทดสอบระบบ
- ผู้ใช้/ลูกค้ากำหนดคุณสมบัติด้านคุณภาพที่ต้องการและเข้าร่วมในการทดสอบความสามารถในการใช้งาน

องค์กรควรจะกำหนดคุณสมบัติด้านคุณภาพของผลิตภัณฑ์ซอฟต์แวร์และวัดคุณสมบัติที่กำหนดขึ้น คำถามและมาตรวัดคือ

คำถาม 3.3.3.1 คุณสมบัตินี้ด้านคุณภาพของผลิตภัณฑ์ซอฟต์แวร์จะวัดได้อย่างไร?

มาตรวัด 3.3.3.1 มาตรวัดเป็นไปตามตารางที่ 3.3 [10]

คุณสมบัติ	มาตรวัด	หมายเหตุ
ความถูกต้อง	จำนวนข้อบกพร่อง/จำนวนบรรทัดคำสั่ง	ข้อบกพร่องเกี่ยวกับความต้องการและมาตรฐาน
ความน่าเชื่อถือ	ดูมาตรวัด 3.3.2.5	
ประสิทธิภาพ	การใช้ประโยชน์จริง/การใช้ประโยชน์ที่จัดสรร	
ความเที่ยงตรง	จำนวนข้อบกพร่อง/จำนวนบรรทัดคำสั่ง	ข้อบกพร่องเกี่ยวกับความปลอดภัย
ความสามารถในการใช้งาน	ระยะเวลาที่ผู้ใช้ใช้ในการปฏิบัติงานเทียบกับเกณฑ์ (Benchmark)	
ความสามารถในการบำรุงรักษา	ระยะเวลาโดยเฉลี่ยที่ใช้ในการบำรุงรักษา	
ความยืดหยุ่น	ระยะเวลาโดยเฉลี่ยที่ใช้ในการเปลี่ยนแปลง	
ความสามารถในการทดสอบ	<u>ความพยายามที่ใช้ในการทดสอบ</u> ความพยายามที่ใช้ในการพัฒนา	

ตาราง 3.3 มาตรวัดสำหรับคุณสมบัติด้านคุณภาพของผลิตภัณฑ์ซอฟต์แวร์

คุณสมบัติ	มาตรวัด	หมายเหตุ
ความสามารถในการย้ายไปทำงานกับระบบอื่น	<u>ความพยายามที่ใช้ในการย้ายไปทำงานกับระบบอื่น</u> ความพยายามที่ใช้ในการพัฒนา	
ความสามารถในการนำกลับมาใช้ใหม่	<u>ความพยายามที่ใช้ในการปรับเปลี่ยน</u> ความพยายามที่ใช้ในการพัฒนา	
ความสามารถในการทำงานร่วมกับระบบอื่น	<u>ความพยายามที่ใช้ในการทำงานร่วมกับระบบอื่น</u> ความพยายามที่ใช้ในการพัฒนา	

ตาราง 3.3 มาตรวัดสำหรับคุณสมบัติด้านคุณภาพของผลิตภัณฑ์ซอฟต์แวร์ (ต่อ)

โดยสรุปมาตรวัดขั้นต่ำสำหรับระดับ 4 เป็นไปตามตารางที่ 3.4

เป้าหมาย	มุมมอง	มาตรวัด	หน่วย
3.3.1	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- จำนวนข้อบกพร่องที่ถูกขจัดโดยการทบทวน	ข้อบกพร่อง
3.3.2	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- จำนวนข้อบกพร่องในแต่ละระดับความรุนแรง - จำนวนข้อบกพร่องในแต่ละประเภทของข้อบกพร่อง - ผลผลิตของผู้ทดสอบ (ดูมาตรวัด 3.3.2.4)	ข้อบกพร่อง ข้อบกพร่อง
3.3.3	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- คูมาตรวัด 3.3.3.1	

ตาราง 3.4 มาตรวัดขั้นต่ำสำหรับที่เอ็มเอ็มในระดั 4

3.4 มาตรวัดในระดับ 5 การทำให้มีผลดีที่สุด การป้องกันข้อบกพร่อง และ การควบคุมคุณภาพ

ระดับ 5 ของทีเอ็มเอ็มประกอบไปด้วยเป้าหมายทั้งหมด 3 เป้าหมาย ซึ่งในแต่ละเป้าหมายมีรายละเอียดต่างๆ แสดงถึงกิจกรรมที่จะต้องกระทำเพื่อให้บรรลุเป้าหมาย โดยเป้าหมายในระดับ 5 มีดังต่อไปนี้

3.4.1 เป้าหมายการใช้ข้อมูลจากกระบวนการในการป้องกันข้อบกพร่อง

จุดประสงค์สำหรับเป้าหมายนี้คือ ระบุสาเหตุที่ทำให้เกิดข้อบกพร่องและหาวิธีการป้องกันไม่ให้ข้อบกพร่องนั้นเกิดขึ้นอีก กิจกรรมในการป้องกันข้อบกพร่องคือ การบันทึกและติดตามข้อบกพร่อง การวิเคราะห์หาสาเหตุของข้อบกพร่อง การทำแผนงานในการป้องกันข้อบกพร่อง การนำแผนงานไปปฏิบัติ และมีการติดตามผลของการนำแผนงานไปใช้

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) ระบุสาเหตุของข้อบกพร่อง
- 2) วิเคราะห์แนวโน้มซึ่งน่าจะเป็นสาเหตุราก (Root causes) ของข้อบกพร่อง
- 3) ระบุข้อบกพร่องที่น่าจะเกิดขึ้นอีก
- 4) ดำเนินวิธีการที่จะป้องกันไม่ให้ข้อบกพร่องนั้นเกิดขึ้นอีก
- 5) ติดตามความก้าวหน้าในการดำเนินการ

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้า มีดังต่อไปนี้

- ผู้จัดการให้นโยบายสนับสนุนและจัดสรรทรัพยากรและเงินทุน และมอบหมายทีมงานในการทำกิจกรรมการป้องกันข้อบกพร่อง ผู้จัดการป้องกันข้อบกพร่อง (Defect prevention manager) ทำแผนงาน ควบคุมและติดตามกิจกรรมการป้องกันข้อบกพร่อง และรายงานให้แก่ผู้บริหารระดับสูงทราบ
- ผู้พัฒนา/ผู้ทดสอบ ได้รับการมอบหมายให้อยู่ในทีมงานป้องกันข้อบกพร่อง ทำตามแผนงานที่วางไว้ กิจกรรมที่ทำในการป้องกันข้อบกพร่องต้องเริ่มทำตั้งแต่ช่วงเริ่มต้นของวงจรพัฒนาซอฟต์แวร์ ได้แก่ การบันทึกข้อบกพร่อง หาสาเหตุข้อบกพร่อง ดำเนินวิธีการแก้ไขข้อบกพร่องนั้น และทำรายงานให้แก่ผู้จัดการทราบ
- ผู้ใช้/ลูกค้า รายงานปัญหาหรือข้อบกพร่องที่พบและผลกระทบที่ได้รับหลังจากการใช้งานซอฟต์แวร์

องค์กรควรติดตามสถานะของกิจกรรมที่ดำเนินการป้องกันข้อบกพร่อง คำถามและมาตรวัดคือ

คำถาม 3.4.1.1 กิจกรรมที่ทำในการป้องกันข้อบกพร่องเป็นอย่างไรเมื่อเทียบกับแผนงาน?

มาตรวัด 3.4.1.1 - ความพยายามในการหาสาเหตุรากของข้อบกพร่อง

- ความแตกต่างของวันเริ่มกิจกรรม = วันเริ่มกิจกรรมจริง - ตามแผนงาน
- ความแตกต่างของวันสิ้นสุดกิจกรรม = วันสิ้นสุดกิจกรรมจริง - ตามแผนงาน
- %ความแตกต่างของค่าใช้จ่ายในรูปแบบตัวเงินหรือความพยายาม

$$= \frac{\text{ค่าใช้จ่ายจริง} - \text{ตามแผนงาน}}{\text{ค่าใช้จ่ายตามแผนงาน}} * 100$$

- จำนวนข้อเสนอในการป้องกันข้อบกพร่องที่นำเสนอ (Proposed) จำนวนข้อเสนอในการป้องกันข้อบกพร่องที่เปิด (Open) และจำนวนข้อเสนอในการป้องกันข้อบกพร่องที่เสร็จสิ้น (Closed)

องค์กรควรจะติดตามผลลัพธ์ของกิจกรรมที่ทำการป้องกันข้อบกพร่องคำถามและมาตรวัดคือ

คำถาม 3.4.1.2 วัดประสิทธิผลของกิจกรรมที่ทำการป้องกันข้อบกพร่องได้อย่างไร?

มาตรวัด 3.4.1.2 เปรียบเทียบมาตรวัดก่อนทำและหลังทำกิจกรรมดังต่อไปนี้

- ความหนาแน่นของข้อบกพร่อง
- ประสิทธิภาพของการจัดข้อบกพร่อง
- การกระจายความถี่ (Frequency distribution) ของข้อบกพร่องตามสาเหตุต่างๆ

หมายเหตุ ในงานวิจัยนี้ ไม่ได้ให้สาเหตุต่างๆ ที่ทำให้เกิดข้อบกพร่อง องค์กรจะต้องกำหนดสาเหตุต่างๆ ขึ้นเอง

3.4.2 เป้าหมายการควบคุมคุณภาพ

จุดประสงค์ของเป้าหมายนี้คือการใช้วิธีการทางสถิติในการควบคุมกระบวนการทดสอบ องค์กรควรที่จะกำหนดเกณฑ์ทางสถิติที่ใช้ในการควบคุมกระบวนการทดสอบ คือค่าขีดจำกัดล่าง (LCL: Lower Control Limit) และค่าขีดจำกัดบน (UCL: Upper Control Limit) ของตัวแปรที่ใช้ในการควบคุม เพื่อนำไปสร้างแผนผังควบคุม (Control chart) คำถามและมาตรวัดคือ

คำถาม 3.4.2.1 กำหนดค่าขีดจำกัดล่างและค่าขีดจำกัดบนของมาตรวัดอะไรบ้าง?

มาตรวัด 3.4.2.1 กำหนดค่าขีดจำกัดล่างและค่าขีดจำกัดบนของมาตรวัดดังต่อไปนี้

- ความหนาแน่นของข้อบกพร่อง
- อัตราการทบทวน
- อัตราความพยายามในการทดสอบ
- ความครอบคลุมการทดสอบ ได้แก่ความครอบคลุมข้อความสั่ง ความครอบคลุมกิ่ง และความครอบคลุมทางเดินพื้นฐาน

3.4.3 เป้าหมายการทำให้กระบวนการทดสอบดีที่สุดใน

จุดประสงค์ของเป้าหมายนี้คือการปรับปรุงกระบวนการทดสอบอย่างต่อเนื่องทั้งในระดับโครงการและองค์กร เพื่อปรับปรุงคุณภาพของซอฟต์แวร์และผลิตภัณฑ์ซอฟต์แวร์ให้เสร็จตรงเวลาและอยู่ภายในงบประมาณ

เป้าหมายนี้ประกอบด้วยเป้าหมายย่อยดังต่อไปนี้

- 1) ระบุจุดปฏิบัติงานในการทดสอบ (Testing practices) ที่จำเป็นต้องปรับปรุง
- 2) ดำเนินการปรับปรุง
- 3) ติดตามความก้าวหน้าในการปรับปรุง
- 4) ประเมินและนำเครื่องมือและเทคโนโลยีในการทดสอบมาใช้
- 5) ระบุวิธีการในการกำหนดว่าเมื่อใดจะหยุดการทดสอบ และเมื่อใดจะสามารถปล่อยซอฟต์แวร์ไปใช้งาน

กิจกรรมที่ต้องทำในมุมมองของผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ และผู้ใช้/ลูกค้ามีดังต่อไปนี้

นี้

- ผู้จัดการจัดตั้งแผนงานการปรับปรุงกระบวนการทดสอบ จัดสรรทรัพยากรและเงินทุน มอบหมายความรับผิดชอบให้แก่บุคลากร ร่วมในการประเมินและตัดสินใจในการนำเครื่องมือและเทคโนโลยีในการทดสอบมาใช้งาน และร่วมระบุวิธีการในการกำหนดว่าเมื่อใดจะหยุดการทดสอบ
- ผู้พัฒนา/ผู้ทดสอบร่วมในการพัฒนาแผนงานการปรับปรุงกระบวนการทดสอบ ระบุจุดปฏิบัติงานในการทดสอบที่จำเป็นต้องปรับปรุง แนะนำและใช้เครื่องมือและเทคโนโลยีในการทดสอบ และระบุเกณฑ์ที่ใช้ในการหยุดการทดสอบ
- ความพอใจและความต้องการของผู้ใช้/ลูกค้าเป็นแหล่งข้อมูลที่นำมาใช้ในการระบุจุดปฏิบัติงานในการทดสอบที่ต้องปรับปรุง

องค์กรควรจะติดตามสถานะของกิจกรรมในการปรับปรุงกระบวนการทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.4.3.1 กิจกรรมในการปรับปรุงกระบวนการทดสอบเป็นไปตามแผนหรือไม่?

มาตรวัด 3.4.3.1 - ความแตกต่างของวันเริ่มกิจกรรม = วันเริ่มกิจกรรมจริง - ตามแผนงาน

- ความแตกต่างของวันสิ้นสุดกิจกรรม = วันสิ้นสุดกิจกรรมจริง - ตามแผนงาน

- %ความแตกต่างของจำนวนบุคลากร = $\frac{\text{จำนวนบุคลากรจริง} - \text{ตามแผนงาน}}{\text{จำนวนบุคลากรตามแผนงาน}} * 100$

จำนวนบุคลากรตามแผนงาน

- ความพยายามที่ใช้ไปในการการประเมิน และตัดสินใจในการนำเครื่องมือและเทคโนโลยีการทดสอบมาใช้งาน
- ความพยายามที่ใช้ไปในการกำหนดว่าเมื่อใดจะหยุดการทดสอบ

องค์กรควรจะกำหนดเกณฑ์ที่ใช้ในการหยุดการทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.4.3.2 อะไรเป็นเกณฑ์ที่ใช้ในการหยุดการทดสอบ?

มาตรวัด 3.4.3.2 * เปรียบเทียบอัตราความล้มเหลวจริง (Actual failure rate) กับที่ได้ตั้งเป้าหมาย (Failure rate goal)

องค์กรควรจะกำหนดเกณฑ์ที่ใช้ในการพิจารณาว่าจะนำเครื่องมือและเทคโนโลยีในการทดสอบมาใช้งานหรือไม่ คำถามและมาตรวัดคือ

คำถาม 3.4.3.3 องค์กรควรจะนำเครื่องมือและเทคโนโลยีในการทดสอบมาใช้งานหรือไม่?

มาตรวัด 3.4.3.3 วิเคราะห์ค่าใช้จ่ายกับผลประโยชน์ (Cost-benefit analysis) ในการนำเครื่องมือและเทคโนโลยีในการทดสอบมาใช้งาน

องค์กรควรจะติดตามผลลัพธ์ของกิจกรรมที่ทำการปรับปรุงกระบวนการทดสอบ คำถามและมาตรวัดคือ

คำถาม 3.4.3.4 วัดประสิทธิผลของกิจกรรมที่ทำการปรับปรุงกระบวนการทดสอบได้อย่างไร?

มาตรวัด 3.4.3.4 ติดตามมาตรวัดซึ่งเป็นดัชนีที่ชี้ให้เห็นถึงคุณภาพของกระบวนการทดสอบดังต่อไปนี้

- ความหนาแน่นของข้อบกพร่อง
- ประสิทธิภาพของการจัดข้อบกพร่อง
- ค่าใช้จ่ายในการทดสอบ
- ผลผลิตของผู้ทดสอบ

ผลิตภัณฑ์ของมหาวิทยาลัย
จุฬาลงกรณ์มหาวิทยาลัย

โดยสรุป มาตรฐานขั้นต่ำสำหรับระดับ 5 เป็นไปตามตารางที่ 3.5

เป้าหมาย	มุมมอง	มาตรฐาน	หน่วย
3.4.1	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- ความพยายามในการหาสาเหตุรากของข้อบกพร่อง	คน. ชั่วโมง
3.4.2	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ	- ค่าขีดจำกัดล่างและค่าขีดจำกัดบนของความหนาแน่นของข้อบกพร่อง อัตราการทบทวน อัตราความพยายามในการทดสอบ และ ความครอบคลุมการทดสอบ ได้แก่ ความครอบคลุมข้อความสั่ง ความครอบคลุมกิ่ง และความครอบคลุมทางเคินพื้นฐาน	
3.4.3	ผู้จัดการ ผู้พัฒนา/ผู้ทดสอบ ผู้ใช้/ลูกค้า	- ความพยายามที่ใช้ไปในการประเมินและตัดสินใจในการนำเครื่องมือและเทคโนโลยีการทดสอบมาใช้งาน - ความพยายามที่ใช้ไปในการกำหนดว่าเมื่อใดจะหยุดการทดสอบ - อัตราความล้มเหลวที่ได้ตั้งเป้าหมาย - อัตราความล้มเหลวจริง	คน. ชั่วโมง คน. ชั่วโมง ความล้มเหลว/ ระยะเวลา

ตาราง 3.5 มาตรฐานขั้นต่ำสำหรับที่เอ็มเอ็มในระดับ 5

บทที่ 4

การออกแบบและพัฒนาเครื่องมือในการเก็บมาตรวัด

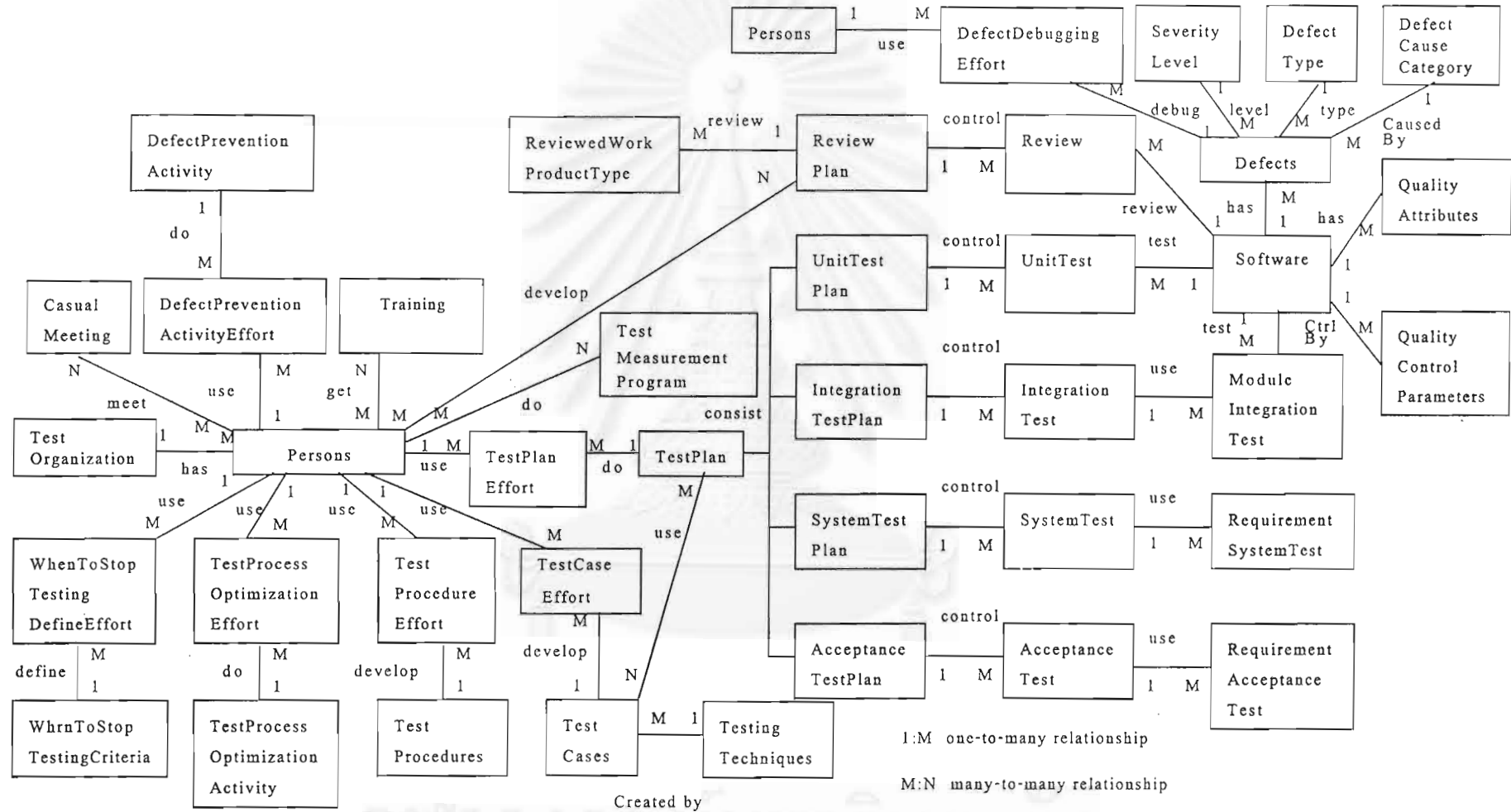
ในการออกแบบและพัฒนาเครื่องมือในการเก็บมาตรวัดของทีเอ็มเอ็ม ประกอบไปด้วย

1. การออกแบบฐานข้อมูลเพื่อเก็บมาตรวัดในแต่ละระดับของทีเอ็มเอ็ม
2. การออกแบบอินพุต สำหรับผู้ใช้ป้อนข้อมูลมาตรวัดเข้าฐานข้อมูล
3. การออกแบบเอาต์พุต ซึ่งเป็นรายงานต่างๆ สำหรับใช้ในการติดตามและวิเคราะห์กระบวนการทดสอบ
4. การพัฒนาเครื่องมือ เครื่องมือต้นแบบที่พัฒนาขึ้นเป็นแบบผู้ใช้คนเดียว (Single-user) โดยสภาพแวดล้อมในการพัฒนาคือ
 - เครื่องไมโครคอมพิวเตอร์ เพนเทียม 133 เมกะเฮิร์ตซ์ หน่วยความจำ 16 เมกะไบต์
 - ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 95
 - เครื่องมือได้รับการพัฒนาบนไมโครซอฟต์แอ็คเซส 97

4.1 การออกแบบฐานข้อมูลเพื่อเก็บมาตรวัด โดยการออกแบบจะใช้หลักการออกแบบฐานข้อมูลเชิงสัมพันธ์ การออกแบบตารางจะคำนึงถึงทีเอ็มเอ็มทั้ง 4 ระดับ (ไม่รวมระดับ 1) คือทีเอ็มเอ็มระดับสูงขึ้นไปจะเก็บข้อมูลมาตรวัดเพิ่มขึ้น ตารางใดตารางหนึ่งอาจจะเก็บข้อมูลมาตรวัดทั้ง 4 ระดับ เช่น ตารางข้อบกพร่อง ระดับ 2 และ 3 จะยังไม่เก็บประเภทและความรุนแรงข้อบกพร่อง แต่ระดับ 4 จะเก็บข้อมูลดังกล่าว การออกแบบตารางข้อบกพร่องจะมีเขตข้อมูลประเภทและความรุนแรงข้อบกพร่องด้วย ถึงแม้จะยังไม่ใช้ใน ระดับ 2 กับ 3

รูปที่ 4.1 แสดงให้เห็นถึงแผนผังความสัมพันธ์ของตารางต่างๆ ในฐานข้อมูล เขตข้อมูลของตารางจะได้มาจากมาตรวัดที่ได้กำหนดขึ้นในบทที่ 3 รายละเอียดของตารางต่างๆ คือ เขตข้อมูล ชนิดและความยาวเขตข้อมูล ดูได้จากภาคผนวก ข วัตถุประสงค์ของตารางต่างๆ เป็นดังต่อไปนี้

- ตารางการทดสอบเพื่อยอมรับ (Acceptance Test) เพื่อเก็บข้อมูลในขั้นตอนการทดสอบเพื่อยอมรับ
- ตารางแผนงานการทดสอบเพื่อยอมรับ (Acceptance Test Plan) เพื่อเก็บข้อมูลความต้องการที่ทดสอบในแผนงานการทดสอบเพื่อยอมรับ
- ตารางการประชุมหาสาเหตุรากของข้อบกพร่อง (Casual Meeting) เพื่อเก็บรายละเอียดในการประชุมหาสาเหตุรากของข้อบกพร่อง



รูปที่ 4.1 แสดงความสัมพันธ์ของตารางต่างๆ ในฐานข้อมูลสำหรับเก็บข้อมูลมาตรวัดของทีเอ็มเอ็ม

- ตารางประเภทของสาเหตุของข้อบกพร่อง (DefectCauseCategory)
เพื่อเก็บประเภทของสาเหตุของข้อบกพร่อง
- ตารางความพยายามในการแก้ไขข้อบกพร่อง (DefectDebuggingEffort)
เพื่อเก็บข้อมูลความพยายามในการแก้ไขข้อบกพร่อง
- ตารางกิจกรรมการป้องกันข้อบกพร่อง (DefectPreventionActivity)
เพื่อเก็บรายละเอียดกิจกรรมการป้องกันข้อบกพร่อง
- ตารางความพยายามในการทำกิจกรรมการป้องกันข้อบกพร่อง
(DefectPreventionActivityEffort) เพื่อเก็บข้อมูลความพยายามในการทำกิจกรรมการป้องกันข้อบกพร่อง
- ตารางข้อบกพร่อง (Defects)
เพื่อเก็บรายละเอียดข้อบกพร่อง
- ตารางความรุนแรงของข้อบกพร่อง (DefectSeverity)
เพื่อเก็บระดับความรุนแรงของข้อบกพร่อง
- ตารางประเภทของข้อบกพร่อง (DefectType)
เพื่อเก็บประเภทของข้อบกพร่อง
- ตารางการทดสอบเบ็ดเสร็จ (IntegrationTest)
เพื่อเก็บข้อมูลในขั้นตอนการทดสอบเบ็ดเสร็จ
- ตารางแผนงานการทดสอบเบ็ดเสร็จ (IntegrationTestPlan)
เพื่อเก็บรายละเอียดของโมดูลต่างๆที่นำมารวมกันในแผนงานการทดสอบเบ็ดเสร็จ
- ตารางการทดสอบโมดูลที่นำมารวมกัน (ModuleIntegrationTest)
เพื่อเก็บผลลัพธ์ของการทดสอบโมดูลต่างๆที่นำมารวมกัน
- ตารางบุคลากร (Persons)
เพื่อเก็บข้อมูลของบุคลากรที่เกี่ยวข้องกับกระบวนการทดสอบ
- ตารางประเภทบุคลากร (PersonType)
เพื่อเก็บข้อมูลประเภทบุคลากรที่เกี่ยวข้องกับกระบวนการทดสอบ
- ตารางพารามิเตอร์ที่ใช้ในการควบคุมกระบวนการทดสอบ
(QualityControlParameters) เพื่อเก็บค่าของพารามิเตอร์ที่ใช้ในการควบคุมกระบวนการทดสอบ

- ตารางการทดสอบความต้องการในขั้นก่อนการทดสอบเพื่อยอมรับ
(RequirementAcceptanceTest) เพื่อเก็บผลลัพธ์ของการทดสอบความต้องการใน
ขั้นก่อนการทดสอบเพื่อยอมรับ
- ตารางการทดสอบความต้องการในขั้นก่อนการทดสอบระบบ
(RequirementSystemTest) เพื่อเก็บผลลัพธ์ของการทดสอบความต้องการในขั้น
ก่อนการทดสอบระบบ
- ตารางการทบทวน (Review)
เพื่อเก็บรายละเอียดในการทบทวน
- ตารางชนิดของผลผลิตที่นำมาทบทวน (ReviewedWorkProductType)
เพื่อเก็บชนิดของผลผลิตที่นำมาทบทวน
- ตารางแผนงานการทบทวน (ReviewPlan)
เพื่อเก็บรายละเอียดของแผนงานการทบทวน
- ตารางซอฟต์แวร์ (Software)
เพื่อเก็บรายละเอียดของซอฟต์แวร์ที่นำมาทดสอบ
- ตารางคุณสมบัติของซอฟต์แวร์ (SoftwareQualityAttributes)
เพื่อเก็บค่ามาตรวัดที่เกี่ยวข้องกับคุณสมบัติของซอฟต์แวร์
- ตารางการทดสอบระบบ (SystemTest)
เพื่อเก็บข้อมูลในขั้นก่อนการทดสอบระบบ
- ตารางแผนงานการทดสอบระบบ (SystemTestPlan)
เพื่อเก็บข้อมูลความต้องการที่ทดสอบในแผนงานการทดสอบระบบ
- ตารางความพยายามในการพัฒนากรณีทดสอบ (TestCaseEffort)
เพื่อเก็บข้อมูลความพยายามในการพัฒนากรณีทดสอบ
- ตารางแผนงานกรณีทดสอบ (TestCasePlan)
เพื่อเก็บข้อมูลแผนงานการนำกรณีทดสอบมาทดสอบในกิจกรรมการทดสอบ
- ตารางกรณีทดสอบ (TestCases)
เพื่อเก็บรายละเอียดกรณีทดสอบ
- ตารางสถานะกรณีทดสอบ (TestCaseStatus)
เพื่อเก็บผลลัพธ์ของการนำกรณีทดสอบมาทำการทดสอบ
- ตารางโปรแกรมการวัดกระบวนการทดสอบ (TestMeasurementProgram)
เพื่อเก็บรายละเอียดโปรแกรมการวัดกระบวนการทดสอบขององค์กร
- ตารางเทคนิคในการทดสอบ (TestingTechniques)
เพื่อเก็บเทคนิคในการทดสอบขององค์กร

- ตารางหน่วยงานการทดสอบขององค์กร (TestOrganization)
เพื่อเก็บรายละเอียดค่าใช้จ่ายของหน่วยงานการทดสอบขององค์กร
- ตารางแผนงานการทดสอบ (TestPlan)
เพื่อเก็บรายละเอียดแผนงานการทดสอบ
- ตารางความพยายามในการพัฒนาแผนงานการทดสอบ
(TestPlanDevelopmentEffort) เพื่อเก็บข้อมูลความพยายามในการพัฒนาแผนงาน
การทดสอบ
- ตารางกิจกรรมในการทำให้กระบวนการทดสอบดีที่สุด
(TestProcessOptimizationActivity) เพื่อเก็บรายละเอียดการประเมินเทคโนโลยี
หรือเครื่องมือการทดสอบในการปรับปรุงกระบวนการทดสอบ
- ตารางความพยายามของกิจกรรมในการทำให้กระบวนการทดสอบดีที่สุด
(TestProcessOptimizationEffort) เพื่อเก็บความพยายามในการประเมินหรือใช้
เทคโนโลยีและเครื่องมือการทดสอบในการปรับปรุงกระบวนการทดสอบ
- ตารางความพยายามในการพัฒนาขั้นตอนปฏิบัติการทดสอบ
(TestProcedureEffort) เพื่อเก็บความพยายามในการพัฒนาขั้นตอนปฏิบัติการ
ทดสอบ
- ตารางขั้นตอนปฏิบัติการทดสอบ (TestProcedures)
เพื่อเก็บรายละเอียดขั้นตอนปฏิบัติการทดสอบ
- ตารางชนิดของเครื่องมือการทดสอบ (TestToolType)
เพื่อเก็บชนิดของเครื่องมือการทดสอบ
- ตารางการฝึกอบรม (Training)
เพื่อเก็บรายละเอียดการฝึกอบรมด้านการทดสอบ
- ตารางการทดสอบโมดูล (UnitTest)
เพื่อเก็บข้อมูลในขั้นตอนการทดสอบ โมดูล
- ตารางแผนงานการทดสอบโมดูล (UnitTestPlan)
เพื่อเก็บข้อมูลแผนงานการทดสอบโมดูล
- ตารางความพยายามในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ
(WhenToStopTestingDefineEffort) เพื่อเก็บข้อมูลความพยายามในการกำหนดว่า
เมื่อใดควรจะหยุดการทดสอบ
- ตารางเกณฑ์ที่ใช้ในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ
(WhenToStopTestingCriteria) เพื่อเก็บเกณฑ์ที่ใช้ในการกำหนดว่าเมื่อใดควรจะ
หยุดการทดสอบ

4.2 การออกแบบอินพุตและเอาต์พุต ผู้ใช้จะป้อนข้อมูลมาตรฐานในแต่ละระดับของที่เฝ้า
เฝ้าผ่านหน้าจอ โดยมีเมนูหลักดังต่อไปนี้

4.2.1 เมนูจัดเตรียม (Setup) ประกอบด้วยเมนูย่อยคือ

4.2.1.1 รายละเอียดซอฟต์แวร์ ผู้ใช้จะต้องป้อนข้อมูลรายละเอียดของโครงการซอฟต์แวร์ก่อนที่จะทำการป้อนข้อมูลในเมนูอื่นๆ ตัวอย่างหน้าจอเป็นดังรูปที่ 4.2 ผู้ใช้ต้องป้อนหมายเลขซอฟต์แวร์ ชื่อซอฟต์แวร์ รุ่นของซอฟต์แวร์ จำนวนหน้าของเอกสารความต้องการ จำนวนความต้องการ จำนวนหน้าของเอกสารออกแบบ จำนวนโมดูล จำนวนบรรทัดคำสั่งของซอฟต์แวร์ จำนวนหน้าของเอกสารการทดสอบโมดูล จำนวนหน้าของเอกสารการทดสอบเบ็ดเสร็จ จำนวนหน้าของเอกสารการทดสอบระบบ จำนวนหน้าของเอกสารการทดสอบเพื่อยอมรับ และหน่วยของความพยายามที่ใช้ในโครงการซอฟต์แวร์

Software Detail	
SoftwareID	1
Name	Library Automation System
Version	1.0
Total Pages of Requirement Document	200
Total Number of Requirements	200
Total Pages of Design Document	400
Total Modules	200
Total Lines Of Code	100
KLDC	KLDC
Total Pages of Unit Test	300
Total Pages of Integration Test	150
Total Pages of System Test	100
Total Pages of Acceptance Test	100
Effort Unit	Man-Hours

Buttons: Add, Edit, Save, Delete, Cancel, Exit

รูปที่ 4.2 ตัวอย่างหน้าจอรายละเอียดซอฟต์แวร์

4.2.1.2 รายละเอียดบุคลากร สำหรับป้อนข้อมูลชื่อ นามสกุลและประเภทบทบาทของบุคลากรในกระบวนการทดสอบคือ ผู้จัดการ ผู้ทดสอบ ผู้พัฒนา และผู้ใช้/ลูกค้า ดังตัวอย่างหน้าจอในรูปที่ 4.3

Person Detail	
PersonID	0002
Name	Pongpan
Surname	Bae
Person Type	Tester

Buttons: Add, Save, Delete, Find, Undo, Exit

Record: 2 of 6

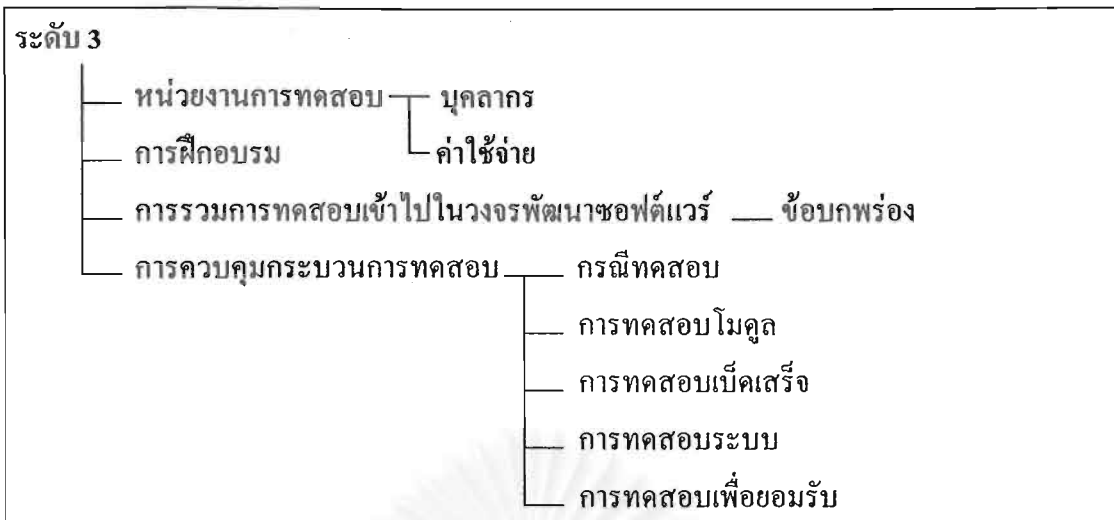
รูปที่ 4.3 ตัวอย่างหน้าจอบุคลากรในกระบวนการทดสอบ

สำหรับเมนูในแต่ละระดับของทีเอ็มเอ็มจากระดับ 2-5 เป็นไปตามรูปที่ 4.4 - 4.7 และ
 เมนุรายงานเป็นไปตามรูปที่ 4.8

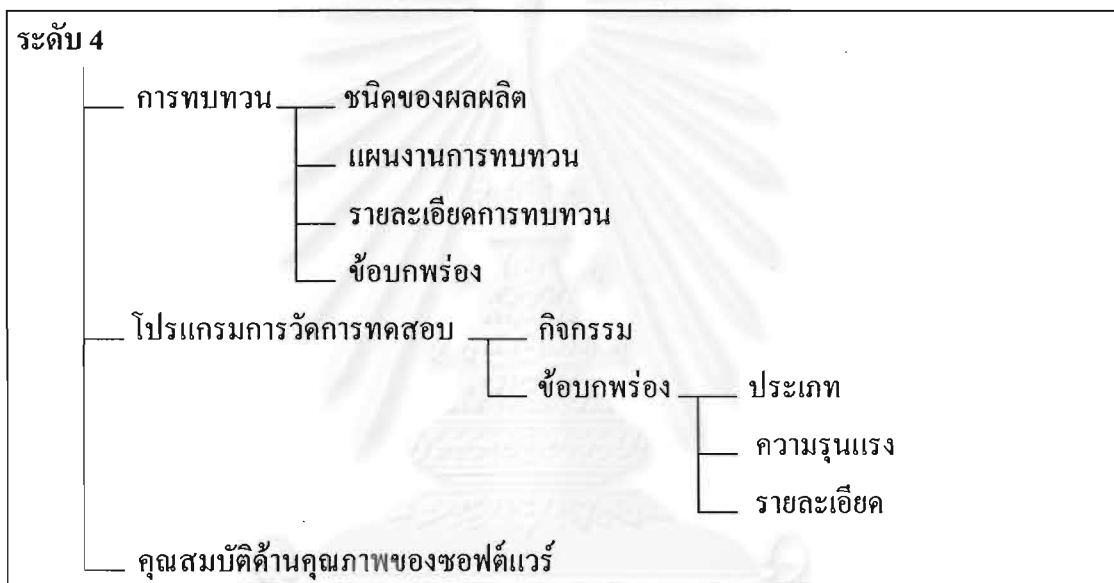


รูปที่ 4.4 โครงสร้างเมนูในระดับ 2

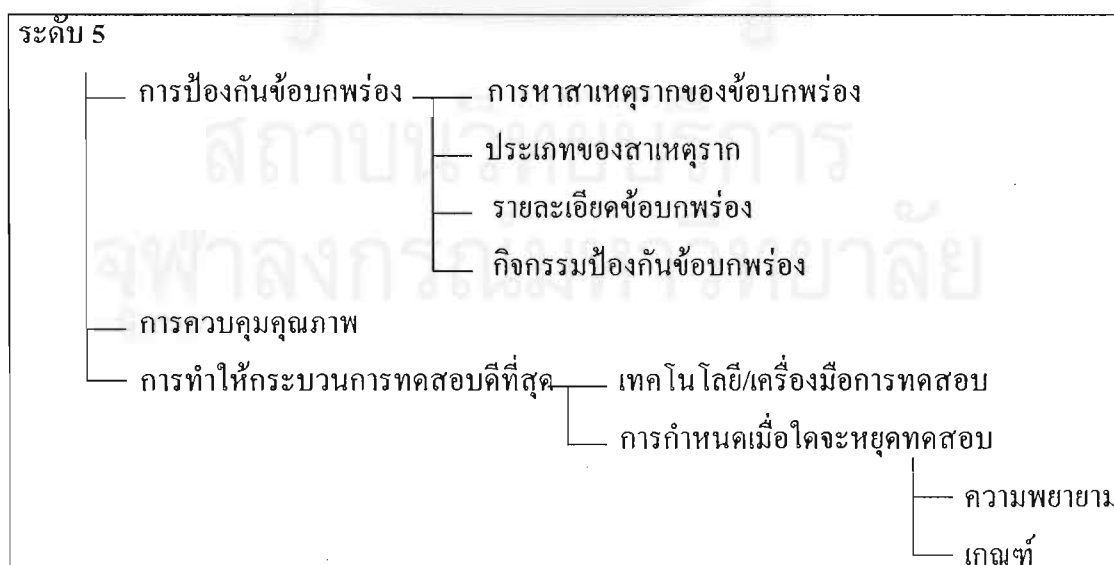
สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย



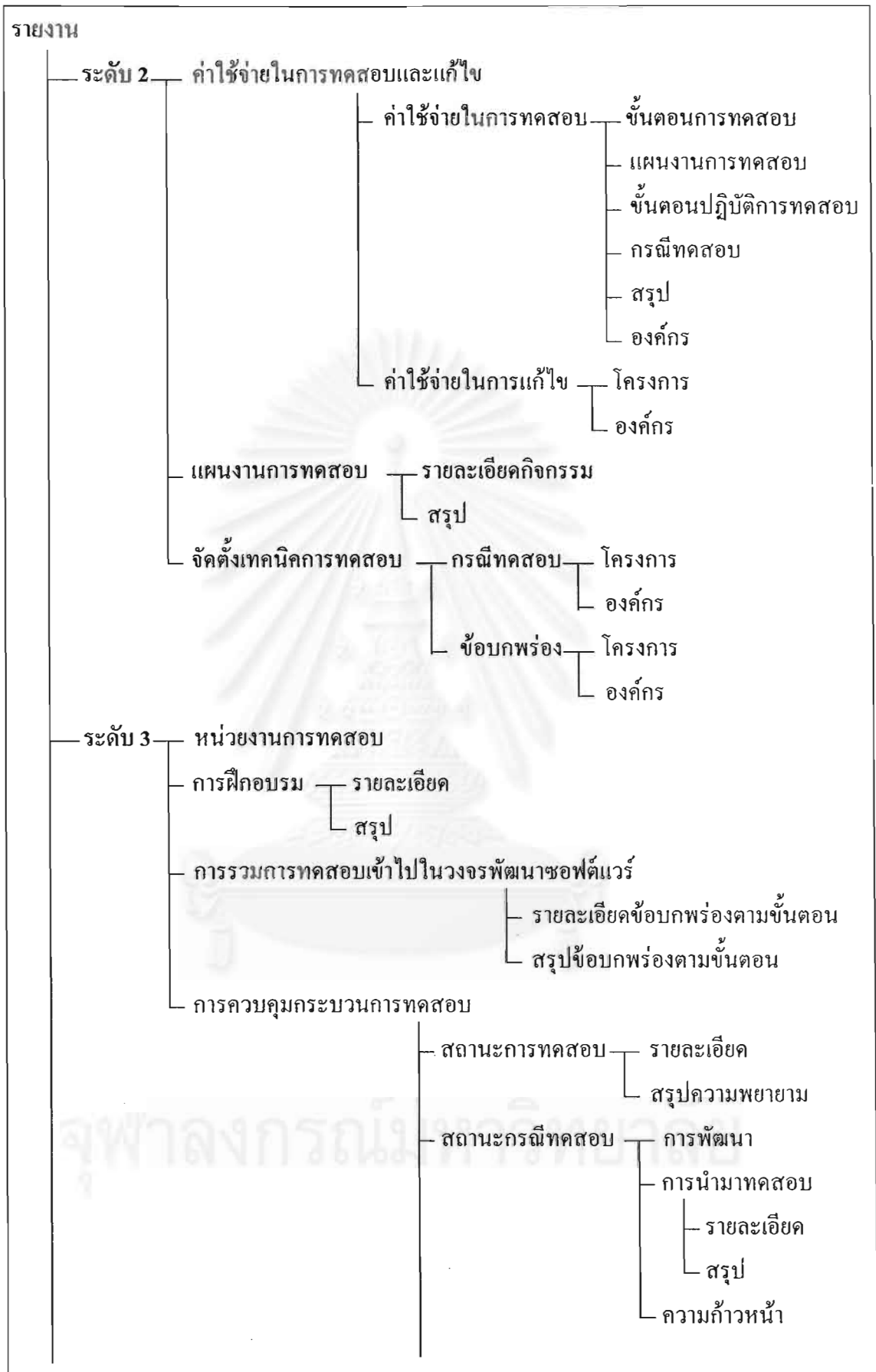
รูปที่ 4.5 โครงสร้างเมนูในระดับ 3



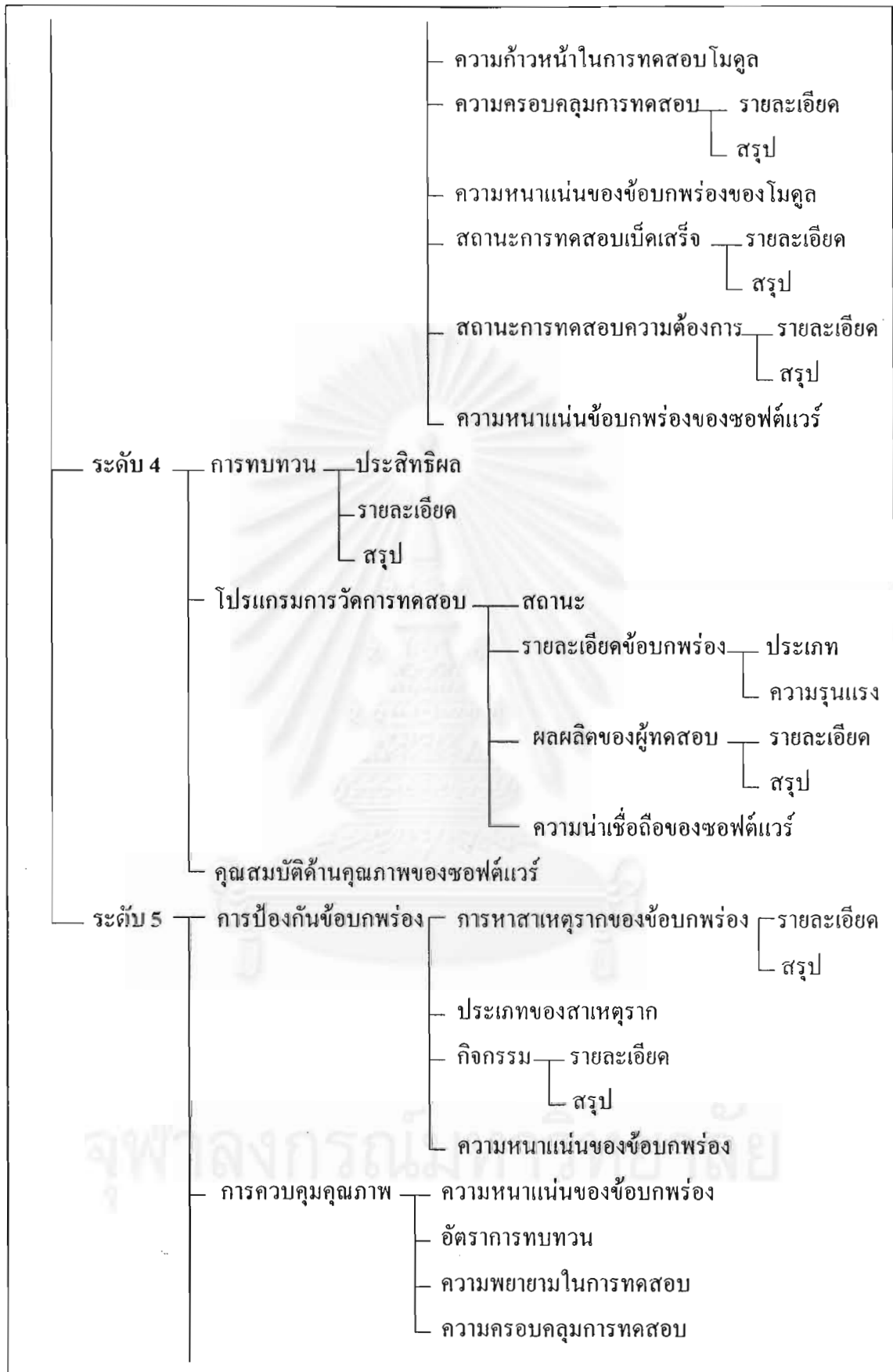
รูปที่ 4.6 โครงสร้างเมนูในระดับ 4



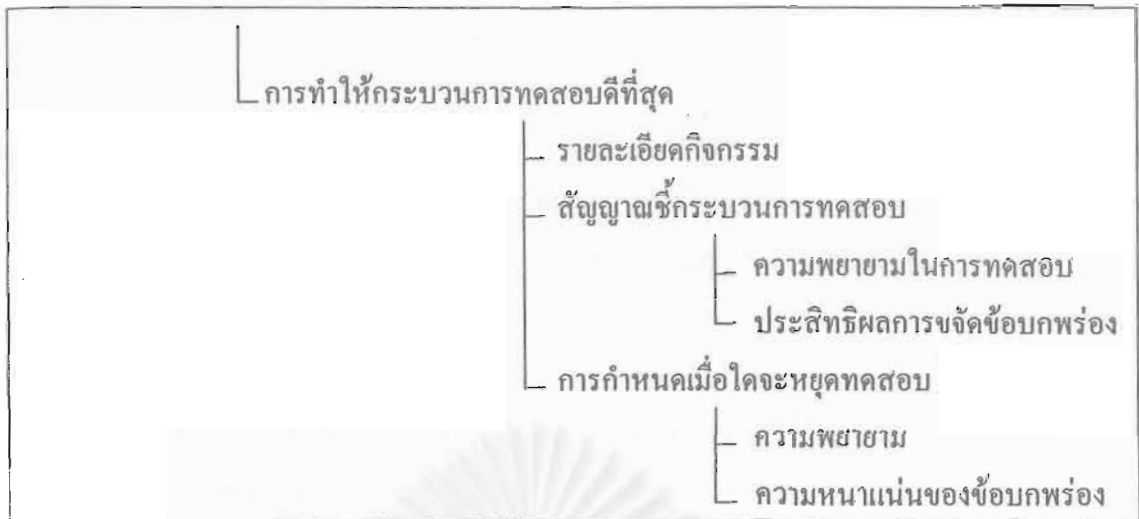
รูปที่ 4.7 โครงสร้างเมนูในระดับ 5



รูปที่ 4.8 โครงสร้างเมนูรายงาน



รูปที่ 4.8 โครงสร้างเมนูรายงาน (ต่อ)



รูปที่ 4.8 โครงสร้างเมนูรายงาน (ต่อ)

ต่อไปจะกล่าวถึงการออกแบบหน้าจอและรายงานรวมถึงตัวอย่างในแต่ละระดับของทีเอ็มเอ็ม

4.2.2 หน้าจอและรายงานของมาตรวัดในระดับ 2

1) พัฒนาเป้าหมายการทดสอบและการแก้ไข

- ค่าใช้จ่ายในการทดสอบ จากมาตรวัด 3.1.1.1 ในบทที่ 3 ต้องเก็บข้อมูลค่าใช้จ่ายในรูปของความพยายามในการดำเนินการทดสอบในแต่ละขั้นตอนของการทดสอบ ซึ่งได้แก่ การทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ ดังนั้นหน้าจอที่ออกแบบต้องให้ผู้ใช้ป้อนข้อมูลความพยายามเข้าไป ดังตัวอย่างหน้าจอรูปที่ 4.9 เป็นหน้าจอในการป้อนข้อมูลในขั้นตอนการทดสอบโมดูล ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ (Software ID) แล้วเลือกหมายเลขกิจกรรมการทดสอบตามแผนงานการทดสอบ (Test Activity ID) ทำการป้อนข้อมูล จำนวนบรรทัดคำสั่งของโมดูลที่ทดสอบ ความพยายามในการทดสอบ ผลลัพธ์ของการทดสอบ และจำนวนข้อบกพร่อง (ถ้ามี)

The screenshot shows a form window titled "Unit Test (TMM Level 2 Goal 1)". The form has the following fields and controls:

- Software ID: 1
- Name: Library Automation System
- Version: 1.0
- Effort unit: Men.Hours
- Lines Of Code unit: KLOC
- Test Activity ID: 1
- Description: Test the module of calculating returned date
- Module ID: 1
- Lines Of Code: 1
- Effort: 4
- Result: Fail
- Number of defects: 2
- Note: (Empty text area)
- Buttons: Add, Edit, Save, Delete, Previous, Next, Cancel, Exit

รูปที่ 4.9 ตัวอย่างหน้าจอการทดสอบโมดูลในระดับ 2

สำหรับหน้าจอการป้อนข้อมูลความพยายามในการทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ มีลักษณะเช่นเดียวกับหน้าจอของการทดสอบโมดูล

นอกจากค่าใช้จ่ายในการดำเนินการทดสอบแล้ว ค่าใช้จ่ายในการทดสอบยังประกอบด้วยค่าใช้จ่ายในการพัฒนากรณีทดสอบ และขั้นตอนปฏิบัติการทดสอบ ดังนั้นหน้าจอที่ออกแบบต้องให้ผู้ป้อนข้อมูลความพยายามเข้าไป ดังตัวอย่างหน้าจอรูปที่ 4.10 เป็นตัวอย่างหน้าจอที่ใช้ในการป้อนข้อมูลการพัฒนากรณีทดสอบ ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขกรณีทดสอบที่ต้องการเพิ่มหรือแก้ไข รายละเอียดของกรณีทดสอบ เทคนิคที่ใช้ในการพัฒนา (ข้อมูลสำหรับเป้าหมายจัดตั้งเทคนิคและวิธีการพื้นฐานในการทดสอบ ซึ่งจะกล่าวต่อไป) และความพยายามของบุคลากรที่ใช้

The screenshot shows a window titled 'Test Case Detail Level 2' with the following fields and data:

- SoftwareID: 1, Name: Library Automation System
- Version: 1.0, Effort unit: Man.Hours, Lines Of Code unit: KLOC
- Test Case ID: 1, Description: For test borrowed and returned data
- Testing technique: Black Box technique, Development Effort: 20

Below these fields is a table for 'Test Case Development Effort':

PersonID	Name	Surname	Date	Development Effort
0001	Anuchit	Tang	13-Sep-99	10
0002	Pongpan	Bae	13-Sep-99	10
*			27-Dec-99	

Buttons at the bottom include Add, Edit, Save, Delete, Cancel, and Exit.

รูปที่ 4.10 ตัวอย่างหน้าจอรายละเอียดกรณีทดสอบในระดับ 2

สำหรับหน้าจอสำหรับการป้อนข้อมูลค่าใช้จ่ายในรูปของความพยายามในการพัฒนาขั้นตอนปฏิบัติการทดสอบเป็นดังรูปที่ 4.11

The screenshot shows a window titled 'Test Procedures: Form' with the following fields and data:

- Software ID: 1, Name: Library Automation System
- Version: 1.0, Effort unit: Man.Hours, Lines Of Code unit: KLOC
- Test Procedure ID: 1, Description: Test procedures for adding members
- Development Effort: 35

Below these fields is a table for 'Test Procedure Effort':

PersonID	Name	Surname	Date	Development Effort
0001	Anuchit	Tang	20-Sep-99	15
0002	Pongpan	Bae	20-Sep-99	20

Buttons at the bottom include Add, Edit, Save, Delete, Cancel, and Exit.

รูปที่ 4.11 ตัวอย่างหน้าจอขั้นตอนปฏิบัติการทดสอบในระดับ 2

- ค่าใช้จ่ายในการแก้ไขข้อบกพร่อง จากมาตรวัด 3.1.1.2 ในบทที่ 3 ต้องเก็บข้อมูลค่าใช้จ่ายในรูปของความพยายามในการแก้ไข ดังนั้นหน้าจอที่ออกแบบต้องให้ผู้ใช้ป้อนข้อมูลความพยายามเข้าไป ดังตัวอย่างหน้าจอรูปที่ 4.12 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ แล้วป้อนหมายเลขข้อบกพร่อง รายละเอียดข้อบกพร่อง วันที่พบ วันที่แก้ไขเสร็จ เทคนิคที่ใช้ในการตรวจพบข้อบกพร่อง (ข้อมูลสำหรับเป้าหมายจัดตั้งเทคนิคและวิธีการพื้นฐานในการทดสอบ ซึ่งจะกล่าวต่อไป) และความพยายามของบุคลากรที่แก้ไขข้อบกพร่อง

รูปที่ 4.12 ตัวอย่างหน้าจอข้อบกพร่องในระดับ 2

2) เป้าหมายเริ่มกระบวนการวางแผนการทดสอบ

จากมาตรวัด 3.1.2.2 ในบทที่ 3 ต้องเก็บข้อมูลค่าใช้จ่ายในรูปของความพยายามในการพัฒนาแผนงานการทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ ดังนั้นหน้าจอที่ออกแบบต้องให้ผู้ใช้ป้อนข้อมูลความพยายามเข้าไป ดังตัวอย่างหน้าจอรูปที่ 4.13 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ เลือกประเภทแผนงานการทดสอบ และป้อนความพยายามของบุคลากรที่พัฒนา

รูปที่ 4.13 ตัวอย่างหน้าจอความพยายามในการพัฒนาแผนงานการทดสอบโมดูลในระดับ 2

สำหรับหน้าจอความพยายามในการพัฒนาแผนงานการทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ มีลักษณะเช่นเดียวกับหน้าจอความพยายามในการพัฒนาแผนงานการทดสอบโมดูล

จากมาตรวัด 3.1.2.3 ในบทที่ 3 ต้องเก็บข้อมูลรายละเอียดของกิจกรรมในแผนงานการทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ ซึ่งได้แก่ วันเริ่มต้นและสิ้นสุดของกิจกรรม ค่าใช้จ่ายในความพยายามที่ใช้ไปในกิจกรรม และจำนวนบุคลากรที่ทำในกิจกรรม ดังนั้นจึงต้องออกแบบหน้าจอเพื่อเก็บข้อมูลดังกล่าว ดังตัวอย่างหน้าจอในรูปที่ 4.14 แสดงหน้าจอแผนงานการทดสอบโมดูล ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขกิจกรรมการทดสอบที่ต้องการเพิ่มหรือแก้ไข รายละเอียดของกิจกรรม วันเริ่มต้นและสิ้นสุดของกิจกรรม จำนวนคนที่ทำการทดสอบ ความพยายามที่ใช้ หมายเลขโมดูลที่นำมาทดสอบ และหมายเลขกรณีทดสอบ

รูปที่ 4.14 ตัวอย่างหน้าจอแผนงานการทดสอบโมดูลในระดับ 2

รูปที่ 4.15 เป็นตัวอย่างหน้าจอในการป้อนข้อมูลแผนงานการทดสอบเบ็ดเสร็จ ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขกิจกรรมการทดสอบที่ต้องการเพิ่มหรือแก้ไข รายละเอียดของกิจกรรม วันเริ่มต้นและสิ้นสุดของกิจกรรม จำนวนคนที่ทำการทดสอบ ความพยายามที่ใช้ หมายเลขกรณีทดสอบ และหมายเลขโมดูลที่นำมาทดสอบรวมกัน

Integration Test Plan (TMM Level 2 Goal 2)

SoftwareID: 1 Name: Library Automation System
 Version: 1.0 Effort unit: Man.Hours Lines Of Code unit: KLDC

TestActivityID: IT1 Last updated date: 10-Aug-99
 Description: Test interface between converting date and calculating date module
 Planned start date: 01-Sep-99 Planned end date: 02-Sep-99
 Planned manpower: 2 Planned effort: 10

TestCaseID
1
2
*

ModuleIDIntegrated
1
2
*

Record: 1 of 2

Record: 1 of 2

Add Edit Save Delete Cancel Exit

รูปที่ 4.15 ตัวอย่างหน้าจอแผนงานการทดสอบเบ็ดเสร็จในระดับ 2

รูปที่ 4.16 แสดงหน้าจอแผนงานการทดสอบระบบ ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขกิจกรรมการทดสอบที่ต้องการเพิ่มหรือแก้ไข รายละเอียดของกิจกรรม วันเริ่มต้นและสิ้นสุดของกิจกรรม จำนวนคนที่ทำการทดสอบ ความพยายามที่ใช้ หมายเลขกรณีทดสอบ และหมายเลขความต้องการที่ทดสอบ

System Test Plan (TMM Level 2 Goal 2)

SoftwareID: 1 Name: Library Automation System
 Version: 1.0 Effort unit: Man.Hours Lines Of Code unit: KLDC

TestActivityID: 5 Last updated date: 10-Aug-99
 Description: Test borrowing books
 Planned start date: 25-Sep-99 Planned end date: 25-Sep-99
 Planned manpower: 2 Planned effort: 16

TestCaseID
10
11
*

RequirementIDTested
6
7
8
*

Record: 1 of 2

Record: 1 of 3

Add Edit Save Delete Cancel Exit

รูปที่ 4.16 ตัวอย่างหน้าจอแผนงานการทดสอบระบบในระดับ 2

รูปที่ 4.17 แสดงหน้าจอแผนงานการทดสอบระบบ ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขกิจกรรมการทดสอบที่ต้องการเพิ่มหรือแก้ไข รายละเอียดของกิจกรรม วันเริ่มต้น และสิ้นสุดของกิจกรรม จำนวนคนที่ทำการทดสอบ ความพยายามที่ใช้ หมายเลขกรณีทดสอบ และ หมายเลขความต้องการที่ทดสอบ

รูปที่ 4.17 ตัวอย่างหน้าจอแผนงานการทดสอบเพื่อยอมรับในระดับ 2

3) เป้าหมายจัดตั้งเทคนิคและวิธีการพื้นฐานในการทดสอบ

จากมาตรวัด 3.1.3.1 ในบทที่ 3 องค์กรต้องมีเทคนิคการทดสอบในการพัฒนากรณีทดสอบ และตรวจพบข้อบกพร่อง ดังนั้นจึงออกแบบหน้าจอสำหรับการป้อนข้อมูลเทคนิคการทดสอบขององค์กรดังรูปที่ 4.18 แต่ละเทคนิค ป้อนรหัสและรายละเอียดของเทคนิคการทดสอบ

รูปที่ 4.18 ตัวอย่างหน้าจอเทคนิคการทดสอบในระดับ 2

สำหรับหน้าจอสำหรับเทคนิคการทดสอบในการพัฒนากรณีทดสอบเป็นเช่นเดียวกับรูปที่ 4.10 และเทคนิคการทดสอบที่ใช้ในการตรวจพบข้อบกพร่องเป็นเช่นเดียวกับรูปที่ 4.12

สำหรับรายงานต่างๆ ในระดับ 2 มีดังต่อไปนี้

1) รายงานค่าใช้จ่ายในการทดสอบและการแก้ไขข้อบกพร่อง

- รายงานค่าใช้จ่ายในการทดสอบ

วัตถุประสงค์ ใช้ติดตามค่าใช้จ่ายในการทดสอบในแต่ละขั้นตอนคือ การทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ รวมถึงค่าใช้จ่ายในการพัฒนากรณีทดสอบ ขั้นตอนปฏิบัติการทดสอบ และแผนงานการทดสอบ

Unit Test Cost Report (TMM Level 2 Goal 1)				
Date: 4/9/99				
Software ID: LAS		Name: Library Automation System		
Version : 1.0	Effort Unit: Man.Hours	Lines Of Code Unit : KLOC		
Test Activity ID	Description	KLOC Tested	Effort	Unit Cost (MH/KLOC)
1	Test the module of calculating returned-date	0.52	4	7.69
2	Test the module of adding a member	1.20	6	5.00
3	Test the module of searching books	0.65	5	7.69
4	Test the module of borrowing books	2.00	8	4.00
.
.
Total:		99.42	523	
Average Unit Cost :		5.26		

รูปที่ 4.19 ตัวอย่างรายงานค่าใช้จ่ายการทดสอบโมดูล

จากตัวอย่างรายงานในรูปที่ 4.19 แสดงให้เห็นถึงจำนวนบรรทัดคำสั่งที่ทดสอบ ความพยายามของกิจกรรมการทดสอบโมดูลที่ใช้จริง และความพยายามต่อจำนวนบรรทัดคำสั่ง เพื่อใช้เปรียบเทียบระหว่างกิจกรรมหรือโครงการ ใช้ในการประมาณการค่าใช้จ่ายสำหรับโครงการอื่น และอาจตั้งค่ามาตรฐานไว้เพื่อใช้ในการควบคุมค่าใช้จ่าย สำหรับรายงานค่าใช้จ่ายการทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ มีลักษณะเช่นเดียวกับรายงานค่าใช้จ่ายการทดสอบโมดูล

รายงานค่าใช้จ่ายในการพัฒนากรณีทดสอบ จะแสดงให้เห็นถึงความพยายามรวมและค่าเฉลี่ยในการพัฒนากรณีทดสอบดังตัวอย่างในรูปที่ 4.20

Test Case Development Effort Report (TMM Level 2 Goal 1)		
Date: 4/9/99		
Software ID: LAS	Name: Library Automation System	
Version : 1.0	Effort Unit: Man.Hours	
Test Case ID	Effort	
1	5	
2	10	
3	8	
.	.	
.	.	
Total	90	433
Average		4.81

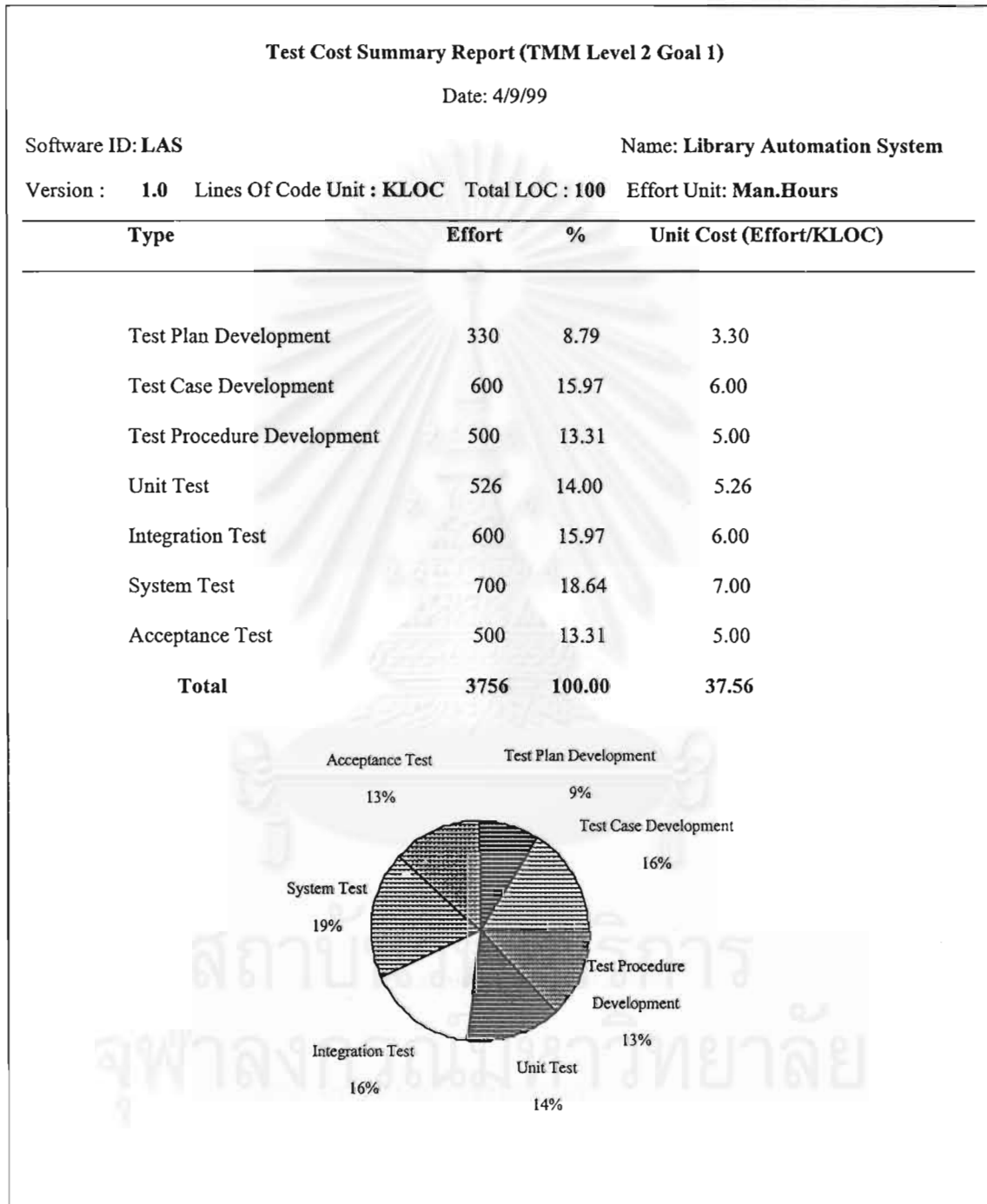
รูปที่ 4.20 ตัวอย่างรายงานค่าใช้จ่ายในการพัฒนากรณีทดสอบ

รายงานค่าใช้จ่ายในการพัฒนาแผนงานการทดสอบ เพื่อใช้ในการติดตามการพัฒนาแผนงานการทดสอบ ดังตัวอย่างในรูปที่ 4.21

Test Plan Development Effort Report (TMM Level 2 Goal 1)		
Date: 4/9/99		
Software ID: LAS	Name: Library Automation System	
Version : 1.0	Effort Unit: Man.Hours	
Test Plan Type	Effort	
Unit Test Plan	70	
Integration Test Plan	80	
System Test Plan	100	
Acceptance Test Plan	80	
Total		330

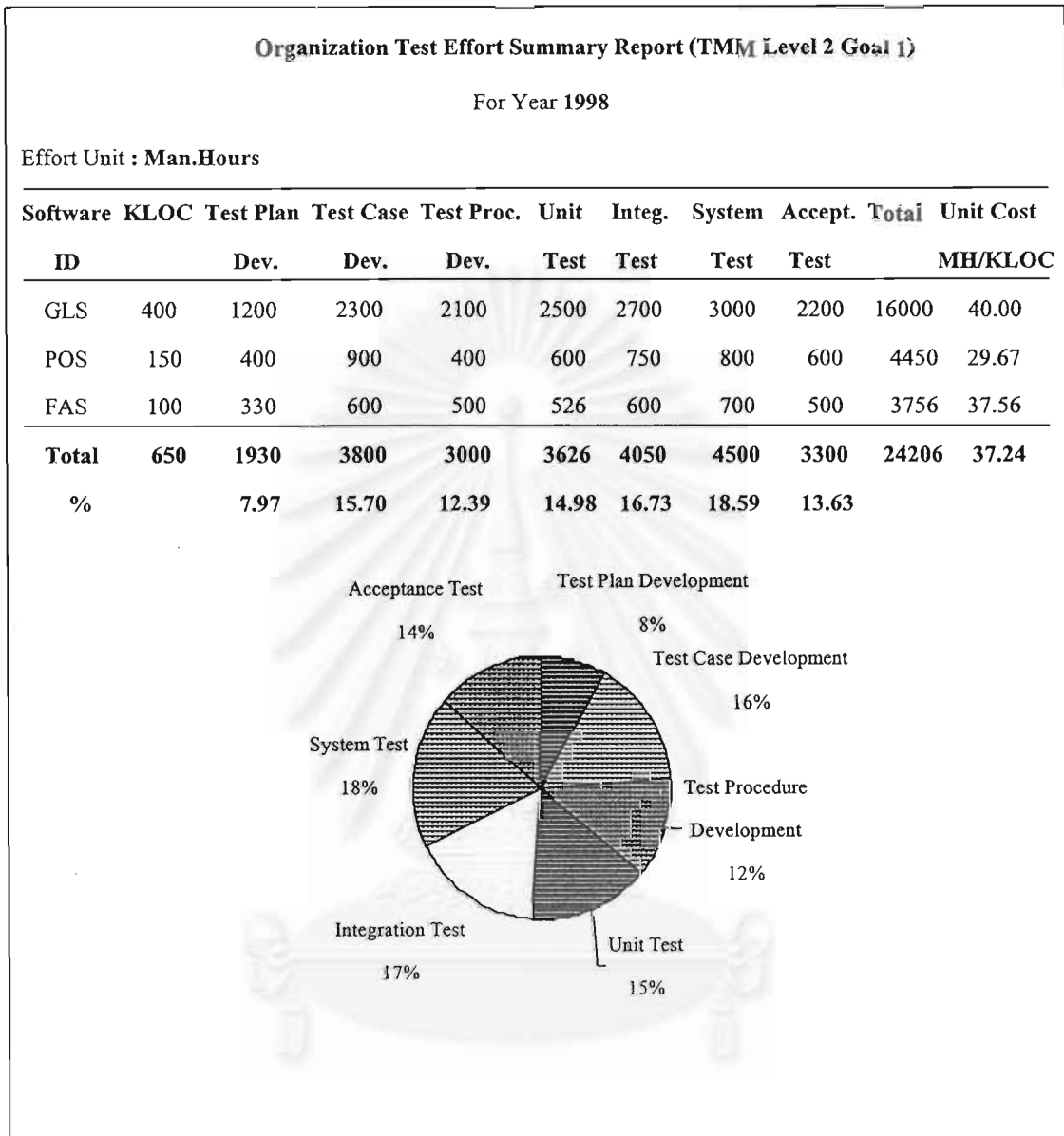
รูปที่ 4.21 ตัวอย่างรายงานค่าใช้จ่ายในการพัฒนาแผนงานการทดสอบ

รายงานสรุปค่าใช้จ่ายในการทดสอบของโครงการ ดังในรูปที่ 4.22 แสดงให้เห็นถึงเปอร์เซ็นต์ของความพยายามในการทดสอบต่างๆ และความพยายามในการทดสอบต่อจำนวนบรรทัดคำสั่ง เพื่อใช้ในการเปรียบเทียบระหว่างโครงการ ใช้ในการประมาณการค่าใช้จ่ายสำหรับโครงการอื่นและอาจตั้งค่ามาตรฐานไว้เพื่อใช้ในการควบคุมค่าใช้จ่ายของกระบวนการทดสอบ



รูปที่ 4.22 ตัวอย่างรายงานสรุปค่าใช้จ่ายในการทดสอบของโครงการ

รายงานค่าใช้จ่ายในการทดสอบโดยรวม ดังรูปที่ 4.23 เพื่อดูภาพรวมของค่าใช้จ่ายในการทดสอบขององค์กรในช่วงระยะเวลาหนึ่ง เช่น 1 ปี



รูปที่ 4.23 ตัวอย่างรายงานสรุปค่าใช้จ่ายในการทดสอบในระดับองค์กร

- รายงานค่าใช้จ่ายในการแก้ไขข้อบกพร่อง

วัตถุประสงค์ เพื่อใช้ในการติดตามในการแก้ไขข้อบกพร่อง ดังตัวอย่างรายงานในรูปที่ 4.24 แสดงให้เห็นถึงรายละเอียดของข้อบกพร่อง วันที่พบ วันที่แก้ไขเสร็จ และความพยายามในการแก้ไข

Defect Debugging Cost Report (TMM Level 2 Goal 1)				
Date: 4/9/99				
Software ID: LAS		Name: Library Automation System		
Version : 1.0		Effort Unit: Man.Hours		
Defect ID	Detail	Found Date	Closed Date	Debugging Effort
1	Not able to update a book status	01-Jul-99	02-Jul-99	2
2	Calculate returned date incorrectly	01-Jul-99	02-Jul-99	3
3	Get member data incorrectly	02-Jul-99	04-Jul-99	4
.
.
Total Number Of Defects : 250		Total Effort		800
Average defect debugging effort : 3.2				

รูปที่ 4.24 ตัวอย่างรายงานค่าใช้จ่ายในการแก้ไขข้อบกพร่อง

รายงานค่าใช้จ่ายในการแก้ไขข้อบกพร่องในภาพรวมขององค์กรในช่วงระยะเวลาหนึ่งเช่น 1 ปี ดังรูปที่ 4.25 แสดงถึงจำนวนข้อบกพร่องทั้งหมดที่แก้ไขและค่าเฉลี่ยของความพยายาม

Organization Debugging Effort Report (TMM Level 2 Goal 1)				
For Year 1998				
Effort Unit : Man.Hours				
Software ID	KLOC	Number Of Defects	Total Debugging Effort	Average Effort
GLS	400	180	800	4.44
POS	150	500	2500	5.00
FAS	100	250	800	3.20
Total	650	930	4100	4.41

รูปที่ 4.25 ตัวอย่างรายงานสรุปค่าใช้จ่ายในการแก้ไขข้อบกพร่องในระดับองค์กร

2) รายงานแผนงานการทดสอบ

วัตถุประสงค์ เพื่อใช้ในการติดตามกิจกรรมต่างๆ ตามแผนงานการทดสอบ ดังตัวอย่างรายงานในรูปที่ 4.26 แสดงให้เห็นถึงแผนงานการทดสอบโมดูล รายละเอียดแต่ละกิจกรรม วันเริ่มต้น วันสิ้นสุด จำนวนคน ความพยายาม โมดูลที่ทดสอบ และกรณีทดสอบ

Unit Test Plan Activity Report (TMM Level 2 Goal 2)							
Date: 4/8/99							
Software ID: LAS				Name: Library Automation System			
Version :1.0				Effort Unit: Man.Hours			
Activity ID	Detail	Start Date	End Date	Manpower	Effort	Module ID Tested	Test Case ID
1	Test converting date	1/7/99	1/7/99	1	3	1	1
							3
2	Test calculating returned date	1/7/99	1/7/99	1	4	2	2
							3
.
.
Total	No. Of Test Run : 100	1/7/99	15/8/99		500	100	300

รูปที่ 4.26 ตัวอย่างรายงานกิจกรรมของแผนงานการทดสอบโมดูล

ตัวอย่างรายงานแผนงานการทดสอบเบ็ดเสร็จเป็นดังรูปที่ 4.27 แสดงให้เห็นถึง รายละเอียดแต่ละกิจกรรม วันเริ่มต้น วันสิ้นสุด จำนวนคน ความพยายาม โมดูลที่นำมาทดสอบรวมกัน และกรณีทดสอบ

Integration Test Plan Activity Report (TMM Level 2 Goal 2)							
Date: 4/8/99							
Software ID: LAS				Name: Library Automation System			
Version :1.0				Effort Unit: Man.Hours			
Activity ID	Detail	Start Date	End Date	Manpower	Effort	Module ID Tested	Test Case ID
1	Test interfacing between converting and calculating returned date module	1/8/99	1/8/99	1	4	1	1
						2	2
2	Test interfacing between reading and calculating late-returned fee	1/8/99	1/8/99	1	4	3	2
						4	3
.
.
Total	No. Of Test Run : 50	1/8/99	1/9/99		500	100	200

รูปที่ 4.27 ตัวอย่างรายงานกิจกรรมของแผนงานการทดสอบเบ็ดเสร็จ

ตัวอย่างรายงานแผนงานการทดสอบระบบเป็นดังรูปที่ 4.28 แสดงให้เห็นถึง รายละเอียดแต่ละกิจกรรม วันเริ่มต้น วันสิ้นสุด จำนวนคน ความพยายาม หมายเลขความต้องการที่ทดสอบ และกรณีทดสอบ

System Test Plan Activity Report (TMM Level 2 Goal 2)							
Date: 4/8/99							
Software ID: LAS				Name: Library Automation System			
Version :1.0				Effort Unit: Man.Hours			
Activity ID	Detail	Start Date	End Date	Manpower	Effort	Req't ID	Test Case ID
1	Test calculating returned date function for members	1/8/99	1/8/99	2	4	1	1
.
.
Total	No. Of Test Run : 30	1/9/99	30/9/99		600	100	200

รูปที่ 4.28 ตัวอย่างรายงานกิจกรรมของแผนงานการทดสอบระบบ

สำหรับแผนงานการทดสอบเพื่อยอมรับมีลักษณะเช่นเดียวกับแผนงานการทดสอบระบบ

รายงานสรุปแผนงานการทดสอบดังในรูปที่ 4.29 แสดงถึง วันเริ่มต้นและสิ้นสุดของการทดสอบ จำนวนครั้งที่ทดสอบในแต่ละขั้นตอน ความพยายาม และจำนวนกรณีทดสอบ

Test Plan Summary Report (TMM Level 2 Goal 2)					
Date: 4/8/99					
Software ID: LAS			Name: Library Automation System		
Version :1.0			Effort Unit: Man.Hours		
	Start Date	End Date	No. Of Test Run	Effort	No. Of Test Cases
Unit Test	1/7/99	15/8/99	100	500	300
Integration Test	1/8/99	1/9/99	50	500	200
System Test	1/9/99	30/9/99	30	600	200
Acceptance Test	1/10/99	31/10/99	30	400	200
Total	1/7/99	31/10/99	200	2000	900

รูปที่ 4.29 ตัวอย่างรายงานสรุปแผนงานการทดสอบ

3) รายงานการจัดตั้งเทคนิคการทดสอบ

วัตถุประสงค์ เพื่อติดตามสถานะในการใช้เทคนิคในการทดสอบต่างๆ

รายงานการใช้เทคนิคการทดสอบในการพัฒนากรณีทดสอบ ดังในรูปที่ 4.30 แสดงให้เห็นถึงในแต่ละเทคนิคการทดสอบพัฒนากรณีทดสอบอะไรบ้าง และสัดส่วนในการใช้ของแต่ละเทคนิค

Test Case Development Technique Report (TMM Level 2 Goal 3)			
Date: 4/9/99			
Software ID: LAS	Name: Library Automation System		
Version :1.0	Effort Unit: Man.Hours		
Testing Technique	Test Case ID		
White Box	1		
	5		
	.		
	.		
	.		
Number Of Test Cases	Subtotal	30	40%
Black Box	2		
	3		
	.		
	.		
	.		
Number Of Test Cases	Subtotal	40	60%
		Total	70 100%

รูปที่ 4.30 ตัวอย่างรายงานการใช้เทคนิคในการพัฒนากรณีทดสอบ

รายงานการใช้เทคนิคการทดสอบในการตรวจพบข้อบกพร่องแสดงในรูปที่ 4.31 แสดงให้เห็นถึงในแต่ละเทคนิคการทดสอบตรวจพบข้อบกพร่องอะไรบ้าง และสัดส่วนในการใช้ของแต่ละเทคนิค

Testing Technique Report in Detecting Defects (TMM Level2 Goal 3)			
Date: 4/9/99			
Software ID: LAS	Name: Library Automation System		
Version :1.0	Effort Unit: Man.Hours		
Testing Technique	Defect ID		
White Box	1		
	2		
Number Of Defects	Subtotal	80	66.67%
Black Box	6		
	9		
Number Of Defects	Subtotal	40	33.33%
	Total	120	100.00%

รูปที่ 4.31 ตัวอย่างรายงานการใช้เทคนิคการทดสอบในการตรวจพบข้อบกพร่อง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.3 หน้าจอและรายงานของมาตรวัดในระดับ 3

1) จัดตั้งหน่วยงานการทดสอบซอฟต์แวร์

จากมาตรวัด 3.2.1.1 ในบทที่ 3 ต้องเก็บข้อมูลบุคลากรในหน่วยงานการทดสอบซอฟต์แวร์ ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลเป็นดังรูปที่ 4.32 ในแต่ละบุคคลป้อนหมายเลขบุคลากร ชื่อ นามสกุล และประเภทของบุคลากรที่ต้องการเพิ่มหรือแก้ไข

The screenshot shows a window titled "Persons In Test Organization (TMM Level 3 Goal 1)". It contains a table with the following data:

PersonID	Name	Surname	Person Type
0001	Aniruth	Bae	Tester
0002	Pompan	Chaiyo	Tester
0003	Sunisa	Rungruang	Tester
0005	Somchai	Krahang	Tester
0006	Charoen	SongTum	Manager

Below the table are buttons for "Add", "Edit", "Save", "Delete", "Undo", and "Exit". At the bottom, it shows "Record: 1 of 6 (Filtered)".

รูปที่ 4.32 ตัวอย่างหน้าจอบุคลากรในหน่วยงานการทดสอบซอฟต์แวร์

จากมาตรวัด 3.2.1.2 ในบทที่ 3 ต้องเก็บข้อมูลค่าใช้จ่ายเทียบกับงบประมาณของหน่วยงานการทดสอบซอฟต์แวร์ ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลเป็นดังรูปที่ 4.33 ผู้ใช้ต้องป้อนหมายเลขค่าใช้จ่ายที่ต้องการเพิ่มหรือแก้ไข รายละเอียดเกี่ยวกับค่าใช้จ่าย เดือนและปีที่เกิดค่าใช้จ่าย จำนวนเงินตามงบประมาณและจำนวนเงินที่ใช้จ่ายจริง

The screenshot shows a window titled "Test Organization Cost (TMM Level 3 Goal 1)". It contains the following fields:

- Test Cost ID: 019901
- Description: Salary
- Month Year Used: Jan-99
- Budget: \$200,000
- Actual Cost: \$220,000

At the bottom are buttons for "Add", "Edit", "Save", "Delete", "Cancel", and "Exit".

รูปที่ 4.33 ตัวอย่างหน้าจอดีค่าใช้จ่ายของหน่วยงานการทดสอบ

2) เป้าหมายจัดตั้งโปรแกรมการฝึกอบรมทางเทคนิค

จากมาตรวัด 3.2.2.1 และ 3.2.2.2 ในบทที่ 3 ต้องเก็บข้อมูลรายละเอียดการฝึกอบรมเทียบกับแผน ซึ่งได้แก่ วันที่ฝึกอบรม ค่าใช้จ่าย ระยะเวลา และจำนวนคนที่เข้ารับการฝึกอบรม ดังนั้นจึงออกแบบหน้าจอเป็นดังรูปที่ 4.34 ผู้ใช้ต้องป้อนหมายเลขหลักสูตรที่ต้องการเพิ่มหรือแก้ไข ชื่อหลักสูตร วันที่ทำการฝึกอบรมตามแผนงานกับที่เกิดขึ้นจริง ระยะเวลาที่ทำการฝึกอบรมตามแผนงานกับที่เกิดขึ้นจริง ค่าใช้จ่ายที่ใช้ในการฝึกอบรมตามแผนงานกับที่เกิดขึ้นจริง และจำนวนคนที่เข้ารับการอบรมตามแผนงานกับที่เกิดขึ้นจริง

รูปที่ 4.34 ตัวอย่างหน้าจอรายละเอียดการฝึกอบรมด้านการทดสอบ

3) เป้าหมายรวมการทดสอบเข้าไปในวงจรพัฒนาซอฟต์แวร์

จากมาตรวัด 3.2.3.1 ในบทที่ 3 ต้องเก็บข้อมูลข้อบกพร่องที่พบในแต่ละขั้นตอน ดังนั้นจึงออกแบบหน้าจอเป็นดังรูปที่ 4.35 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขข้อบกพร่องที่ต้องการเพิ่มเติมหรือแก้ไข รายละเอียดเกี่ยวกับข้อบกพร่อง วันที่พบและแก้ไขข้อบกพร่องเสร็จ เทคนิคที่ใช้ หมายเลขโมดูลที่พบข้อบกพร่อง ขั้นตอนที่พบและที่เกิดของข้อบกพร่อง และความพยายามในการแก้ไขของข้อบกพร่องของบุคคลต่างๆ

PersonID	Name	Surname	Date	Debugging Effort
0001	Aniruth	Bee	05-Aug-99	4
0002	Pompan	Chalyo	05-Aug-99	3

รูปที่ 4.35 ตัวอย่างหน้าจอรายละเอียดข้อบกพร่องในระดับ 3

4) เป้าหมายควบคุมและติดตามกระบวนการทดสอบ

จากมาตรวัด 3.2.4.1 3.2.4.2 3.2.4.3 3.2.4.4 และ 3.2.4.5 ในบทที่ 3 ต้องเก็บข้อมูลวันเริ่มต้นและสิ้นสุดของกิจกรรม ความพยายาม จำนวนบุคลากรที่ทำการทดสอบ ผลลัพธ์ของกรณีทดสอบเมื่อนำมาทดสอบ ข้อความสั่งที่ถูกทดสอบ ข้อความสั่งทั้งหมด จำนวนครั้งที่ถูกทดสอบ จำนวนคำสั่งทั้งหมด จำนวนทางเดินพื้นฐานที่ถูกทดสอบ จำนวนทางเดินพื้นฐานทั้งหมด ผลลัพธ์ของการทดสอบโมดูล จำนวนบรรทัดคำสั่ง และจำนวนข้อบกพร่องของโมดูล ดังนั้นจึงออกแบบหน้าจอจดังรูปที่ 4.36 เพื่อให้ผู้ใช้ป้อนข้อมูลดังกล่าว สำหรับผลลัพธ์ของการใช้กรณีทดสอบป้อนโดยคลิกที่ปุ่ม “TestCases” จะได้หน้าจอจดังรูปที่ 4.37 ทำการเพิ่มเติม แก้ไข ลบ หรือยกเลิกการแก้ไขกรณีทดสอบตามต้องการ

รูปที่ 4.36 ตัวอย่างหน้าจอการทดสอบโมดูลในระดับ 3

TestCaseID	Date	Result
1	10-Aug-99	Pass
3	10-Aug-99	Pass
4	10-Aug-99	Pass
*		

รูปที่ 4.37 ตัวอย่างหน้าจอสถานะกรณีทดสอบ

จากมาตรวัด 3.2.4.1 3.2.4.2 3.2.4.6 และ 3.2.4.7 ในบทที่ 3 ต้องเก็บข้อมูลวันเริ่มต้นและสิ้นสุดของกิจกรรม ความพยายาม จำนวนบุคลากรที่ทำการทดสอบ ผลลัพธ์ของกรณีทดสอบเมื่อนำมาทดสอบ ผลลัพธ์ของการทดสอบของโมดูลที่นำมารวม จำนวนบรรทัดคำสั่ง และจำนวนข้อบกพร่องที่พบคั้งนั้นจึงออกแบบหน้าจอจดังรูปที่ 4.38 เพื่อให้ผู้ใช้ป้อนข้อมูลดังกล่าวสำหรับผลลัพธ์ของการใช้กรณีทดสอบ ป้อนโดยคลิกที่ปุ่ม “TestCases” จะได้น้ำจอเช่นเดียวกับรูปที่ 4.37 ส่วนผลลัพธ์ของแต่ละโมดูลที่นำมาทดสอบรวมกัน ป้อนโดยคลิกที่ปุ่ม “Modules Integrated” จะได้น้ำจอจดังรูปที่ 4.39 ทำการเพิ่มเติม แก้ไข ลบ หรือยกเลิกการแก้ไขโมดูลตามต้องการ

รูปที่ 4.38 ตัวอย่างหน้าจอการทดสอบเบ็ดเสร็จในระดับ 3

ModuleID	Date	Result
1	10-Sep-99	Fail
2	10-Sep-99	Fail
*		

รูปที่ 4.39 ตัวอย่างหน้าจอผลลัพธ์ของแต่ละโมดูลที่ทดสอบรวมกัน

จากมาตรวัด 3.2.4.1 3.2.4.2 3.2.4.8 และ 3.2.4.9 ในบทที่ 3 ต้องเก็บข้อมูลวันเริ่มต้นและสิ้นสุดของกิจกรรม ความพยายาม จำนวนบุคลากรที่ทำการทดสอบ ผลลัพธ์ของกรณีทดสอบเมื่อนำมาทดสอบ ผลลัพธ์ของการทดสอบความต้องการ จำนวนบรรทัดคำสั่ง และจำนวนข้อบกพร่องที่พบ ดังนั้นจึงออกแบบหน้าจอ ดังรูปที่ 4.40 เพื่อให้ผู้ใช้ป้อนข้อมูลดังกล่าว สำหรับผลลัพธ์ของการใช้กรณีทดสอบ ป้อนโดยคลิกที่ปุ่ม “TestCases” จะได้น้ำจอเช่นเดียวกับรูปที่ 4.37 ส่วนผลลัพธ์ของการทดสอบความต้องการ ป้อนโดยคลิกที่ปุ่ม “Requirement Status” จะได้น้ำจอ ดังรูปที่ 4.41 ทำการเพิ่มเติม แก้ไข ลบ หรือยกเลิกการแก้ไขหมายเลขความต้องการตามที่ต้องการ

รูปที่ 4.40 ตัวอย่างหน้าจอการทดสอบระบบในระดับ 3

RequirementID	Date	Result
6	01-Oct-99	Fail
7	01-Oct-99	Pass
8	01-Oct-99	Fail
*		

รูปที่ 4.41 ตัวอย่างหน้าจอผลลัพธ์การทดสอบความต้องการในการทดสอบระบบ

สำหรับหน้าจอการทดสอบเพื่อยอมรับจะมีลักษณะเช่นเดียวกับหน้าจอการทดสอบระบบดังรูปที่ 4.42

The screenshot shows a software window titled "AcceptanceTest : Form". The main heading is "Acceptance Test (TMM Level 3 Goal 4)". The form contains the following fields and values:

- SoftwareID: 1
- Name: Library Automation System
- Version: 1.0
- Effort unit: Man.Hours
- Lines Of Code unit: KLOC
- TestActivityID: 1
- Description: Test adding, editing, deleting and searching members
- Lines Of Code: 100
- Planned Start Date: 02-Oct-99
- Actual Start Date: 17-Oct-99
- Planned End Date: 03-Oct-99
- Actual End Date: 18-Oct-99
- Planned Manpower: 3
- Actual Manpower: 2
- Planned Effort: 24
- Actual Effort: 20
- Result: Fail
- Number of defects: 5

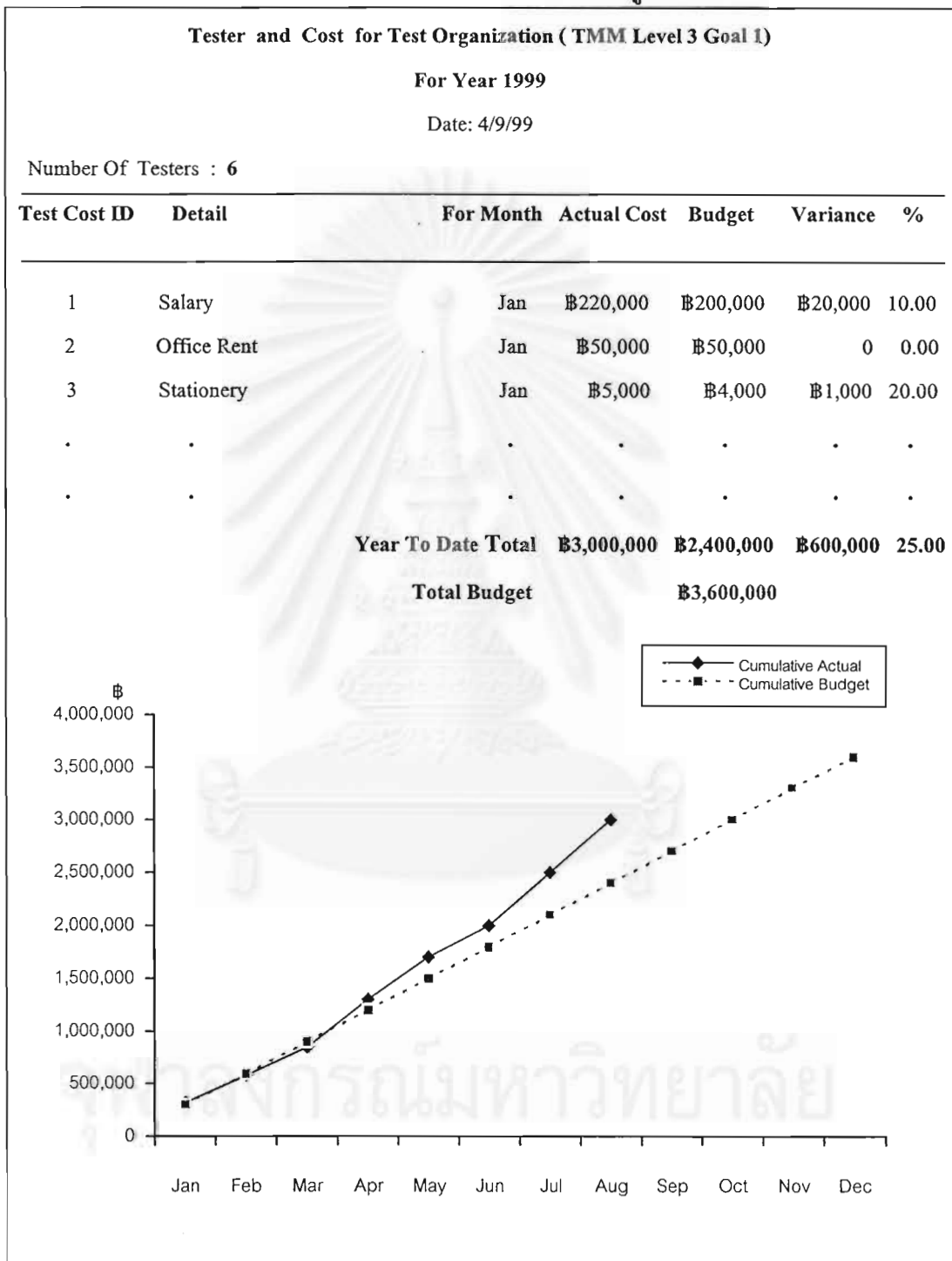
Buttons at the bottom include: TestCases, Requirement Status, Note (text area), Previous, Next, Add, Edit, Save, Delete, Cancel, and Exit.

รูปที่ 4.42 ตัวอย่างหน้าจอการทดสอบเพื่อยอมรับในระดับ 3

สำหรับรายงานต่างๆ ในระดับ 3 มีดังต่อไปนี้

1) รายงานจำนวนผู้ทดสอบและค่าใช้จ่ายของหน่วยงานการทดสอบ

วัตถุประสงค์ เพื่อติดตามสถานะของหน่วยงานการทดสอบว่ามีจำนวนผู้ทดสอบเท่าใด และค่าใช้จ่ายของหน่วยงานเป็นอย่างไร ดังตัวอย่างรายงานในรูปที่ 4.43



รูปที่ 4.43 ตัวอย่างรายงานจำนวนผู้ทดสอบและค่าใช้จ่ายของหน่วยงานการทดสอบ

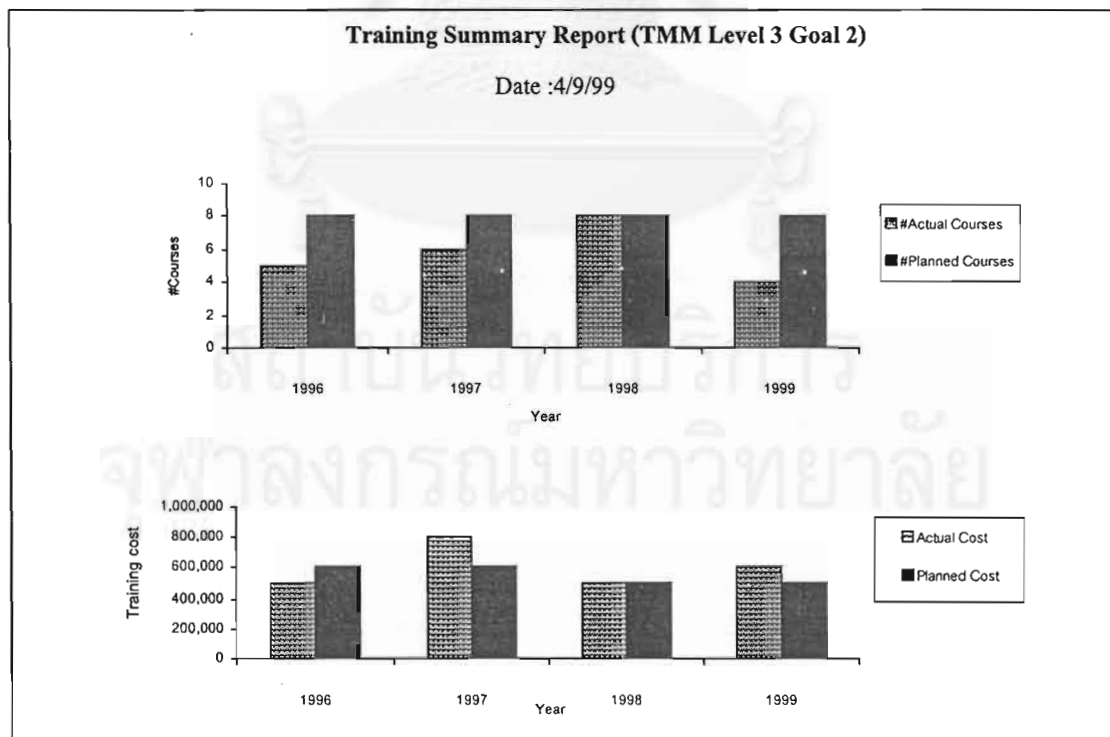
2) รายงานการฝึกอบรมด้านการทดสอบ

วัตถุประสงค์ เพื่อใช้ในการติดตามว่าบุคลากรที่เกี่ยวข้องกับการทดสอบได้รับการฝึกอบรมเกี่ยวกับการทดสอบมากน้อยเพียงใด รวมทั้งติดตามการฝึกอบรมว่าเป็นไปตามแผนงานหรือไม่ รายงานโปรแกรมการฝึกอบรมเมื่อเทียบกับแผนงาน ดังแสดงในรูปที่ 4.44 แสดงให้เห็นถึงวันที่ทำการอบรม ระยะเวลาที่ใช้ในการอบรม จำนวนคนที่เข้ารับการอบรมและค่าใช้จ่ายที่ใช้ในการอบรมเมื่อเทียบกับแผนงาน

Training Status Report (TMM Level 3 Goal 2)													
Date : 4/9/99													
Course ID	Course Name	Start Date			Duration(days)			#Attendances			Cost(฿)		
		Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	%Var
1	Test coverage tool	3/1/99	3/1/99	0	2	2	0	6	6	0	20,000	20,000	0
2	Requirement trace tool	15/3/99	1/3/99	14	3	3	0	4	6	0	50,000	40,000	25
.
.
Total Courses : 8					15	13	2	42	48	-6	300,000	270,000	11

รูปที่ 4.44 ตัวอย่างรายงานสถานะของการฝึกอบรมด้านการทดสอบ

รายงานสรุปจำนวนหลักสูตร และค่าใช้จ่ายเกี่ยวกับการทดสอบต่อไตรมาสหรือปี เป็นดังรูปที่ 4.45



รูปที่ 4.45 ตัวอย่างรายงานสรุปการฝึกอบรมประจำปี

3) รายงานการรวมการทดสอบเข้าไปในวงจรพัฒนาซอฟต์แวร์

วัตถุประสงค์ เพื่อใช้ในการประเมินประสิทธิผลของกระบวนการทดสอบในการรวมเข้าไปในวงจรพัฒนาซอฟต์แวร์

รายงานในรูปที่ 4.46 แสดงให้เห็นถึงรายละเอียดข้อบกพร่องของโครงการว่าถูกพบที่ขั้นตอนใดและเกิดจากขั้นตอนใดในวงจรพัฒนาซอฟต์แวร์

Defect By Phase Detail Report (TMM Level 3 Goal 3)					
Date:4/9/99					
Software ID: LAS			Name: Library Automation System		
Version :1.0			Effort Unit: Man.Hours		
Defect ID	Detail	Found Date	Closed Date	Found	Origin
1	Not able to update a book status	01-Jul-99	02-Jul-99	Design	Design
2	Calculate returned date incorrectly	01-Jul-99	02-Jul-99	Design	Req't
.
.

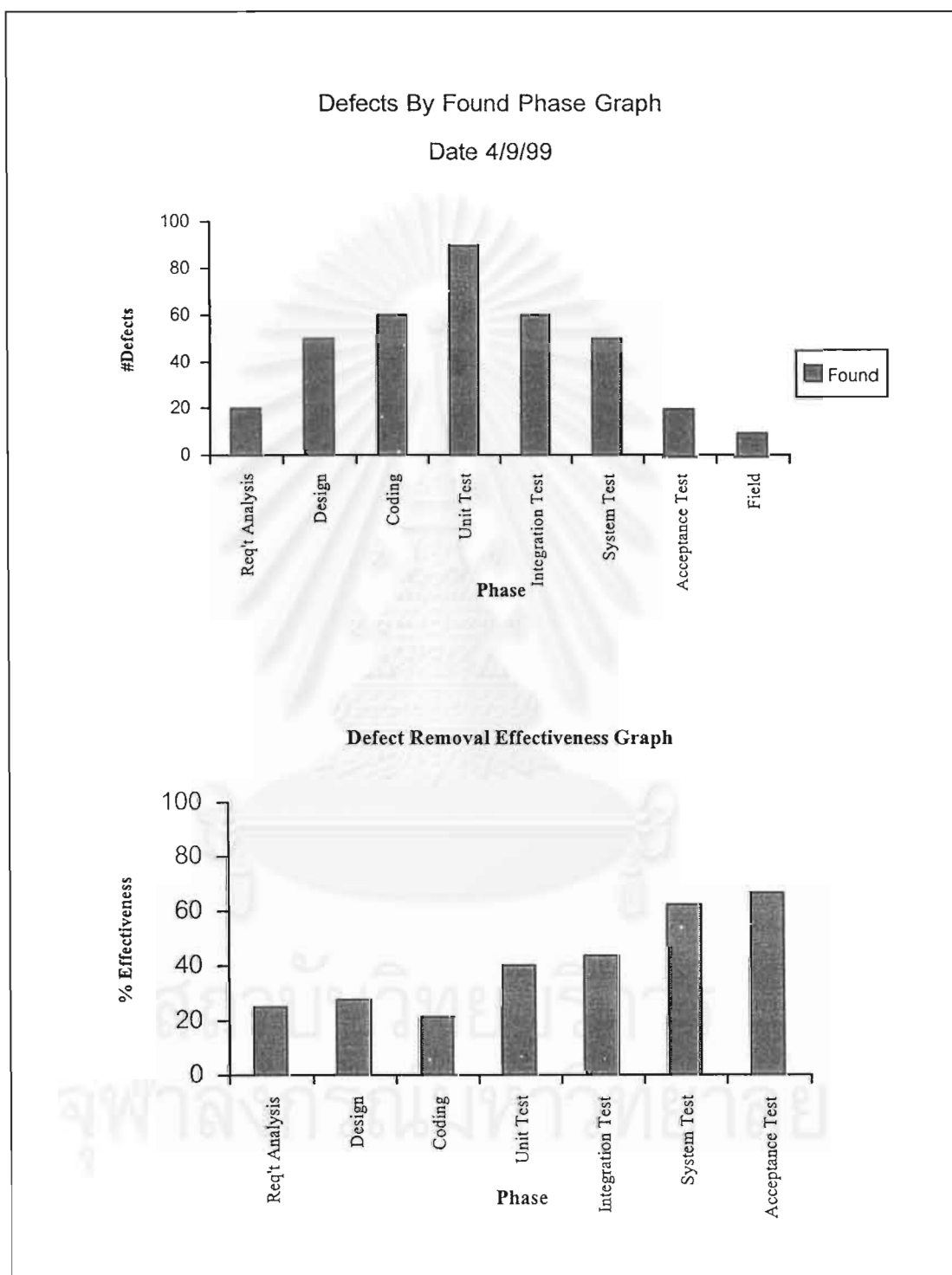
รูปที่ 4.46 ตัวอย่างรายงานรายละเอียดของข้อบกพร่องในแต่ละขั้นตอน

รายงานในรูปที่ 4.47 เป็นรายงานสรุปจำนวนและสัดส่วนข้อบกพร่องของโครงการที่ถูกพบ เกิดขึ้น นี้ออก และประสิทธิผลของการจัดข้อบกพร่องในแต่ละขั้นตอน

Defect By Phase Summary Report (TMM Level 3 Goal 3)							
Date:4/9/99							
Software ID: LAS				Name: Library Automation System			
Version : 1.0				Effort Unit: Man.Hours			
Phase	Number Of Defects						
	Origin	%	Total in Phase	Found	%	Escape	% Effectiveness
Req't Analysis	80	22.22	80	20	5.55	60	25.00
Design	120	33.33	180	50	13.89	130	27.78
Coding	150	41.67	280	60	16.67	220	21.43
Unit Test	4	1.11	224	90	25.00	134	40.18
Integration Test	3	0.83	137	60	16.67	77	43.80
System Test	3	0.83	80	50	13.89	30	62.50
Acceptance Test	0	0.00	30	20	5.55	10	66.67
Field	0	0.00	10	10	2.78	0	100.00
Total	360	100.00		360	100.00		

รูปที่ 4.47 ตัวอย่างรายงานสรุปข้อบกพร่องในแต่ละขั้นตอน

ในรายงานสรุปข้อบกพร่องของโครงการในแต่ละขั้นตอน สามารถแสดงผลในรูปแบบของกราฟ ดังรูปที่ 4.48 เป็นกราฟแสดงถึงจำนวนข้อบกพร่องที่ถูกลบและประสิทธิภาพในการกำจัดข้อบกพร่องในแต่ละขั้นตอน



รูปที่ 4.48 ตัวอย่างกราฟในรายงานสรุปข้อบกพร่องในแต่ละขั้นตอน

4) รายงานการควบคุมกระบวนการทดสอบ

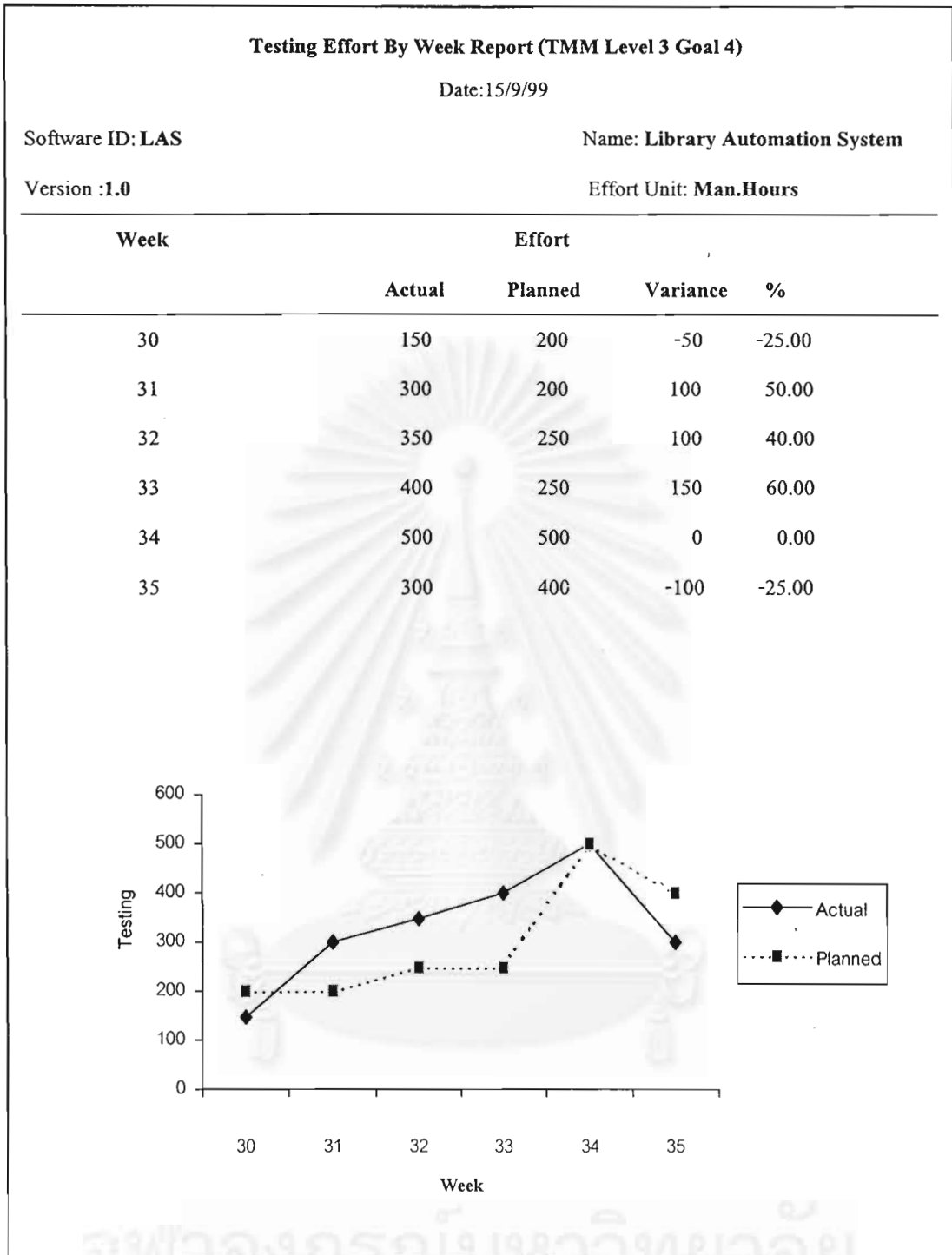
- รายงานสถานะการทดสอบ

วัตถุประสงค์ ใช้ติดตามและควบคุมสถานะของกิจกรรมการทดสอบของโครงการเมื่อเทียบกับแผนงานการทดสอบได้แก่ การทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ ตัวอย่างรายงานในรูปที่ 4.49 แสดงถึงสถานะของกิจกรรมการทดสอบโมดูลคือความแตกต่างระหว่างวันที่เริ่มกิจกรรมจริงเทียบกับแผนงาน วันสิ้นสุดกิจกรรมจริงเทียบกับแผนงาน จำนวนคนที่ทำกิจกรรมจริงเทียบกับแผนงาน ความพยายามที่แท้จริงเทียบกับแผนงาน ความพยายามต่อจำนวนบรรทัดคำสั่ง เพื่อใช้เปรียบเทียบระหว่างกิจกรรมหรือโครงการใช้ในการประมาณการค่าใช้จ่ายสำหรับโครงการอื่นและอาจตั้งค่ามาตรฐานไว้เพื่อใช้ในการควบคุมค่าใช้จ่าย ผลลัพธ์ของการทดสอบและจำนวนข้อบกพร่อง

Unit Test Activity Report (TMM Level 3 Goal 4)																	
Date:4/9/99																	
Software ID: LAS								Name: Library Automation System									
Version : 1.0				Effort Unit: Man.Hours				Lines Of Code Unit : KLOC									
Test Act	KLOC	Start Date	End Date	Manpower		Effort		MH	Result	#Defects							
ID	Tested	Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	Var	%	/KLOC					
1	0.52	1/7/99	1/7/99	0	1/7/99	1/7/99	0	1	1	0	4	3	1	33.33	7.69	P	
2	1.20	2/7/99	1/7/99	1	2/7/99	1/7/99	1	1	1	0	6	4	2	50.00	5.00	P	
3	0.65	2/7/99	2/7/99	0	3/7/99	2/7/99	1	2	1	1	5	5	0	0.00	7.69	F	5
.
.
Total :		99.42						523		450		73		16.22		100	
												Average Unit Cost :				5.26	
												Total Test Run:100		# Passed 60			
														# Failed 40			

รูปที่ 4.49 ตัวอย่างรายงานกิจกรรมการทดสอบโมดูล

รายงานความพยายามในการทดสอบของโครงการตามระยะเวลา ดังรูปที่ 4.50 แสดงให้เห็นถึงความพยายามตามระยะเวลาเมื่อเทียบกับแผนงาน ใช้ในการติดตามและควบคุมค่าใช้จ่ายของการทดสอบ



รูปที่ 4.50 ตัวอย่างรายงานความพยายามในการทดสอบของโครงการตามระยะเวลา

- รายงานสถานะกรณีทดสอบ

วัตถุประสงค์ เพื่อใช้ในการติดตามสถานะในการพัฒนากรณีทดสอบและผลลัพธ์ของการนำกรณีทดสอบมาทดสอบในการทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับของโครงการ

รายงานในรูปที่ 4.51 แสดงให้เห็นถึงสถานะในการพัฒนากรณีทดสอบของโครงการคือความแตกต่างระหว่างวันที่เริ่มกิจกรรมจริงเทียบกับแผนงาน วันสิ้นสุดกิจกรรมจริงเทียบกับแผนงาน และความพยายามที่ทำจริงเทียบกับแผนงาน

Test Case Development Status Report (TMM Level 3 Goal 4)										
Date: 4/9/99										
Software ID: LAS					Name: Library Automation System					
Version : 1.0					Effort Unit: Man.Hours					
Test Case ID	Start Date			End Date			Effort			
	Actual	Plan	Var.	Actual	Plan	Var.	Actual	Plan	Var.	%
1	1/6/99	1/6/99	0	1/6/99	1/6/99	0	5	5	0	0.00%
2	1/6/99	1/6/99	0	2/6/99	1/6/99	1	10	8	2	25.00%
.
.
Total	90						433	400	33	8.25%
Average							4.81	4.44	0.37	8.25 %

รูปที่ 4.51 ตัวอย่างรายงานสถานะในการพัฒนากรณีทดสอบ

รายงานสถานะของการนำกรณีทดสอบมาทดสอบในแต่ละกิจกรรมการทดสอบ ดังตัวอย่างรายงานในรูปที่ 4.52 เพื่อติดตามผลลัพธ์ของการนำกรณีทดสอบมาทดสอบ

Test Case Status Of Unit Test Activity Report (TMM Level 3 Goal 4)				
Date: 4/9/99				
Software ID: LAS		Name: Library Automation System		
Version :1.0		Effort Unit: Man.Hours		
Test Activity ID	Date	Test Case ID	Result	
1	1/7/99	1	P	
	1/7/99	2	P	
2	1/7/99	3	P	
.	.	.	.	
.	.	.	.	
Total Test Cases : 150 # Test Cases Exercised : 80 # Passed : 60				
# Failed : 20				

รูปที่ 4.52 ตัวอย่างรายงานการใช้กรณีทดสอบในแต่ละกิจกรรมการทดสอบ

รายงานสรุปสถานะของการนำกรณีทดสอบมาทดสอบของโครงการในแต่ละขั้นตอนการทดสอบดังในรูปที่ 4.53 แสดงให้เห็นถึงจำนวนกรณีทดสอบที่ต้องทดสอบทั้งหมดในแต่ละขั้นตอน จำนวนที่ได้ทดสอบไป จำนวนที่ทดสอบผ่านและจำนวนที่ทดสอบไม่ผ่าน

Test Case Status Summary Report (TMM Level 3 Goal 4)							
Date: 15/7/99							
Software ID: LAS				Name: Library Automation System			
Version :1.0				Effort Unit: Man.Hours			
Test Phase	Number Of Test Cases						
	Total	Exercised	%	Passed	%	Failed	%
Unit Test	150	50	33.33	30	20.00	20	13.33
Integration Test	130	0					
System Test	120	0					
Acceptance Test	100	0					
Total	500	50	10.00	30	6.00	20	4.00

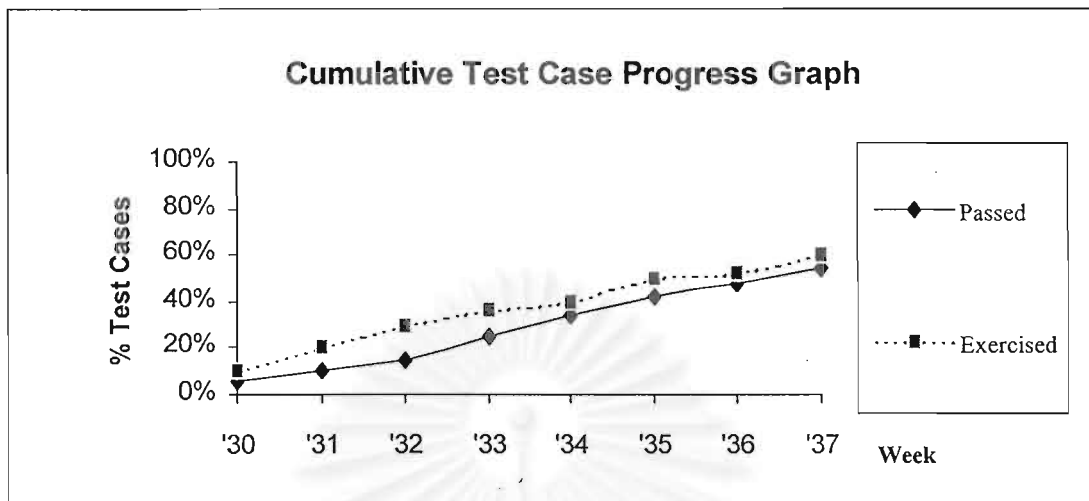
รูปที่ 4.53 ตัวอย่างรายงานสรุปสถานะของการใช้กรณีทดสอบในแต่ละขั้นตอนการทดสอบ

รายงานในรูปที่ 4.54 แสดงให้เห็นถึงสถานะของการนำกรณีทดสอบมาทดสอบของโครงการตามระยะเวลา ว่าแต่ละสัปดาห์มีการนำกรณีทดสอบมาทำการทดสอบจำนวนเท่าใด และผ่านเท่าใด โดยแสดงเป็นจำนวนสะสมตามระยะเวลา

Test Case Progress Report (TMM Level 3 Goal 4)							
Date: 4/9/99							
Software ID: LAS				Name: Library Automation System			
Version : 1.0				Effort Unit: Man.Hours			
Week	Number Of Test Cases						
	Total	Exercised	Cumulative	Passed	Cumulative	%	
30	500	50	50	30	30	6.00	
31	500	50	100	20	50	10.00	
.	
.	

รูปที่ 4.54 ตัวอย่างรายงานความก้าวหน้าของกรณีทดสอบในการทดสอบตามระยะเวลา

รายงานความก้าวหน้าของกรณีทดสอบในการทดสอบตามระยะเวลาแสดงผลในรูปของกราฟดังในรูปที่ 4.55



รูปที่ 4.55 ตัวอย่างกราฟความก้าวหน้าของกรณีทดสอบในการทดสอบตามระยะเวลา

- รายงานความก้าวหน้าการทดสอบโมดูล

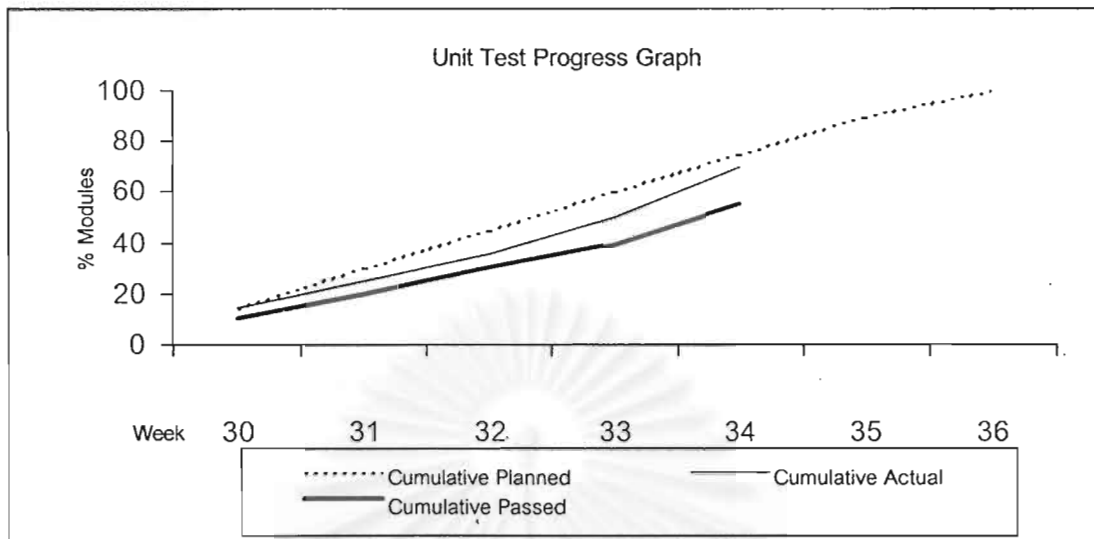
วัตถุประสงค์ เพื่อใช้ในการติดตามความก้าวหน้าของโมดูลของโครงการที่นำมาทดสอบว่าทดสอบผ่านไปแล้วเท่าใด และใช้ในการประมาณการว่าจะใช้เวลาอีกเท่าใดที่ทุกโมดูลจะผ่านการทดสอบ โมดูลทั้งหมด

รายงานในรูปที่ 4.56 แสดงให้เห็นถึงสถานะของการทดสอบโมดูลตามระยะเวลาว่าแต่ละสัปดาห์มีการทดสอบโมดูลจำนวนเท่าใดตามแผน จำนวนที่ทดสอบจริงและจำนวนที่ผ่าน โดยแสดงเป็นจำนวนสะสมตามระยะเวลา

Unit Test Progress Report (TMM Level 3 Goal 4)									
Date: 4/9/99									
Software ID: LAS					Name: Library Automation System				
Version : 1.0					Effort Unit: Man.Hours				
Week	Number Of Modules								
	Planned	Cumulative	%	Exercised	Cumulative	%	Passed	Cumulative	%
30	15	15	15.00	15	15	15.00	10	10	10.00
31	20	35	35.00	20	35	35.00	15	25	25.00
.
.
Total Number Of Unit Test Runs : 100									

รูปที่ 4.56 ตัวอย่างรายงานความก้าวหน้าของการทดสอบโมดูลตามระยะเวลา

รายงานความก้าวหน้าของการทดสอบโมดูลตามระยะเวลาแสดงผลในรูปแบบของกราฟดังในรูปที่ 4.57



รูปที่ 4.57 ตัวอย่างกราฟความก้าวหน้าของการทดสอบโมดูลตามระยะเวลา

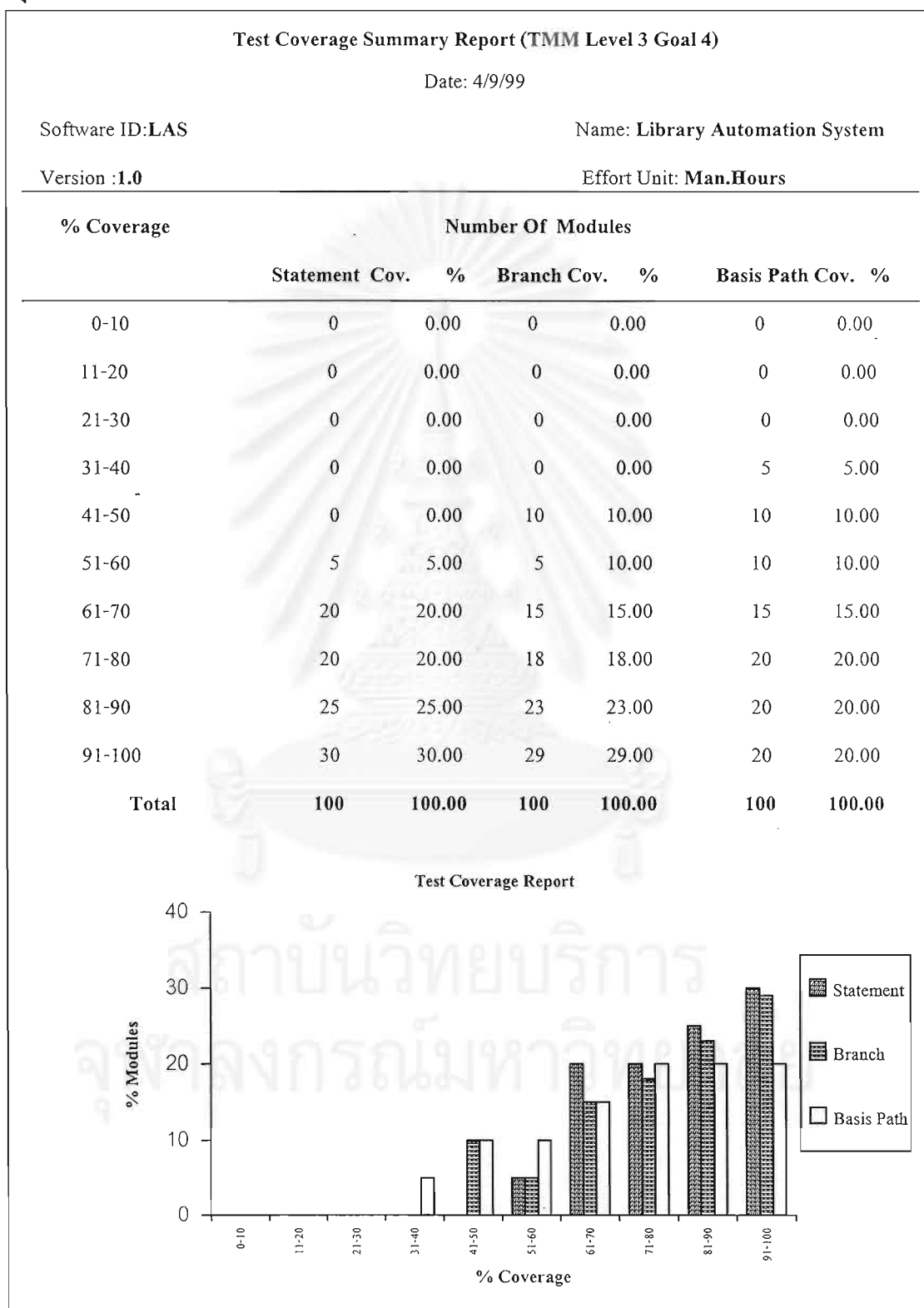
- รายงานความครอบคลุมการทดสอบของการทดสอบโมดูล

วัตถุประสงค์ เพื่อติดตามความครอบคลุมการทดสอบข้อความสั่ง ความครอบคลุมกิ่ง และความครอบคลุมทางเดินพื้นฐานของการทดสอบโมดูลของโครงการ ดังตัวอย่างรายงานในรูปที่ 4.58 แสดงให้เห็นถึง จำนวนบรรทัดคำสั่ง จำนวนข้อความสั่งที่ถูกทดสอบ จำนวนข้อความสั่งทั้งหมด เปอร์เซ็นต์ความครอบคลุมข้อความสั่ง จำนวนกิ่งที่ถูกทดสอบ จำนวนกิ่งทั้งหมด เปอร์เซ็นต์ความครอบคลุมกิ่ง จำนวนทางเดินพื้นฐานที่ถูกทดสอบ จำนวนทางเดินพื้นฐานทั้งหมด และ เปอร์เซ็นต์ความครอบคลุมทางเดินพื้นฐานของโมดูล

Test Coverage Report (TMM Level 3 Goal 4)											
Date: 4/9/99											
Software ID: LAS						Name: Library Automation System					
Version : 1.0				Effort Unit: Man.Hours				Lines Of Code Unit : KLOC			
Module ID	LOC	Statement Coverage			Branch Coverage			Basis Path Coverage			
		Exercised	Total	%	Exercised	Total	%	Exercised	Total	%	
1	0.5	400	450	88.88	35	40	87.50	20	25	80.00	
2	0.8	750	750	100.00	60	65	92.31	80	90	88.88	
.	
.	

รูปที่ 4.58 ตัวอย่างรายงานความครอบคลุมการทดสอบของการทดสอบโมดูล

รายงานในรูปแบบที่ 4.59 แสดงให้เห็นถึงเปอร์เซ็นต์ของจำนวนโมดูลในช่วงของเปอร์เซ็นต์ความครอบคลุมต่างๆ เพื่อคุณภาพของการทดสอบโมดูลโดยรวม ถ้าจำนวนของโมดูลที่มีความครอบคลุมสูงมีอยู่จำนวนน้อย แสดงว่าการทดสอบโมดูลมีคุณภาพต่ำ ถ้ามีอยู่จำนวนสูงคุณภาพก็ดี



รูปที่ 4.59 ตัวอย่างรายงานแสดงจำนวนโมดูลในช่วงเปอร์เซ็นต์ความครอบคลุมต่างๆ

- รายงานความหนาแน่นของข้อบกพร่องของโมดูล

วัตถุประสงค์ เพื่อใช้ในการติดตามคุณภาพโมดูลของโครงการ โดยการดูความหนาแน่นของข้อบกพร่องของโมดูลคือจำนวนข้อบกพร่องหารด้วยจำนวนบรรทัดคำสั่งของโมดูล ดังตัวอย่างรายงานในรูปที่ 4.60

Defect Density Report (TMM Level 3 Goal 4)			
Date: 24/9/99			
Software ID:LAS	Name: Library Automation System		
Version :1.0	Effort Unit: Man.Hours		
Module ID	KLOC	#Defects	#Defects/KLOC
1	0.52	5	9.62
2	1.20	4	3.33
.	.	.	.
.	.	.	.

รูปที่ 4.60 ตัวอย่างรายงานความหนาแน่นของข้อบกพร่องของโมดูล

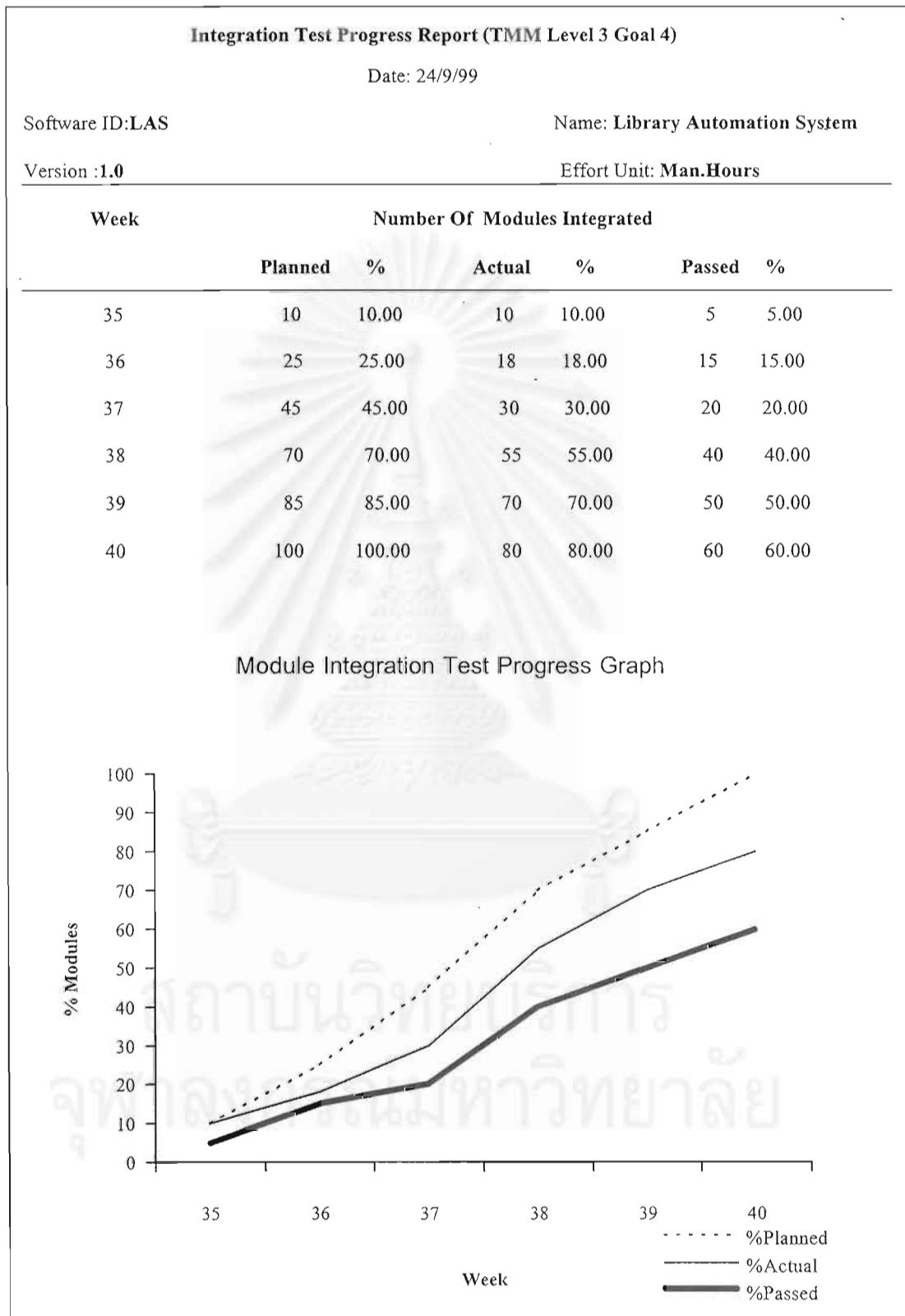
- รายงานสถานะการทดสอบเบ็ดเสร็จ

วัตถุประสงค์ เพื่อใช้ติดตามสถานะการทดสอบของโครงการเมื่อนำโมดูลต่าง ๆ มารวมกันว่าทดสอบผ่านหรือไม่ รายงานรายละเอียดผลลัพธ์ของโมดูลของโครงการเมื่อนำมาทดสอบรวมกันตามกิจกรรมการทดสอบเบ็ดเสร็จ ดังตัวอย่างรายงานในรูปที่ 4.61

Integration Test Status Report (TMM Level 3 Goal 4)			
Date: 24/9/99			
Software ID:LAS	Name: Library Automation System		
Version :1.0	Effort Unit: Man.Hours		
Test Activity ID	Date	Module ID	Result
1	1/9/99	1	P
		2	P
		3	F
2	2/9/99	1	P
		2	P
		3	F
.	.	.	.
.	.	.	.
Total Test Run : 50		# Modules Integrated : 80	# Modules Integrated Successfully : 50

รูปที่ 4.61 ตัวอย่างรายงานสถานะของโมดูลตามกิจกรรมการทดสอบเบ็ดเสร็จ

รายงานความก้าวหน้าของการทดสอบเบ็ดเสร็จตามระยะเวลา ดังตัวอย่างในรูปที่ 4.62 แสดงให้เห็นถึงจำนวนโมดูลที่นำมาทดสอบรวมตามแผนงาน จำนวนโมดูลที่นำมาทดสอบรวมกันจริงและจำนวน โมดูลที่ทดสอบรวมกันผ่าน



รูปที่ 4.62 ตัวอย่างรายงานความก้าวหน้าการทดสอบเบ็ดเสร็จตามระยะเวลา

- รายงานสถานะการทดสอบความต้องการ

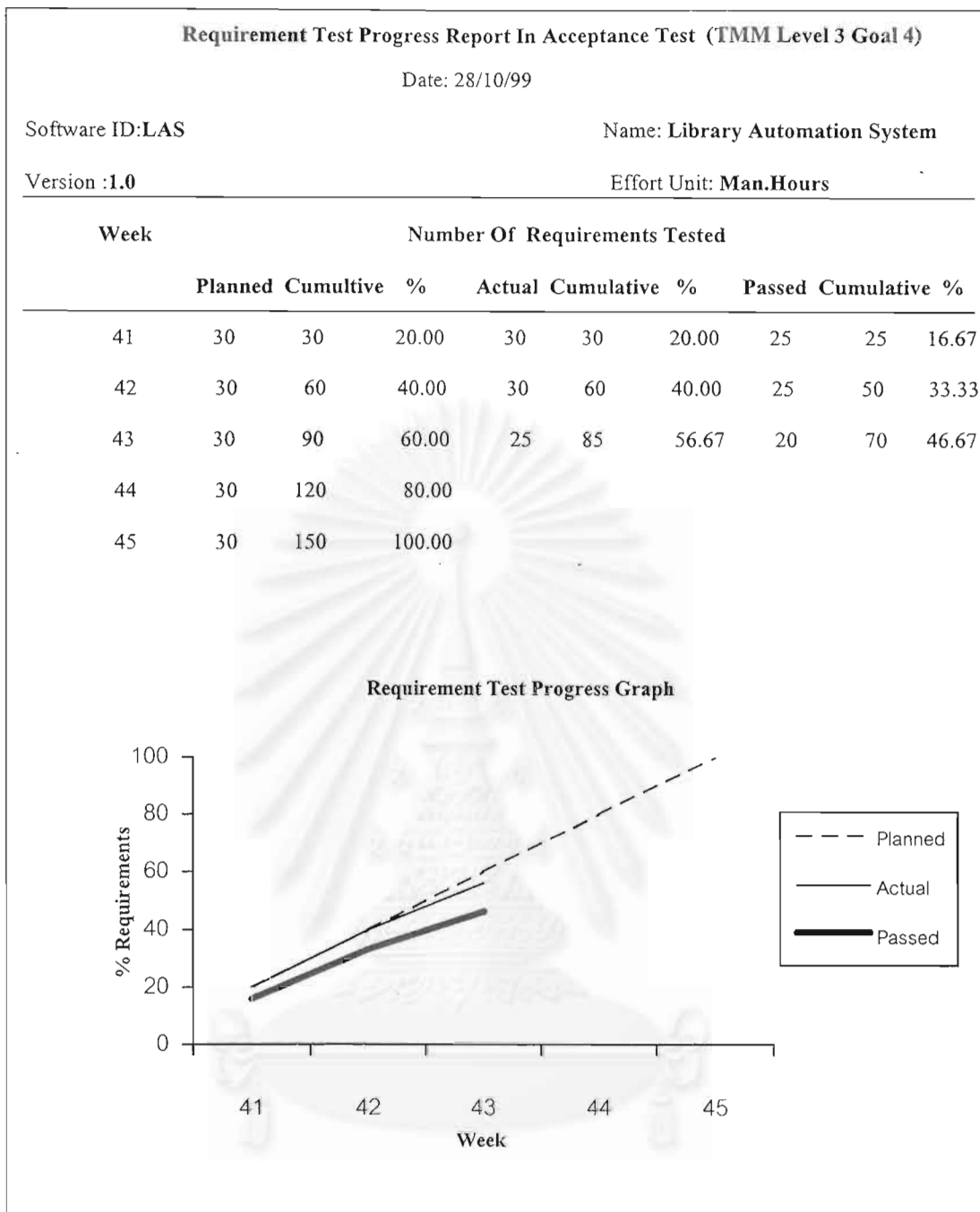
วัตถุประสงค์ เพื่อติดตามสถานะการทดสอบความต้องการของโครงการในขั้นตอนการทดสอบระบบและการทดสอบเพื่อยอมรับ ว่าแต่ละความต้องการได้รับการทดสอบหรือไม่ ผลลัพธ์การทดสอบเป็นอย่างไร

รายงานรายละเอียดผลลัพธ์ของการทดสอบความต้องการตามกิจกรรมการทดสอบ ดังตัวอย่างรายงานในรูปที่ 4.63 แสดงถึงผลลัพธ์ของการทดสอบความต้องการในการทดสอบเพื่อยอมรับ

Requirement Test Status of Acceptance Test Activity Report (TMM Level 3 Goal 4)			
Date: 1/10/99			
Software ID: LAS		Name: Library Automation System	
Version :1.0		Effort Unit: Man.Hours	
Test Activity ID	Date	Requirement ID	Result
1	1/9/99	1	P
		2	P
		3	F
2	2/9/99	4	P
		5	P
		6	F
Total Test Run : 30		# Total Requirements : 150	
		# Requirement Tested : 100 66.67 %	
		# Requirements Passed : 80 53.33 %	

รูปที่ 4.63 ตัวอย่างรายงานสถานะของการทดสอบความต้องการ

รายงานความก้าวหน้าของการทดสอบความต้องการตามระยะเวลา ดังตัวอย่างในรูปที่ 4.64 แสดงให้เห็นถึงจำนวนความต้องการที่ทดสอบตามแผนงาน จำนวนความต้องการที่ทดสอบจริงและจำนวนความต้องการที่ทดสอบผ่าน



รูปที่ 4.64 ตัวอย่างรายงานความก้าวหน้าการทดสอบความต้องการตามระยะเวลา

- รายงานความหนาแน่นข้อบกพร่องของซอฟต์แวร์

วัตถุประสงค์ เพื่อติดตามคุณภาพของซอฟต์แวร์ตั้งแต่การทดสอบระบบตามระยะเวลา โดยดูได้จากความหนาแน่นของข้อบกพร่องคือจำนวนข้อบกพร่องหารด้วยจำนวนบรรทัดคำสั่ง นอกจากนี้ยังสามารถใช้รายงานในการประเมินความน่าเชื่อถือของระบบในทีเอ็มเอ็มระดับ 4 และ 5 ดังตัวอย่างรายงานในรูปที่ 4.65 แสดงให้เห็นถึงความหนาแน่นของข้อบกพร่องของซอฟต์แวร์และความหนาแน่นสะสมตามระยะเวลา

Defect Density By Time Report (TMM Level 3 Goal 4)

Date:1/11/99

Software ID:LAS

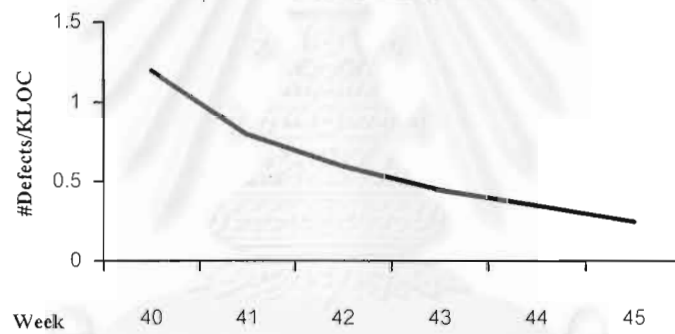
Name: Library Automation System

Version :1.0

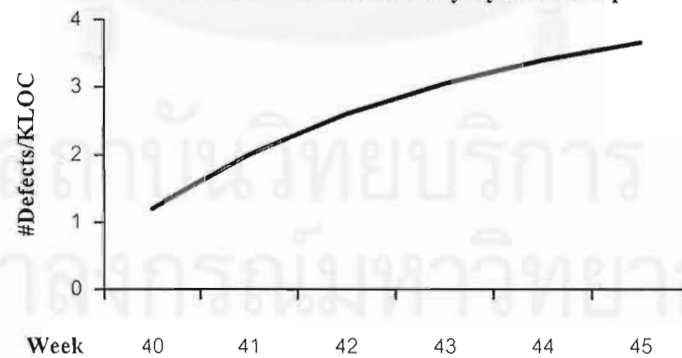
Effort Unit: Man.Hours

Week	#Defects	KLOC	#Defects/KLOC	
			Arrival	Cumulative
40	120	100	1.20	1.20
41	80	100	0.80	2.00
42	60	100	0.60	2.60
43	45	100	0.45	3.05
44	35	100	0.35	3.40
45	25	100	0.25	3.65

Defect Density By Time Graph



Cumulative Defect Density By Time Graph



รูปที่ 4.65 ตัวอย่างรายงานความหนาแน่นของข้อบกพร่องตามระยะเวลา

4.2.4 หน้าจอและรายงานของมาตรวัดในระดับ 4

1) เป้าหมายจัดตั้งโปรแกรมการทบทวนทั่วทั้งองค์กร

จากมาตรวัด 3.3.1.2 ในบทที่ 3 ต้องเก็บข้อมูลแผนงานการทบทวน ได้แก่ ปริมาณที่ทำการทบทวน วันที่ทำการทบทวน จำนวนคนที่ทำการทบทวน เวลาที่ใช้ในการเตรียมการ และเวลาที่ใช้ในการตรวจ ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลดังรูปที่ 4.66 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขการทบทวนที่ต้องการเพิ่มหรือแก้ไข รายละเอียดการทบทวน ประเภทของผลผลิตที่ทบทวน ปริมาณที่ทำการทบทวน วันที่ทำการทบทวน จำนวนคนที่ทำการทบทวน เวลาที่ใช้ในการเตรียมการ และเวลาที่ใช้ในการตรวจ

The screenshot shows a software window titled "ReviewPlan : Form" with a subtitle "Review Plan (TMM Level 4 Goal 1)". The form contains the following fields and values:

- SoftwareID: 1
- Name: Library Automation System
- Version: 1.0
- Effort unit: Man:Hours
- Lines Of Code unit: KLOC
- Review ID: 1
- Last updated date: 16-May-99
- Description: Review requirement, pages 1-20
- Work Product Type: Requirement Docur
- Planned Reviewed Quantity: 20 Pages
- Planned reviewed date: 10-Jun-99
- Planned Number Of Reviewers: 5
- Planned Preparation Time: 2 Hours
- Planned InspectionTime: 2 Hours

At the bottom of the form, there are six buttons: Add, Edit, Save, Delete, Cancel, and Exit.

รูปที่ 4.66 ตัวอย่างหน้าจอแผนงานการทบทวนในระดับ 4

จากมาตรวัด 3.3.1.1 ในบทที่ 3 ต้องเก็บข้อมูลข้อบกพร่องที่พบโดยการทบทวน และจากมาตรวัด 3.3.1.2 ต้องเก็บข้อมูลการทบทวนโดยเทียบกับแผนงานการทบทวน ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลการทบทวนดังรูปที่ 4.67 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขการทบทวนตามแผนงานการทบทวน วันที่ทำการทบทวน ปริมาณที่ทำการทบทวน จำนวนคนที่ทำการทบทวน เวลาที่ใช้ในการเตรียมการ เวลาที่ใช้ในการตรวจ ผลลัพธ์ของการทบทวน และจำนวนข้อบกพร่องที่พบโดยการทบทวน(ถ้ามี)

ReviewDetail : Form

Review Detail (TMM Level 4 Goal 1)

SoftwareID: 1 Name: Library Automation System
 Version: 1.0 Effort unit: Man.Hours Lines Of Code unit: KLOC

Review ID: 1 Description: Review requirement, pages 1-20
 Work Product Type: Requirement Document
 Planned Date: 10-Jun-99 Actual Date: 12-Jul-99

Planned Quantity: 20 Actual Quantity: 20 Pages
 Planned Number Of Reviewers: 5 Actual Number Of Reviewers: 5
 Planned Preparation Time: 2 Actual Preparation Time: 2 Hours
 Planned Inspection Time: 2 Actual Inspection Time: 2 Hours

Result: Modify Number of defects: 2

Note:

Buttons: Add, Edit, Save, Delete, Cancel, Exit, Previous, Next

รูปที่ 4.67 ตัวอย่างหน้าจอรายละเอียดการทบทวนในระดับ 4

รูปที่ 4.68 เป็นตัวอย่างหน้าจอป้อนข้อมูลรายละเอียดข้อบกพร่องในระดับ 4 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขข้อบกพร่องที่ต้องการเพิ่มเติมหรือแก้ไข รายละเอียดเกี่ยวกับข้อบกพร่อง วันที่พบและแก้ไขข้อบกพร่องเสร็จ เทคนิคที่ใช้ในการตรวจพบ หมายเลขโมดูลที่พบข้อบกพร่อง ขั้นตอนที่พบและที่เกิดของข้อบกพร่อง ความรุนแรงและประเภทข้อบกพร่อง (ข้อมูลสำหรับเป้าหมายการจัดตั้งโปรแกรมการวัดการทดสอบ ซึ่งจะกล่าวต่อไป) และความพยายามในการแก้ไขข้อบกพร่องของบุคคลต่างๆ

DefectDetail Level 4 : Form

Defect Detail (TMM Level 4 Goal 1 and 2)

SoftwareID: 1 Name: Library Automation System
 Version: 1.0 Effort unit: Man.Hours Lines Of Code unit: KLOC

DefectID: 1
 Defect detail: Calculate returned date incorrectly
 Found date: 03-Aug-99 Closed date: 05-Aug-99
 Testing techniques: Black Box techniq Module ID Found: 1
 Found phase: Design Origin phase: Requireme
 Defect severity: 3 Defect type: Incomplete or erroneous specific

DefectDebuggingEffort
 Total Debugging Effort: 7

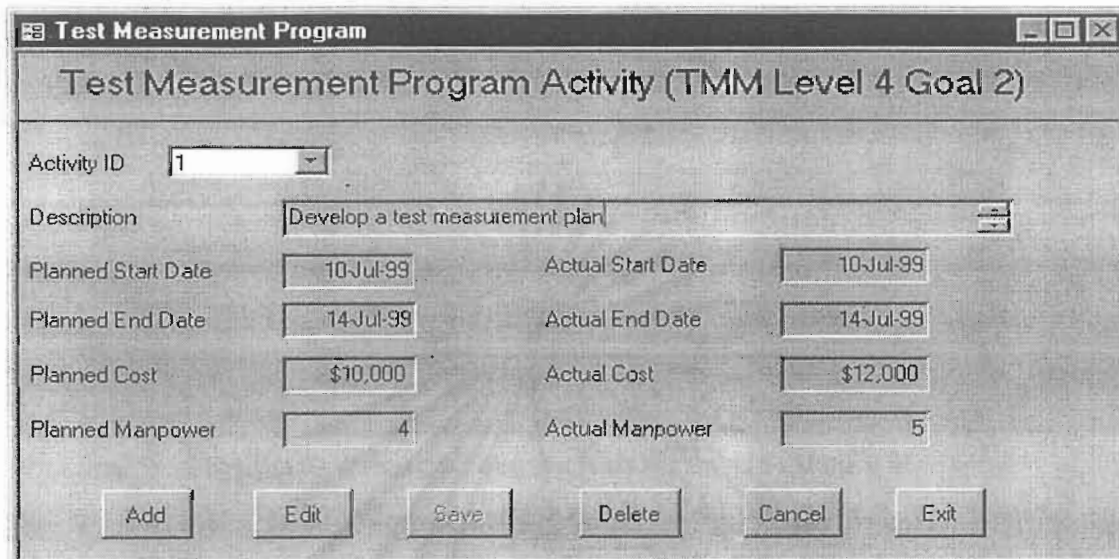
PersonID	Name	Surname	Date	DebuggingEffort
0001	Anuchit	Tang	05-Aug-99	4
0002	Pongpan	Bee	05-Aug-99	3

Buttons: Add, Edit, Delete, Undo, Record, Add, Edit, Save, Delete, Cancel, Exit

รูปที่ 4.68 ตัวอย่างหน้าจอรายละเอียดข้อบกพร่องในระดับ 4

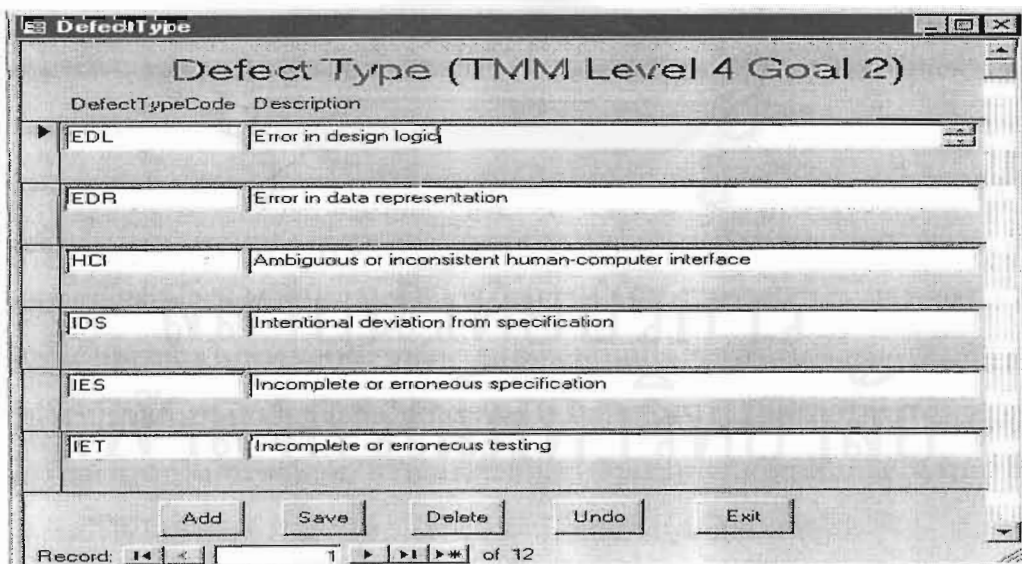
2) เป้าหมายการจัดตั้งโปรแกรมการวัดการทดสอบ

จากมาตรวัด 3.3.2.2 ในบทที่ 3 ต้องเก็บข้อมูลกิจกรรมในการจัดตั้งโปรแกรมการวัดการทดสอบ ได้แก่ วันเริ่มต้น และสิ้นสุดกิจกรรม ค่าใช้จ่ายที่เกิดขึ้นในการทำกิจกรรม และจำนวนคนที่ทำกิจกรรม เทียบกับแผนงาน ดังนั้นจึงออกแบบหน้าจอ ดังรูปที่ 4.69 สำหรับป้อนข้อมูลดังกล่าว



รูปที่ 4.69 ตัวอย่างหน้าจอกิจกรรมโปรแกรมการวัดการทดสอบในระดับ 4

จากมาตรวัด 3.3.2.1 ในบทที่ 3 ต้องเก็บข้อมูลประเภทและระดับความรุนแรงของข้อบกพร่อง องค์กรต้องกำหนดระดับความรุนแรงและประเภทของข้อบกพร่อง ดังตัวอย่างหน้าจอ สำหรับป้อนข้อมูลรูปที่ 4.70 ผู้ใช้ป้อนรหัสและรายละเอียดประเภทข้อบกพร่อง



รูปที่ 4.70 ตัวอย่างหน้าจอประเภทข้อบกพร่องในระดับ 4

รูปที่ 4.71 เป็นตัวอย่างหน้าจอในการป้อนข้อมูลความรุนแรงของข้อบกพร่อง ผู้ใช้ป้อนระดับและรายละเอียดความรุนแรงของข้อบกพร่อง

The screenshot shows a window titled "DefectSeverity (TMM Level 4 Goal 2)". It contains a table with the following data:

DefectSeverity	Description
1	Most severity
2	Critical
3	Moderate
4	Minor
5	Least severity

Below the table are buttons for "Add", "Save", "Delete", "Undo", and "Exit". At the bottom, there is a record indicator showing "Record: 1 of 5".

รูปที่ 4.71 ตัวอย่างหน้าจอความรุนแรงข้อบกพร่องในระดับ 4

สำหรับข้อมูลข้อบกพร่องที่ต้องป้อนเข้าไปมีหน้าจอเดียวกับหน้าจอในรูปที่ 4.68

3) เป้าหมายการประเมินคุณภาพซอฟต์แวร์

จากมาตรวัด 3.3.3.1 ในบทที่ 3 ต้องเก็บข้อมูลในการประเมินคุณสมบัติเชิงคุณภาพของซอฟต์แวร์ หน้าจอสำหรับป้อนข้อมูลเป็นดังรูปที่ 4.72 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนวันที่วัดคุณภาพ ป้อนจำนวนข้อบกพร่องเกี่ยวกับความต้องการและจำนวนบรรทัดคำสั่ง โปรแกรมจะคำนวณหาค่าคุณสมบัติความถูกต้องให้ ป้อนการใช้ประโยชน์จริงและการใช้ประโยชน์ที่จัดสรร โปรแกรมจะคำนวณหาค่าคุณสมบัติประสิทธิภาพให้ ป้อนจำนวนข้อบกพร่องเกี่ยวกับความปลอดภัยและจำนวนบรรทัดคำสั่ง โปรแกรมจะคำนวณหาค่าคุณสมบัติความเที่ยงตรงให้ ป้อนระยะเวลาที่ผู้ใช้ใช้ในการปฏิบัติงานเทียบกับเกณฑ์แสดงถึงคุณสมบัติความสามารถในการใช้งาน ป้อนระยะเวลาโดยเฉลี่ยที่ใช้ในการบำรุงรักษาแสดงถึงคุณสมบัติความสามารถในการบำรุงรักษา ป้อนระยะเวลาโดยเฉลี่ยที่ใช้ในการเปลี่ยนแปลงแสดงถึงคุณสมบัติความยืดหยุ่น ป้อนความพยายามที่ใช้ในการทดสอบ ความพยายามที่ใช้ในการย้ายไปทำงานกับระบบอื่น ความพยายามที่ใช้ในการปรับเปลี่ยน ความพยายามที่ใช้ในการทำงานร่วมกับระบบอื่นและความพยายามที่ใช้ในการพัฒนา โปรแกรมจะคำนวณหาค่าคุณสมบัติความสามารถในการทดสอบ ความสามารถในการย้ายไปทำงานกับระบบอื่น ความสามารถในการนำกลับมาใช้ใหม่ และความสามารถในการทำงานร่วมกับระบบอื่นตามลำดับ

SoftwareQualityAttributes : Form

Software Quality Attributes (TMM Level 4 Goal 3)

SoftwareID: 1 Name: Library Automation System
 Version: 1.0 Effort unit: Man.Hours Lines Of Code unit: KLOC

Date: 01-Nov-99

Correctness	#Defects(about requirement)	1	Lines Of Code	100	0.01	#defects/KLOC
Efficiency	Actual utilization	60	Allocated utilization	80	0.75	
Integrity	#Defects(about security)	2	Lines Of Code	100	0.02	
Usability			Average labor hours to use		2	Hours
Maintainability			Average labor days to fix		1	Days
Flexibility			Average labor days to change		1	Days
Testability	Effort to test	20	Effort to develop	100	0.20	
Portability	Effort to transport	10	Effort to develop	100	0.10	
Reusability	Effort to convert	29	Effort to develop	100	0.29	
Interoperability	Effort to couple	16	Effort to develop	100	0.16	

Previous Next

Add Edit Save Delete Cancel Exit

รูปที่ 4.72 ตัวอย่างหน้าจอกฎสมบัติด้านคุณภาพของซอฟต์แวร์ในระดับ 4

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สำหรับรายงานต่างๆ ในระดับ 4 มีดังต่อไปนี้

1) รายงานการทบทวน

- รายงานประสิทธิผลของการทบทวน

วัตถุประสงค์ เพื่อติดตามและประเมินประสิทธิผลของการทบทวน ดังตัวอย่างรายงานในรูปที่ 4.73 แสดงให้เห็นถึงจำนวนและเปอร์เซ็นต์ข้อบกพร่องที่ตรวจพบโดยการทบทวน การทดสอบและเมื่อนำซอฟต์แวร์ไปใช้งาน การกระจายของข้อบกพร่องจะชี้ให้เห็นว่ากระบวนการขั้นตอนที่มีประสิทธิผลต่ำ เช่นถ้าเปอร์เซ็นต์ข้อบกพร่องที่พบโดยการทบทวนมีค่าต่ำเมื่อเทียบกับการทดสอบ แสดงว่าการทบทวนมีประสิทธิผลต่ำ องค์กรจะต้องหาทางปรับปรุง เป็นต้น นอกจากนี้ องค์กรยังสามารถใช้รายงานในระดับ 3 ในรูปที่ 4.47 รายงานสรุปจำนวนและสัดส่วนข้อบกพร่องของโครงการ ที่ถูกพบ เกิดขึ้น หนีออก และประสิทธิผลของการจัดข้อบกพร่องในแต่ละขั้นตอน ในการประเมินประสิทธิผลของการทบทวน

Review Effectiveness Report (TMM Level 4 Goal 1)		
Date: 4/9/99		
Software ID: LAS	Name: Library Automation System	
Version : 1.0	Effort Unit: Man.Hours	
Reviewed Work Products	#Defects	%
Requirement	60	10.26
Design	90	15.38
Code	50	8.55
Unit Test Document	20	3.42
Integration Test Document	15	2.56
System Test Document	20	3.42
Acceptance Test Document	10	1.71
Total defects detected by reviews	265	45.30
Test Type	#Defects	%
Unit Test	90	15.38
Integration Test	90	15.38
System Test	60	10.26
Acceptance Test	30	5.13
Total defects detected by testing	270	46.15
Defects found at field	50	8.55
Total defects	585	100.00

รูปที่ 4.73 ตัวอย่างรายงานประสิทธิผลของการทบทวน

Review Detail Report (TMM Level 4 Goal 1)

Date: 4/9/99

Software ID: LAS

Name: Library Automation System

Version : 1.0

Effort Unit: Man.Hours

Review ID	Reviewed	Unit	Date			#Reviewers			Quan. Reviewed				Prep. Hours			Insp. Hours			Total Hours			Result	Defects
			Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	Var	%Cov	Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	Var		
1	Requirement	Pages	1/4/99	1/4/99	0	6	6	0	15	20	-5	7.50	4	4	0	4	4	0	8	8	0	M	8
2	Requirement	Pages	4/4/99	4/4/99	0	6	6	0	25	20	5	12.50	4	4	0	4	4	0	8	8	0	M	10
.
.
Total Requirement Reviews : 10						50	60	-10	180	200	-20	90.00	50	40	10	44	40	4	94	80	14		60
11	Design	Pages	1/5/99	1/5/99	0	4	5	-1	20	20	0	10.00	4	4	0	4	4	0	8	8	0	A	
.
.
Total Design Reviews					
.
Total Code Reviews					

A = Accept without further modification M = Accept with further modification R = Reject

รูปที่ 4.74 ตัวอย่างรายงานรายละเอียดการทบทวน

- รายงานรายละเอียดและสรุปการทบทวน

วัตถุประสงค์ เพื่อติดตามความก้าวหน้าของการทบทวน ดังตัวอย่างรายงานในรูปที่ 4.74 แสดงให้เห็นถึงรายงานรายละเอียดสถานะของการทบทวนเมื่อเทียบกับแผนงานคือ วันที่ทำการทบทวน จำนวนผู้ทำการทบทวน ปริมาณของผลผลิตที่นำมาทบทวน ความครอบคลุมการทบทวนเวลาเตรียมการ เวลาในการตรวจ ผลลัพธ์ของการทบทวนและจำนวนข้อบกพร่องที่ตรวจพบโดยการทบทวน

รายงานสรุปการทบทวนดังตัวอย่างรายงานในรูปที่ 4.75 แสดงให้เห็นถึงสรุปการทบทวนของผลผลิตแต่ละชนิด ปริมาณของผลผลิตทั้งหมด ปริมาณที่ได้ทำการทบทวนไปแล้ว เปอร์เซ็นต์ความครอบคลุมการทดสอบ เวลาที่ใช้ไปในการทบทวนที่ประกอบไปด้วย เวลาในการเตรียมการและเวลาในการตรวจสอบ อัตราการทบทวนซึ่งองค์การอาจจะนำไปใช้ในการประมาณเวลาในการทบทวนสำหรับโครงการอื่น หรือใช้ในการตั้งเป็นค่ามาตรฐานในการทบทวนขององค์กร เพราะอัตราการทบทวนอาจจะมีผลต่อประสิทธิภาพของการทบทวน คือถ้าอัตราการทบทวนเร็วไป ข้อบกพร่องอาจจะถูกพบน้อยลง แต่ก็เสียค่าใช้จ่ายในการทบทวนค่า ถ้าอัตราการทบทวนช้าไป ค่าใช้จ่ายในการทบทวนก็สูง จำนวนข้อบกพร่องก็อาจจะไม่พบมากขึ้นอีกเท่าใดนัก

Review Summary Report (TMM Level 4 Goal 1)											
Date: 4/9/99											
Software ID: LAS						Name: Library Automation System					
Version : 1.0						Effort Unit: Man.Hours					
Note : Rate = Quantity/Hours											
Reviewed Work	Unit	Quantity		Prep Hours	Insp Hours	Total Hours		#Defects			
Products		Total	To Date	%Cov To Date	Rate To Date	Rate To Date	Rate To Date	Rate To Date	Rate To Date	Rate To Date	Rate To Date
Requirement	Pages	200	180	90.00	50	3.60	44	4.1	94	2.13	60
Design	Pages	400	300	75.00	90	3.33	100	3.0	190	1.58	90
Code	KLOC	100	80	80.00	120	0.67	120	0.67	240	0.33	50
Unit Test Document	Pages	300	250	83.33	60	4.17	70	3.57	130	1.92	20
Integration Test Document	Pages	150	100	66.67	35	2.86	40	2.50	75	1.33	15
System Test Document	Pages	100	70	70.00	20	3.50	20	3.50	40	1.75	20
Acceptance Test Document	Pages	100	80	80.00	25	3.20	30	2.67	55	1.45	10

รูปที่ 4.75 ตัวอย่างรายงานสรุปการทบทวน

2) รายงานการจัดตั้งโปรแกรมการวัดการทดสอบ

- รายงานสถานะโปรแกรมการวัดการทดสอบ

วัตถุประสงค์ เพื่อติดตามสถานะของกิจกรรมการวัดการทดสอบเมื่อเทียบกับแผนงาน ดังตัวอย่างรายงานในรูปที่ 4.76 แสดงให้เห็นถึงวันเริ่มกิจกรรม วันสิ้นสุดกิจกรรม จำนวนคน และค่าใช้จ่ายที่ใช้ในการทำกิจกรรมจริงเทียบกับแผนงาน

Test Measurement Activity Status (TMM Level 4 Goal 2)														
Date: 1/9/99														
Activity ID	Detail	Start Date			End Date			Manpower			Cost			
		Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	Var	%
1	Define test measurement goal	1/7/99	1/7/99	0	5/7/99	5/7/99	0	5	5	0	100	100	0	0.00
2	Define test-related data to be collected	8/7/99	7/7/99	1	15/7/99	14/7/99	1	5	5	0	160	150	10	6.67
.
.

รูปที่ 4.76 ตัวอย่างรายงานสถานะโปรแกรมการวัดการทดสอบ

- รายงานรายละเอียดข้อบกพร่อง

วัตถุประสงค์ เพื่อใช้ในการวิเคราะห์ข้อบกพร่องอันจะเป็นประโยชน์ในการหาทางลดข้อบกพร่อง

รายงานข้อบกพร่องตามประเภทของข้อบกพร่องดังตัวอย่างรายงานในรูปที่ 4.77 แสดงให้เห็นถึงการกระจายของข้อบกพร่องเรียงลำดับจากมากที่สุดไปน้อยสุด โดยการมุ่งเน้นไปที่ประเภทข้อบกพร่องที่มีจำนวนมากสุด 2-3 อันดับแรก จะเป็นข้อมูลที่ใช้ในการป้องกันข้อบกพร่องในระดับ 5 สำหรับรูปที่ 4.78 เป็นตัวอย่างรายงานสรุประดับความรุนแรงของข้อบกพร่อง ว่าในแต่ละระดับมีจำนวนข้อบกพร่องเท่าใด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Defect Type Report (TMM Level 4 Goal 2)

Date: 1/9/99

Software ID: **LAS**

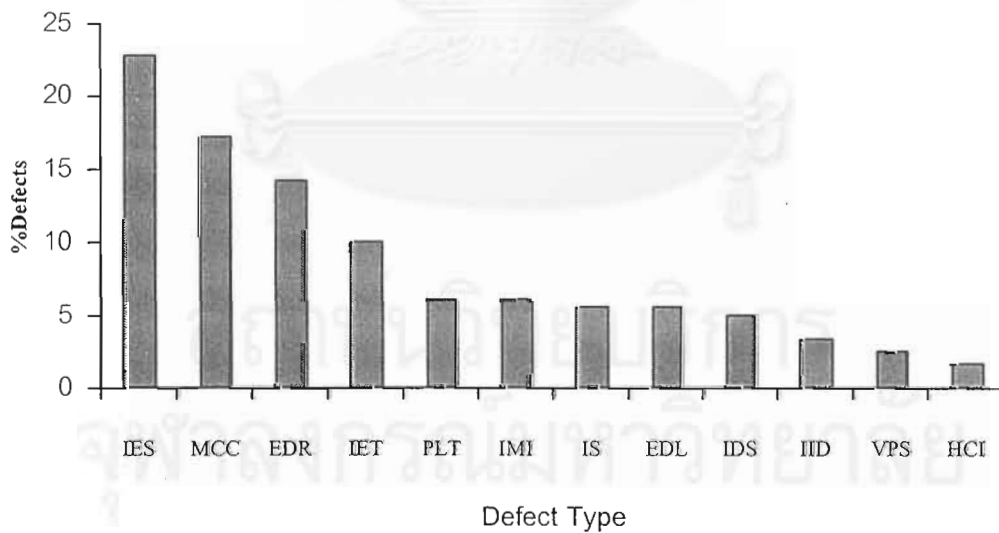
Name: **Library Automation System**

Version : **1.0**

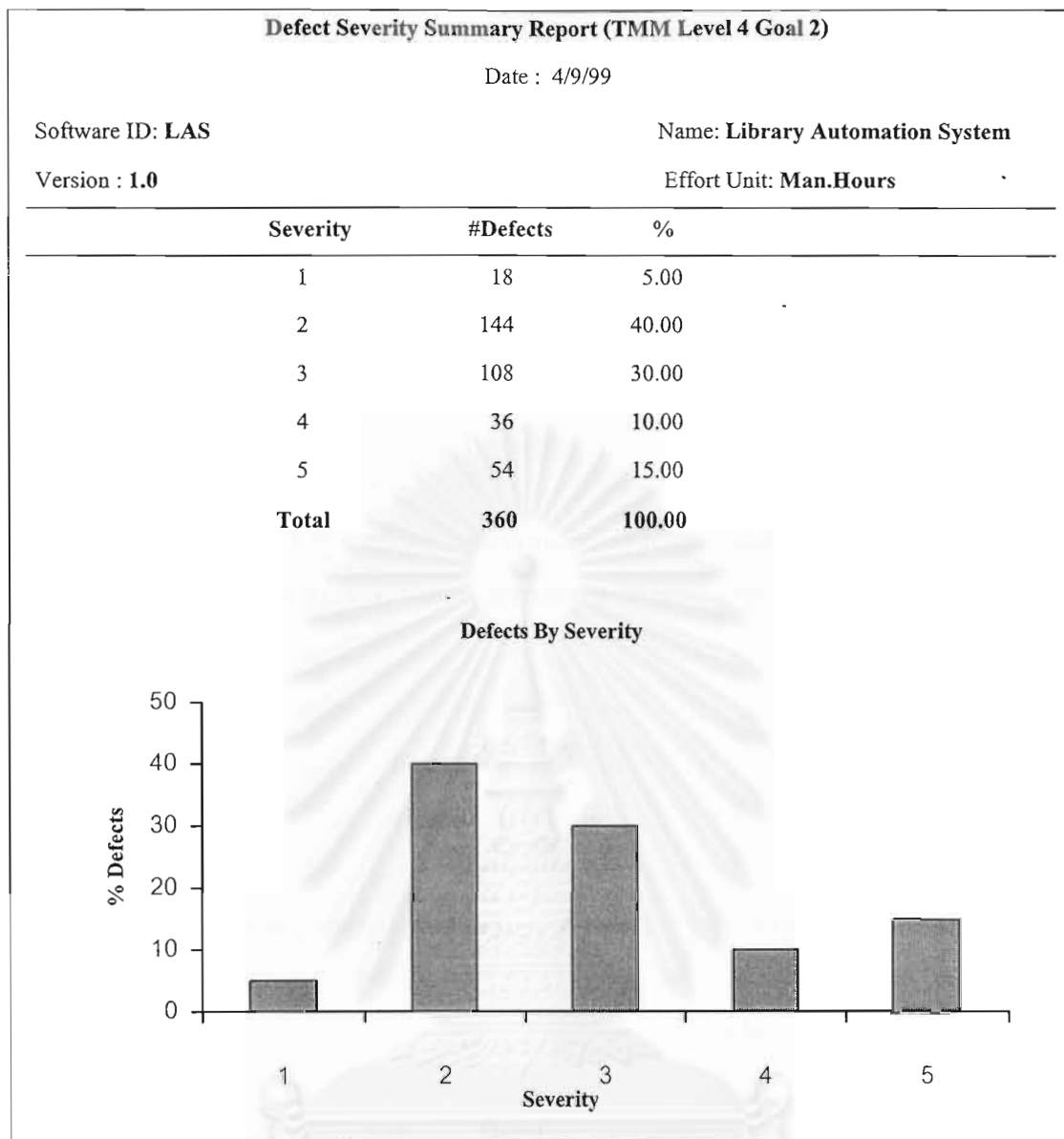
Effort Unit: **Man.Hours**

Defect Type	#Defects	%
IES Incomplete or erroneous specification	82	22.78
MCC Misinterpretation of customer communication	62	17.22
EDR Error in data representation	51	14.17
IET Incomplete or erroneous testing	36	10.00
PLT Error in programming language translation of design	22	6.11
IMI Inconsistent module interface	22	6.11
MIS Miscellaneous	20	5.56
EDL Error in design logic	20	5.56
IDS Intentional deviation from specification	18	5.00
IID Inaccurate or incomplete documentation	12	3.33
VPS Violation of programming standards	9	2.50
HCI Ambiguous or inconsistent human-computer interface	6	1.66
Total	360	100.00

Defects Type Graph



รูปที่ 4.77 ตัวอย่างรายงานจำนวนข้อบกพร่องตามประเภทของข้อบกพร่อง



รูปที่ 4.78 ตัวอย่างรายงานสรุปจำนวนข้อบกพร่องตามระดับความรุนแรง

- รายงานผลผลิตของผู้ทดสอบ

วัตถุประสงค์ เพื่อใช้ในการประเมินผลการทำงานของผู้ทดสอบ ดังตัวอย่างรายงานในรูปที่ 4.79 แสดงให้เห็นถึงผลผลิตของผู้ทดสอบในแต่ละสัปดาห์ของผู้ทดสอบว่าทำการทดสอบซอฟต์แวร์ได้ปริมาณเท่าใด ใ้ความพยายามเท่าใด จำนวนข้อบกพร่องที่ตรวจพบ ค่าใช้จ่ายในรูปของความพยายามที่ใช้ไปในการตรวจพบข้อบกพร่องต่อหนึ่งข้อบกพร่อง องค์กรอาจจะตั้งเป็นค่ามาตรฐานในการปฏิบัติงานของผู้ทดสอบ

Testers' Productivity Detail Report (TMM Level 4 Goal 2)					
Date : 4/9/99					
Software ID: LAS			Name: Library Automation System		
Version : 1.0		Lines Of Code Unit : KLOC		Effort Unit: Man.Hours	
Week	LOC Tested	Testing Effort	Productivity (LOC/Testing Effort)	#Defects	Defect Detection Cost (Testing Effort/#Defects)
26	8	40	0.20	40	1
27	12	50	0.24	10	5
.
.

รูปที่ 4.79 ตัวอย่างรายงานรายละเอียดผลผลิตของผู้ทดสอบ

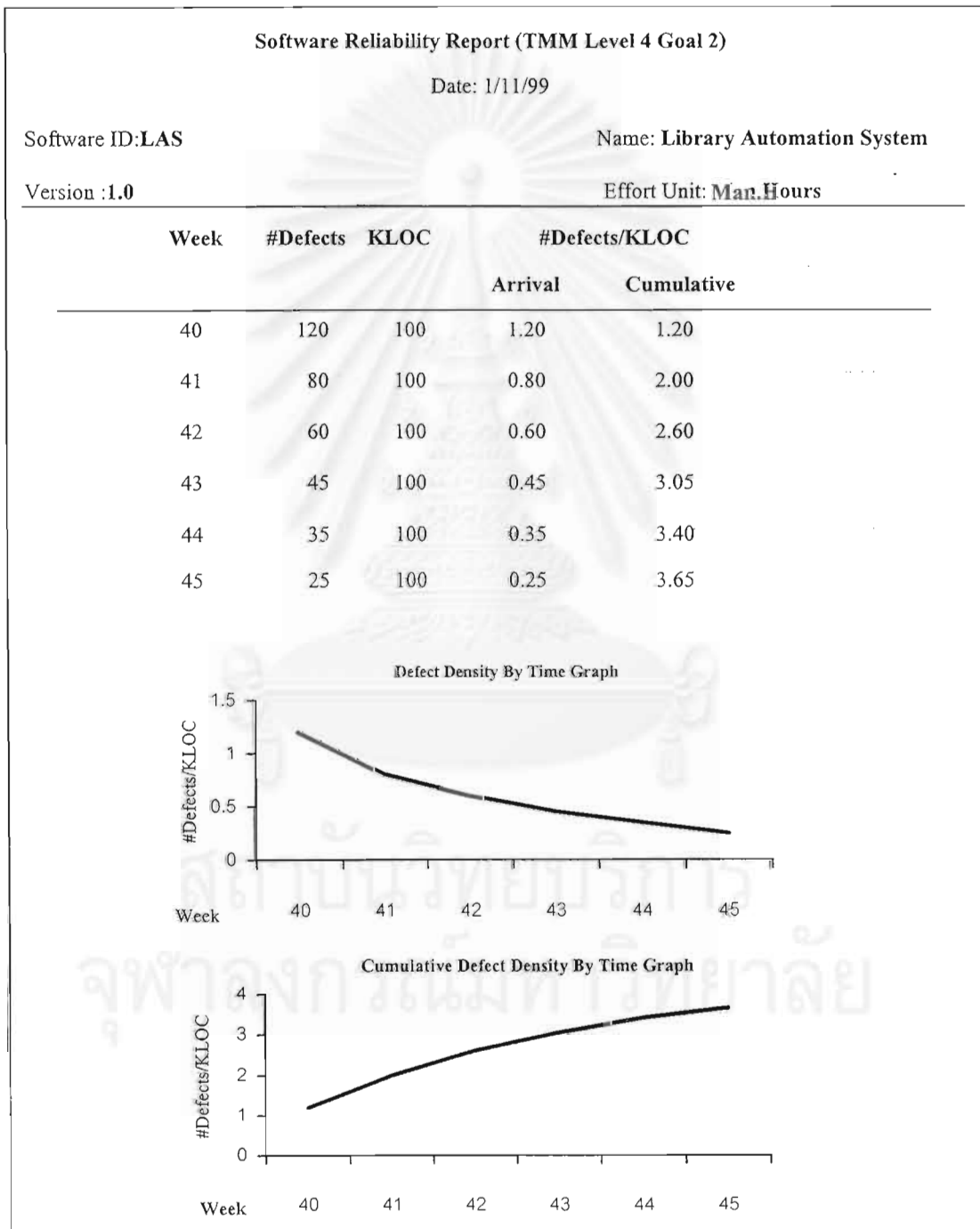
รายงานรูปที่ 4.80 เป็นตัวอย่างรายงานสรุปผลผลิตของผู้ทดสอบว่า สามารถพัฒนาแผนงานการทดสอบได้ปริมาณเท่าใดต่อความพยายาม สามารถพัฒนากรณีทดสอบได้ปริมาณเท่าใดต่อความพยายาม สามารถพัฒนาขั้นตอนปฏิบัติการทดสอบได้ปริมาณเท่าใดต่อความพยายาม และสามารถทดสอบซอฟต์แวร์ได้ปริมาณเท่าใดต่อความพยายามที่ใช้ไปในการทดสอบโมดูล การทดสอบเบ็ดเสร็จ การทดสอบระบบ และการทดสอบเพื่อยอมรับ

Testers' Productivity Summary Report (TMM Level 4 Goal 2)					
Date : 4/9/99					
Software ID: LAS			Name: Library Automation System		
Version : 1.0		Lines Of Code Unit : KLOC		Effort Unit: Man.Hours	
Type	Unit	Quantity	Effort	%	Productivity (Quantity/Effort)
Test Plan Development	Pages	650	330	8.79	1.97
Test Case Development	#Test Cases	400	600	15.97	0.67
Test Procedure Development	#TestProcedures	200	500	13.31	0.40
Unit Test	KLOC	100	526	14.00	0.19
Integration Test	KLOC	100	600	15.97	0.17
System Test	KLOC	100	700	18.64	0.14
Acceptance Test	KLOC	100	500	13.31	0.20
Total			3756	100.00	

รูปที่ 4.80 ตัวอย่างรายงานสรุปผลผลิตของผู้ทดสอบ

- รายงานความน่าเชื่อถือของซอฟต์แวร์

วัตถุประสงค์ เพื่อใช้ประเมินคุณสมบัติด้านความน่าเชื่อถือของซอฟต์แวร์ ดังตัวอย่างรายงานในรูปแบบที่ 4.81 แสดงให้เห็นถึงจำนวนข้อบกพร่องที่เกิดขึ้นตามระยะเวลาตั้งแต่การทดสอบระบบ ถ้าซอฟต์แวร์มีความน่าเชื่อถือที่ดี จำนวนข้อบกพร่องก็จะลดจำนวนลงไปอย่างรวดเร็ว กราฟจะลดลงเป็นเอ็กซ์โปเนนเชียล ถ้าดูจากกราฟจำนวนข้อบกพร่องสะสม กราฟจะมีความชันลดลงเรื่อยๆ ซึ่งจะมีประโยชน์ในการตัดสินใจได้ว่าควรจะหยุดการทดสอบและนำซอฟต์แวร์ไปใช้งานจริงได้หรือยัง



รูปที่ 4.81 ตัวอย่างรายงานความน่าเชื่อถือของซอฟต์แวร์

3) รายงานคุณสมบัติด้านคุณภาพของซอฟต์แวร์

วัตถุประสงค์ เพื่อใช้ประเมินคุณภาพของซอฟต์แวร์ ดังตัวอย่างรายงานในรูปที่

4.82 แสดงให้เห็นถึงค่าคุณสมบัติต่างๆ ของซอฟต์แวร์

Software Quality Attribute Report (TMM Level 4 Goal 3)			
Date: 1/11/99			
Software ID: LAS		Name: Library Automation System	
Version : 1.0		Effort Unit: Man.Hours	
Quality Attribute	Metric		Unit
Correctness	#defects(about requirement)/LOC	3.60	#defects/KLOC
Efficiency	actual utilization/allocated utilization	0.60	
Integrity	#defects(about security)/LOC	0.02	#defects/KLOC
Usability	average labor hours to use	2.00	hours
Maintainability	average labor days to fix	2.00	days
Flexibility	average labor days to change	3.00	days
Testability	effort to test/effort to develop	0.25	
Portability	effort to transport/effort to develop	0.10	
Reusability	effort to convert/effort to develop	0.10	
Interoperability	effort to couple/effort to develop	0.05	

รูปที่ 4.82 ตัวอย่างรายงานคุณสมบัติด้านคุณภาพของซอฟต์แวร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.5 หน้าจอและรายงานของมาตรวัดในระดับ 5

1) เป้าหมายการใช้ข้อมูลจากระบวนการในการป้องกันข้อบกพร่อง

จากมาตรวัด 3.4.1.1 ในบทที่ 3 ต้องเก็บข้อมูลเกี่ยวกับความพยายามในการหาสาเหตุรากของข้อบกพร่อง ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลจากการประชุมหาสาเหตุรากของข้อบกพร่อง ดังตัวอย่างหน้าจอในรูปที่ 4.83 ผู้ใช้ต้องป้อนหมายเลขการประชุมที่ต้องการเพิ่มหรือแก้ไข วันที่ทำการประชุม ระยะเวลาที่ใช้ในการประชุม จำนวนคนที่เข้าทำการประชุม ความพยายามที่ใช้ในการหาสาเหตุราก ตามแผนงานกับที่เกิดขึ้นจริง และจำนวนกิจกรรมที่จะทำการป้องกันข้อบกพร่องที่ได้จากการประชุม

Casual Meeting (TMM Level 5 Goal 1)

Causal Meeting ID	1/99	Actual Date	04-Jan-99
Planned Date	04-Jan-99	Actual Duration	2 Hours
Planned Duration	2 Hours	Actual Manpower	5
Planned Manpower	5	Actual Effort	10
Planned Effort	10	Number Of Actions	2
Number Of Actions	2	Note:	

Buttons: Add, Edit, Save, Delete, Cancel, Exit

รูปที่ 4.83 ตัวอย่างหน้าจอการประชุมหาสาเหตุรากของข้อบกพร่องในระดับ 5

องค์กรต้องกำหนดประเภทสาเหตุรากของข้อบกพร่อง หน้าจอสำหรับป้อนข้อมูลเป็นดังรูป 4.84 ในแต่ละสาเหตุ ผู้ใช้ป้อนรหัสและรายละเอียดของสาเหตุ

Defect Cause Category (TMM Level 5 Goal 1)

CauseCode	Description
COMM	Communication failure(incorrect or lost information)
EDUC	Education lacking(lacking knowledge or understanding)
OTHR	Others
OVER	Oversight or Overlooked(failed to consider all cases and conditions)
TRAN	Transcription error(transcribed or copied something incorrectly)
*	

Buttons: Add, Save, Delete, Cancel, Exit

Record: 1 of 5

รูปที่ 4.84 ตัวอย่างหน้าจอประเภทสาเหตุของข้อบกพร่องในระดับ 5

รูปที่ 4.85 เป็นตัวอย่างหน้าจอในการป้อนข้อมูลรายละเอียดข้อบกพร่องใน ระดับ 5 ผู้ใช้ต้องเลือกหมายเลขซอฟต์แวร์ ป้อนหมายเลขข้อบกพร่องที่ต้องการเพิ่มเติมหรือแก้ไข รายละเอียดข้อบกพร่อง วันที่พบข้อบกพร่อง วันที่แก้ไขข้อบกพร่องเสร็จ เทคนิคการทดสอบที่ใช้ ในการตรวจพบข้อบกพร่อง หมายเลขโมดูลที่พบข้อบกพร่อง ขั้นตอนที่พบข้อบกพร่อง ขั้นตอน ที่ข้อบกพร่องเกิดขึ้น ระดับความรุนแรงของข้อบกพร่อง ประเภทของข้อบกพร่อง และรหัสของ สาเหตุของข้อบกพร่อง

รูปที่ 4.85 ตัวอย่างหน้าจอรายละเอียดข้อบกพร่องในระดับ 5

จากมาตรวัด 3.4.1.1 ในบทที่ 3 ต้องเก็บข้อมูลของกิจกรรมป้องกันข้อบกพร่อง ได้แก่ วันเริ่มต้นและสิ้นสุดกิจกรรม และค่าใช้จ่ายในรูปตัวเงินหรือความพยายาม ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลดังรูปที่ 4.86 ผู้ใช้ป้อนหมายเลขกิจกรรมที่ต้องการเพิ่มหรือแก้ไข รายละเอียดกิจกรรม วันที่เริ่มต้นและเสร็จสิ้นการทำกิจกรรม ความพยายามในการทำกิจกรรมตาม แผน ค่าใช้จ่ายที่เกิดขึ้น ตามแผนงานกับที่เกิดขึ้นจริง ผลลัพธ์ของการทำกิจกรรมที่คาดหวังและที่ เกิดขึ้นจริง และความพยายามของบุคลากรที่ทำกิจกรรมจริง

รูปที่ 4.86 ตัวอย่างหน้าจอกิจกรรมป้องกันข้อบกพร่องในระดับ 5

2) เป้าหมายการควบคุมคุณภาพ

จากมาตรวัด 3.4.2.1 ในบทที่ 3 ต้องเก็บข้อมูลค่าขีดจำกัดบน และขีดจำกัดล่างของความหนาแน่นของข้อบกพร่อง อัตราการทบทวน อัตราความพยายามในการทดสอบ และความครอบคลุมการทดสอบ ได้แก่ ความครอบคลุมข้อความสั่ง ความครอบคลุมกึ่ง และความครอบคลุมทางเดินพื้นฐาน ตัวอย่างหน้าจอตั้งรูปที่ 4.87 เป็นหน้าจอสำหรับป้อนข้อมูลค่าขีดจำกัดล่างและขีดจำกัดบนของตัวแปรดังกล่าว

Parameter	Lower Control Limit (LCL)	Upper Control Limit (UCL)	Unit
Module Defect Density	1	10	Defects/KLOC
Requirement Review Rate	5	20	Pages/Hours
Design Review Rate	5	20	Pages/Hours
Code Review Rate	5	3	KLOC/Hours
Test Document Review Rate	5	20	Pages/Hours
Test Effort Rate	.5	5	KLOC/Man.Hours
Statement Coverage	70	100	%
Branch Coverage	50	100	%
Basis Path Coverage	50	100	%

รูปที่ 4.87 ตัวอย่างหน้าจอการควบคุมคุณภาพในระดับ 5

3) เป้าหมายการทำให้กระบวนการทดสอบดีที่สุด

จากมาตรวัด 3.4.3.1 ในบทที่ 3 ต้องเก็บข้อมูลความพยายามที่ใช้ไปในการประเมินและตัดสินใจในการนำเครื่องมือและเทคโนโลยีการทดสอบมาใช้งาน ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลดังรูปที่ 4.88 ผู้ใช้ป้อนหมายเลขเทคโนโลยีหรือเครื่องมือการทดสอบ รายละเอียด วันที่ทำการประเมิน ประเภทของเครื่องมือ ผลประโยชน์และค่าใช้จ่ายที่คาดไว้และที่เกิดขึ้นจริง ผลลัพธ์ของการประเมิน และความพยายามของบุคลากรที่ใช้ในการประเมิน

PersonID	Name	Surname	Date	Effort
0001	Anuchit	Tang	10-Jan-99	20
0005	Pornnapa		10-Jan-99	20

รูปที่ 4.88 ตัวอย่างหน้าจอการประเมินเทคโนโลยีหรือเครื่องมือการทดสอบ

จากมาตรวัด 3.4.3.1 ในบทที่ 3 ต้องเก็บข้อมูลความพยายามที่ใช้ไปในการกำหนดว่าเมื่อใดจะหยุดการทดสอบ ดังนั้นจึงออกแบบหน้าจอสำหรับป้อนข้อมูลดังรูปที่ 4.89 ผู้ใช้เลือกหมายเลขซอฟต์แวร์ และป้อนข้อมูลความพยายามของบุคลากรที่ใช้ไปในการกำหนดว่าเมื่อใดจะหยุดทดสอบซอฟต์แวร์

PersonID	Name	Surname	Date	Effort
0001	Anuchit	Tang	04-Jan-99	20
0002	Pongpan	Bae	04-Jan-99	20
0003	Sunisa		10-Jan-99	30

รูปที่ 4.89 ตัวอย่างหน้าจอความพยายามที่ใช้ในการกำหนดเมื่อใดจะหยุดทดสอบ

จากมาตรวัด 3.4.3.2 ในบทที่ 3 ต้องเก็บข้อมูลเกณฑ์ที่ใช้ในการหยุดการทดสอบคือ เป้าหมายความหนาแน่นของข้อบกพร่องตามระยะเวลา จึงออกแบบหน้าจอสำหรับป้อนข้อมูลดังรูปที่ 4.90 ผู้ใช้ป้อนสัปดาห์ตั้งแต่การทดสอบระบบ และความหนาแน่นของข้อบกพร่องตามเป้าหมายแต่ละสัปดาห์

Week	DefectDensityGoal
40	1.1
41	.6
42	.3
43	.2

รูปที่ 4.90 ตัวอย่างหน้าจอเกณฑ์ที่ใช้ในการกำหนดเมื่อใดจะหยุดทดสอบ

สำหรับรายงานต่างๆ ในระดับ 5 มีดังต่อไปนี้

1) รายงานการป้องกันข้อบกพร่อง

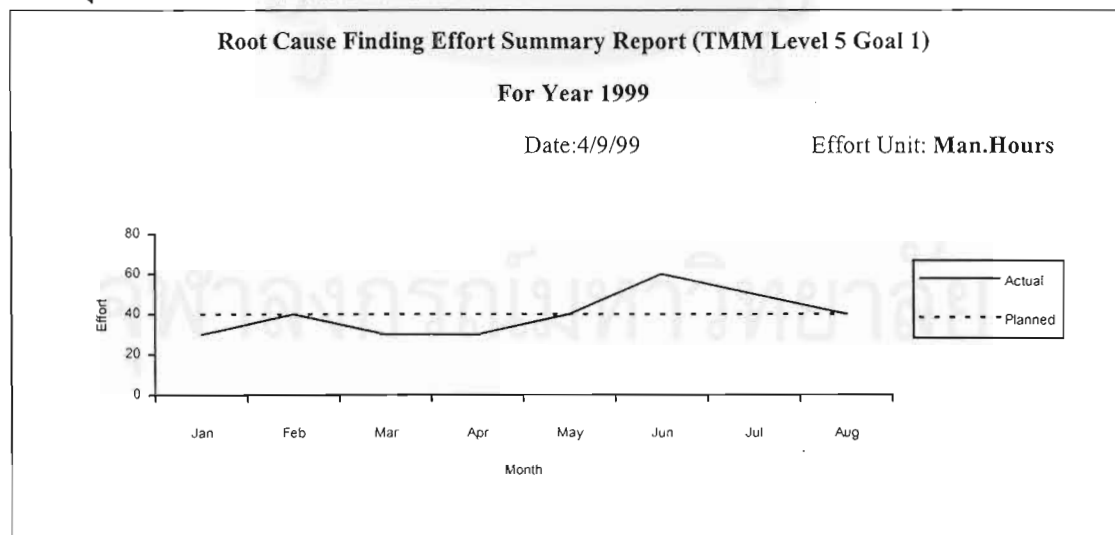
- รายงานรายละเอียดและสรุปในการหาสาเหตุรากของข้อบกพร่อง

วัตถุประสงค์ เพื่อใช้ติดตามสถานะในการหาสาเหตุรากของข้อบกพร่อง ดังตัวอย่างรายงานในรูปที่ 4.91 แสดงให้เห็นถึงรายละเอียดของการประชุมในการหาสาเหตุรากของข้อบกพร่องคือ วันที่ทำการประชุม ระยะเวลาในการประชุม จำนวนคนที่เข้าประชุม และความพยายามที่ใช้ไป เทียบกับแผนงาน

Root Cause Finding Status Detail Report (TMM Level 5 Goal 1)												
From 1/1/99 To 31/8/99												
Date: 4/9/99												
Effort unit : Man.Hours												
Casual Meeting ID	Date			Duration(Hours)			Manpower			Effort		
	Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	%Var
1	4/1/99	4/1/99	0	2	2	0	5	5	0	10	10	0.00
2	11/1/99	11/1/99	0	1	2	-1	4	5	-1	4	10	-60.00
.
.
Total Casual Meeting : 30				65	60	5	140	150	-10	290	320	-9.37

รูปที่ 4.91 ตัวอย่างรายงานสถานะในการหาสาเหตุรากของข้อบกพร่อง

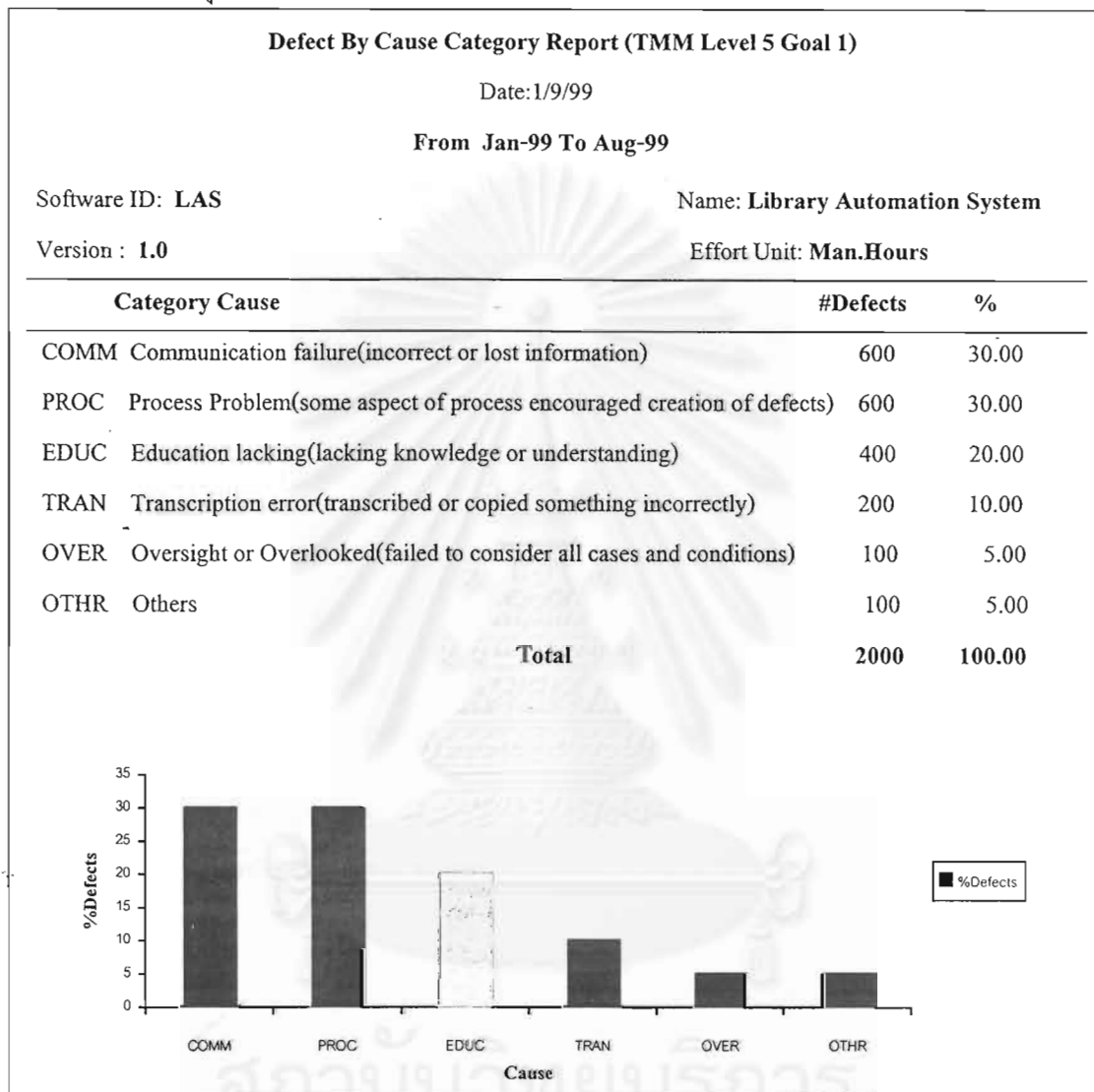
ตัวอย่างรายงานในรูปที่ 4.92 เป็นกราฟรายงานสรุปความพยายามในการหาสาเหตุรากของข้อบกพร่องเมื่อเทียบกับแผนงาน



รูปที่ 4.92 ตัวอย่างกราฟรายงานสรุปความพยายามในการหาสาเหตุรากของข้อบกพร่อง

- รายงานข้อบกพร่องตามประเภทของสาเหตุราก

วัตถุประสงค์ เพื่อใช้ในการวิเคราะห์สาเหตุที่ทำให้เกิดข้อบกพร่อง ที่จะช่วยในการหาวิธีในการป้องกันข้อบกพร่องไม่ให้เกิดขึ้นอีก ดังตัวอย่างรายงานในรูปที่ 4.93 แสดงให้เห็นถึงจำนวนข้อบกพร่องตามประเภทของสาเหตุ องค์กรจะต้องหาวิธีการในการลดสาเหตุที่ทำให้เกิดข้อบกพร่องมากที่สุด



รูปที่ 4.93 ตัวอย่างรายงานข้อบกพร่องตามประเภทของสาเหตุราก

นอกจากนี้รายงานในระดับ 3 คือรายงานข้อบกพร่องในแต่ละขั้นตอนและประสิทธิภาพของการจัดข้อบกพร่องในรูปที่ 4.47 จะช่วยในการวิเคราะห์ข้อบกพร่องและดูผลของการทำกิจกรรมป้องกันข้อบกพร่องได้ รวมถึงรายงานในระดับ 4 คือรายงานข้อบกพร่องตามประเภทดังรูปที่ 4.77 จะช่วยในการหาวิธีการป้องกันข้อบกพร่อง เช่น การติดตามประเภทของข้อบกพร่องใน 2-3 อันดับแรกว่ามีการลดจำนวนลงหรือไม่หลังจากทำกิจกรรมป้องกันข้อบกพร่อง

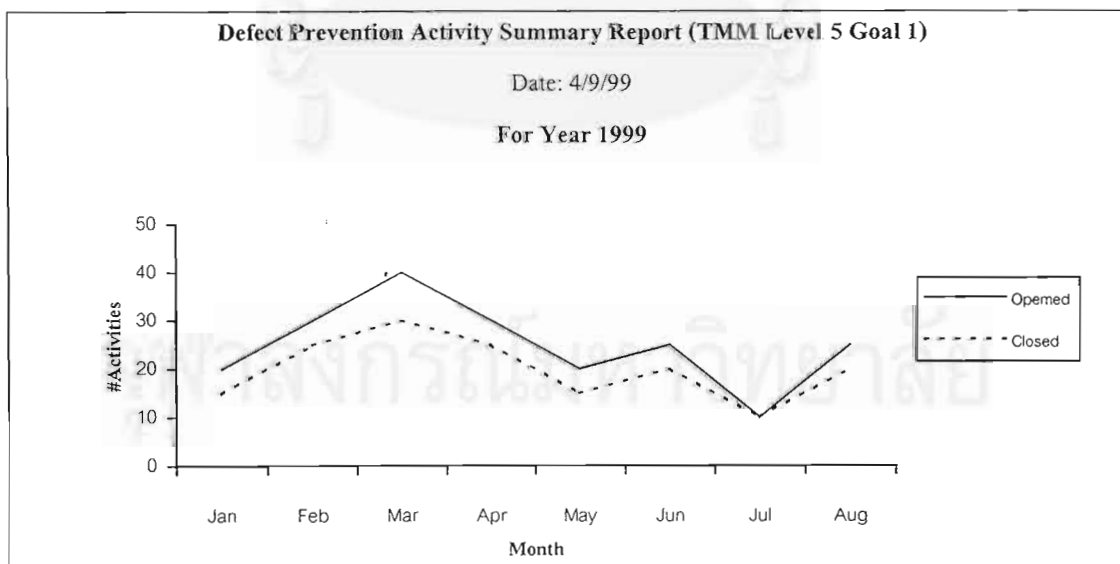
- รายงานรายละเอียดและสรุปกิจกรรมการป้องกันข้อบกพร่อง

วัตถุประสงค์ เพื่อติดตามสถานะกิจกรรมการป้องกันข้อบกพร่อง ดังตัวอย่างรายงานในรูปที่ 4.94 แสดงถึงรายละเอียดของกิจกรรมที่เพื่อขจัดสาเหตุที่ทำให้เกิดข้อบกพร่องเทียบกับแผนงานคือ วันเริ่มต้นและสิ้นสุด ความพยายามที่ใช้ในการทำกิจกรรม และค่าใช้จ่ายที่เกิดขึ้น

Defect Prevention Activity Report (TMM Level 5 Goal 1)													
Date:4/9/99													
Effort unit : Man.Hours													
Activity	Detail	Start Date			End Date			Effort			Cost		
ID		Act'l	Plan	Var	Act'l	Plan	Var	Act'l	Plan	%Var	Act'l	%Var	
1	Write a tool to trace registers during unit test	5/1/99	5/1/99	0	14/2/99	27/2/99	13	120	100	20.00			
2	Add item to common error to warn people that late changes are more prone error	5/1/99	5/1/99	0	10/1/99	10/1/99	0	12	10	20.00			
.	
.	
Total number of activity : 200								2500	1500	66.67			

รูปที่ 4.94 ตัวอย่างรายงานรายละเอียดสถานะกิจกรรมการป้องกันข้อบกพร่อง

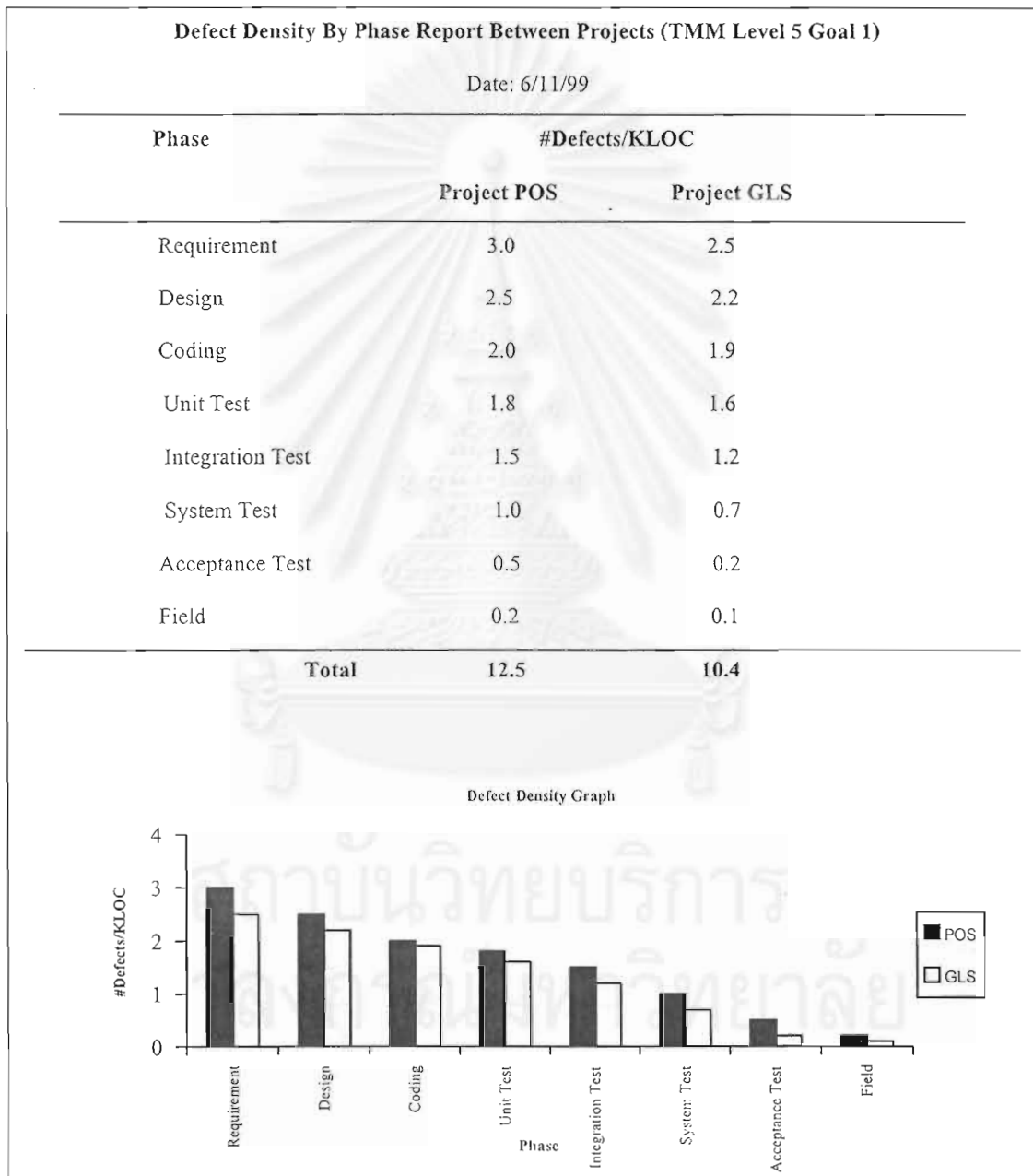
ตัวอย่างรายงานในรูปที่ 4.95 เป็นรายงานสรุปสถานะกิจกรรมการป้องกันข้อบกพร่อง ว่าในแต่ละเดือน มีจำนวนกิจกรรมใหม่เท่าใด และมีจำนวนกิจกรรมที่เสร็จสิ้นเท่าใด



รูปที่ 4.95 ตัวอย่างรายงานสรุปสถานะกิจกรรมการป้องกันข้อบกพร่อง

- รายงานเปรียบเทียบความหนาแน่นข้อบกพร่องระหว่างโครงการ

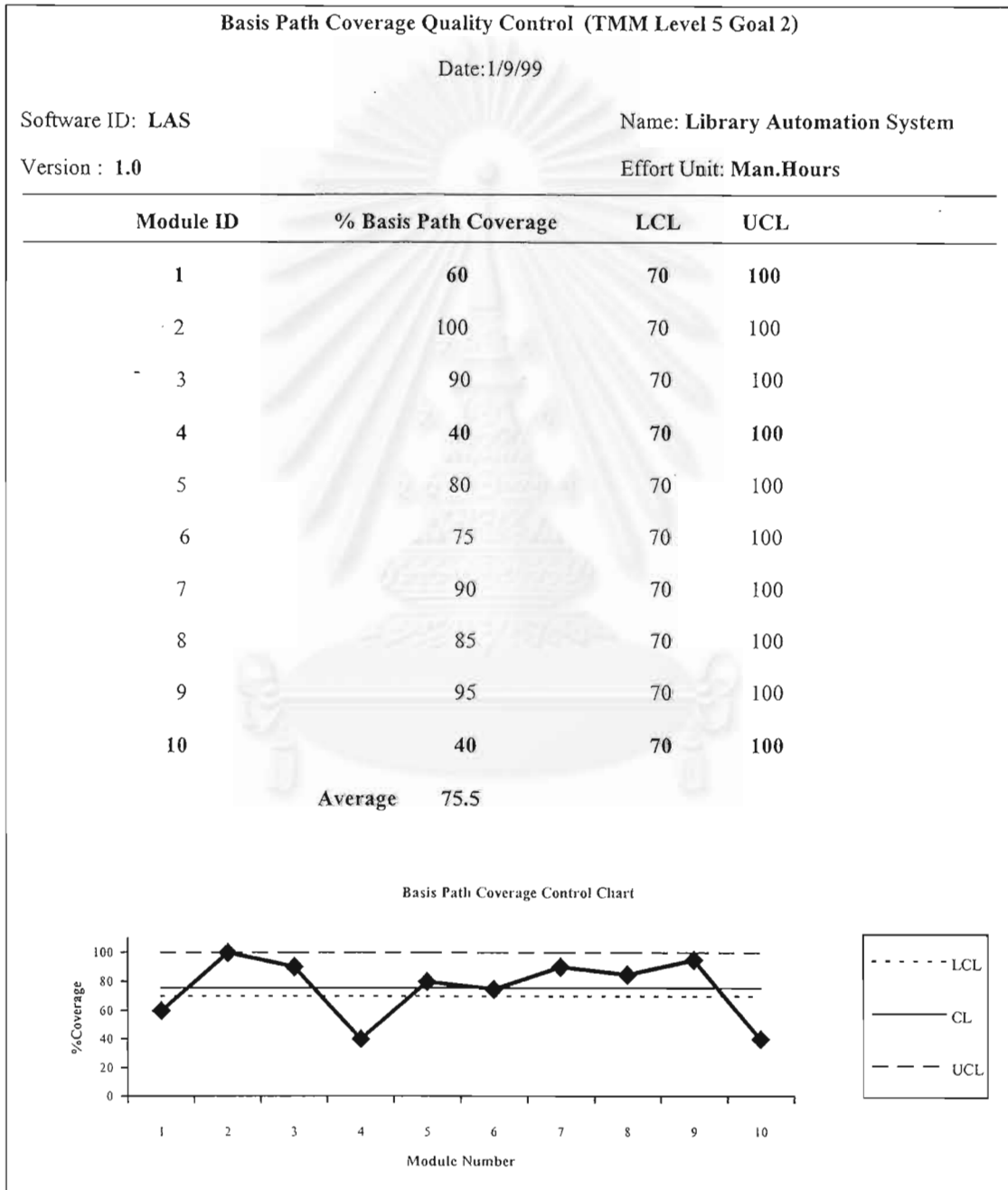
วัตถุประสงค์ เพื่อให้ผู้ผลของการทำกิจกรรมป้องกันข้อบกพร่องจะสามารถลดข้อบกพร่องลงได้มากน้อยเพียงใด ดังตัวอย่างรายงานในรูปที่ 4.96 แสดงให้เห็นถึง 2 โครงการ โดยที่โครงการ GLS เป็นโครงการที่มีการทำกิจกรรมป้องกันข้อบกพร่องเปรียบเทียบกับโครงการอื่นก่อนทำคือโครงการ POS ถ้าค่าความหนาแน่นข้อบกพร่องมีค่าลดลง ก็แสดงว่ากิจกรรมที่ทำไปได้ผล ถ้าไม่ก็ต้องหาสาเหตุว่าทำไมถึงไม่ได้ผล และจะปรับเปลี่ยนวิธีการในการป้องกันข้อบกพร่องได้อย่างไร



รูปที่ 4.96 ตัวอย่างรายงานเปรียบเทียบความหนาแน่นข้อบกพร่องระหว่างโครงการ

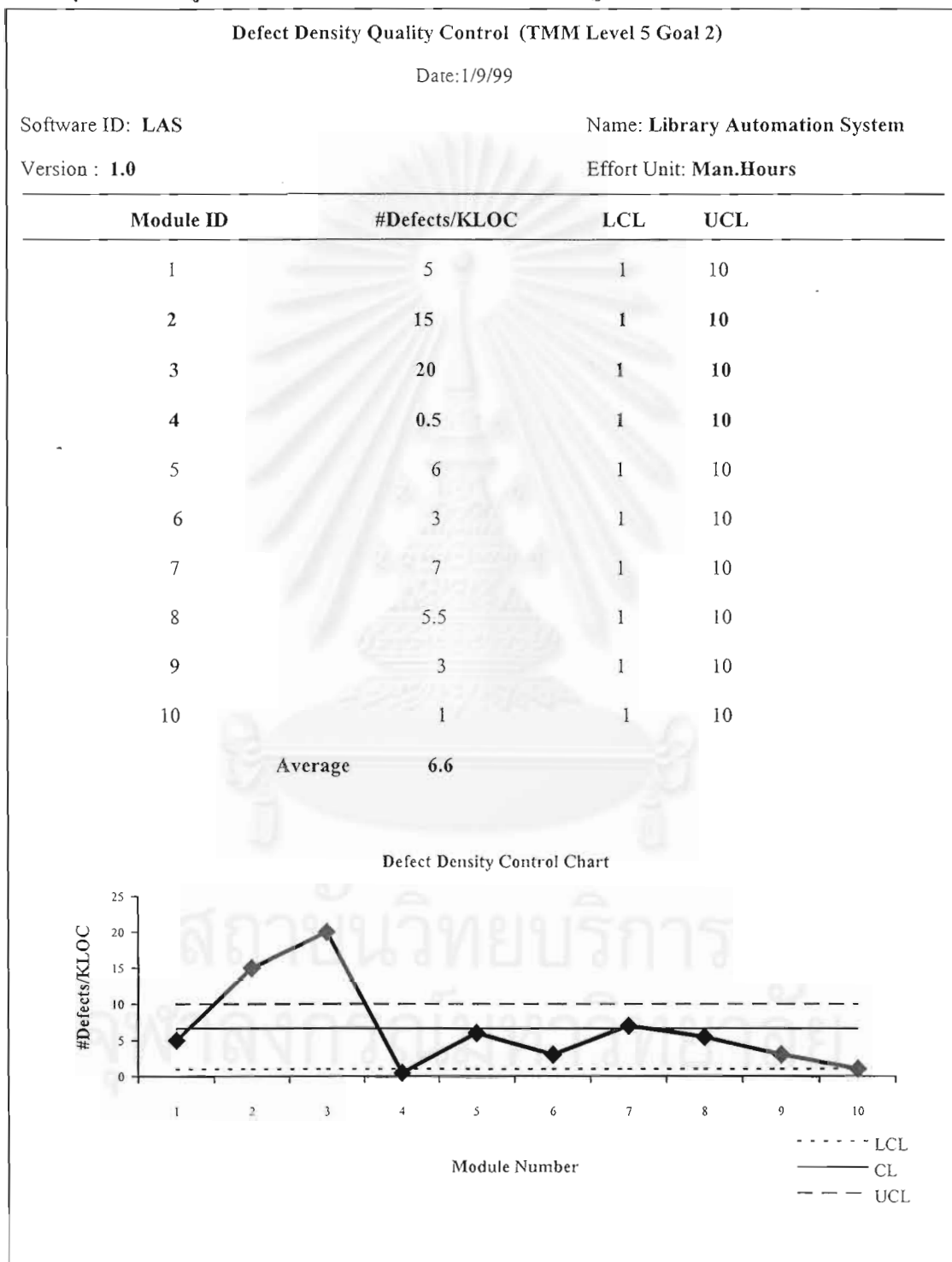
2) รายงานการควบคุมคุณภาพกระบวนการทดสอบ

วัตถุประสงค์ เพื่อใช้ในการติดตามและควบคุมคุณภาพกระบวนการทดสอบให้อยู่ในเกณฑ์มาตรฐานที่ตั้งไว้ ดังตัวอย่างรายงานในรูปที่ 4.97 แสดงให้เห็นถึงการควบคุมความครอบคลุมทางเดินพื้นฐานของโมดูลต่างๆ ค่าขีดจำกัดล่าง ค่าเฉลี่ย(Center Line) และค่าขีดจำกัดบน ถ้าโมดูลใดตกนอกค่าขีดจำกัด โมดูลนั้นๆ อาจจะมีปัญหาด้านคุณภาพ กระบวนการทดสอบไม่สามารถทดสอบให้ค่าความครอบคลุมได้ตามมาตรฐาน จำเป็นจะต้องปรับปรุงการทดสอบให้ครอบคลุมเป็นไปตามเกณฑ์



รูปที่ 4.97 ตัวอย่างรายงานแผนผังควบคุมความครอบคลุมทางเดินพื้นฐาน

ตัวอย่างรายงานในรูปที่ 4.98 แสดงให้เห็นถึงการควบคุมความหนาแน่นข้อบกพร่องของโมดูล ถ้าค่าต่ำกว่าค่าขีดจำกัดล่าง ต้องมีการตรวจสอบว่าเป็นเพราะโมดูลนั้นมีข้อผิดพลาดน้อยอยู่แล้ว หรือเป็นเพราะกระบวนการทดสอบไม่สามารถตรวจหาข้อบกพร่องที่มีอยู่นั้นพบ จำเป็นจะต้องใช้ความพยายามในการทดสอบมากขึ้น ถ้าค่าสูงเกินกว่าค่าขีดจำกัดบน ก็ต้องหาสาเหตุว่าทำไมโมดูลนั้นจึงมีข้อผิดพลาดมากกว่าเกณฑ์มาตรฐาน



รูปที่ 4.98 ตัวอย่างรายงานแผนผังควบคุมความหนาแน่นข้อบกพร่องของโมดูล

3) รายงานการทำให้กระบวนการทดสอบที่ดีที่สุด

- รายงานรายละเอียดกิจกรรมในการปรับปรุงกระบวนการทดสอบ

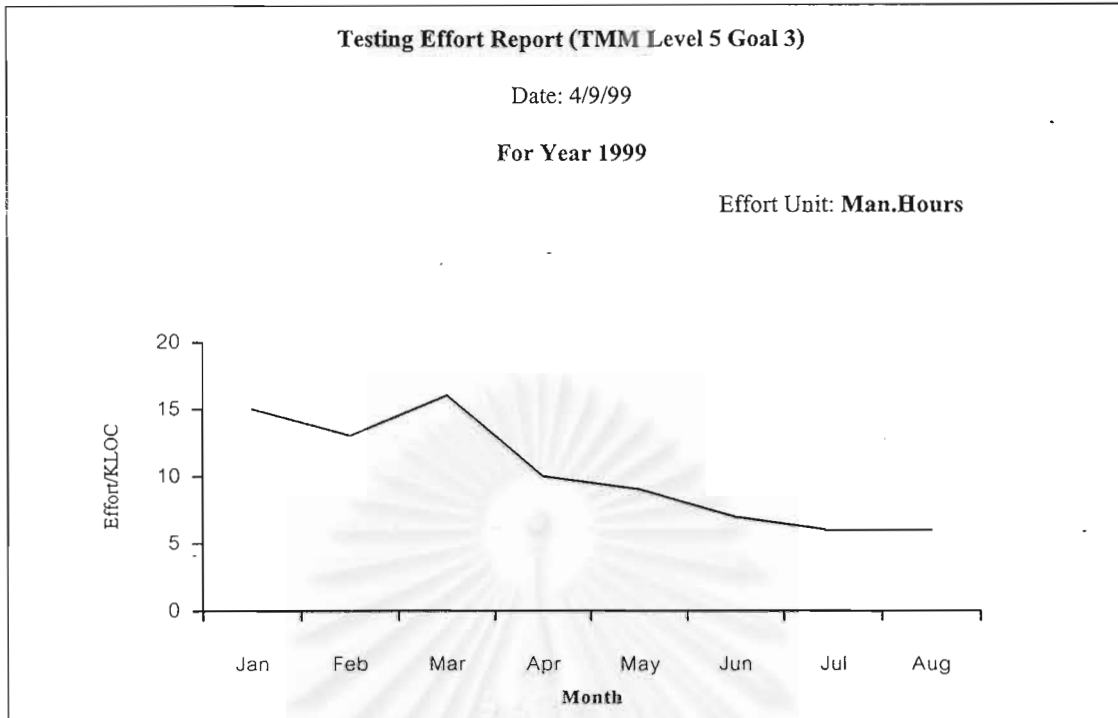
วัตถุประสงค์ เพื่อใช้ติดตามสถานะของกิจกรรมในการปรับปรุงกระบวนการทดสอบ ดังตัวอย่างรายงานในรูปที่ 4.99 แสดงให้เห็นถึงรายละเอียดในการประเมินเครื่องมือหรือเทคโนโลยีในการทดสอบมาใช้งาน ประเภทของเครื่องมือ ความพยายามที่ใช้ในการประเมิน ผลประโยชน์ที่คาดว่าจะได้และที่เกิดขึ้นจริง ค่าใช้จ่ายที่คาดว่าจะเกิดขึ้นและที่เกิดขึ้นจริง และผลลัพธ์ของการประเมิน

Testing Tool / Technology Evaluation Detail Report (TMM Level 5 Goal 3)										
Date: 4/9/99										
For Year 1999										
Effort Unit: Man.Hours										
TestingTool/ Technology ID	Description	Tool Type	Date	Effort	Benefit Expected	Actual	Cost Expected	Actual	Evaluation Result	
1	Test Power S/W Test Expert Co.	Test Case Generator	10/1/99	30	฿100,000	฿70,000	฿50,000	฿50,000	Accept	
2	Web Tester S/W ATT Co.	Capture/ Replay	12/1/99	20	฿200,000		฿300,000		Reject	
.	
.	
Total number of evaluated testing tool/technology :				20	Total evaluated effort :		500			
Total expected benefit (only accepted) :					฿5,000,000		Total actual benefit :			฿4,000,000
Total expected cost (only accepted) :					฿3,000,000		Total actual cost :			฿3,500,000
Expected net :					฿2,000,000		Actual net :			฿500,000

รูปที่ 4.99 ตัวอย่างรายงานสถานะของกิจกรรมในการปรับปรุงกระบวนการทดสอบ

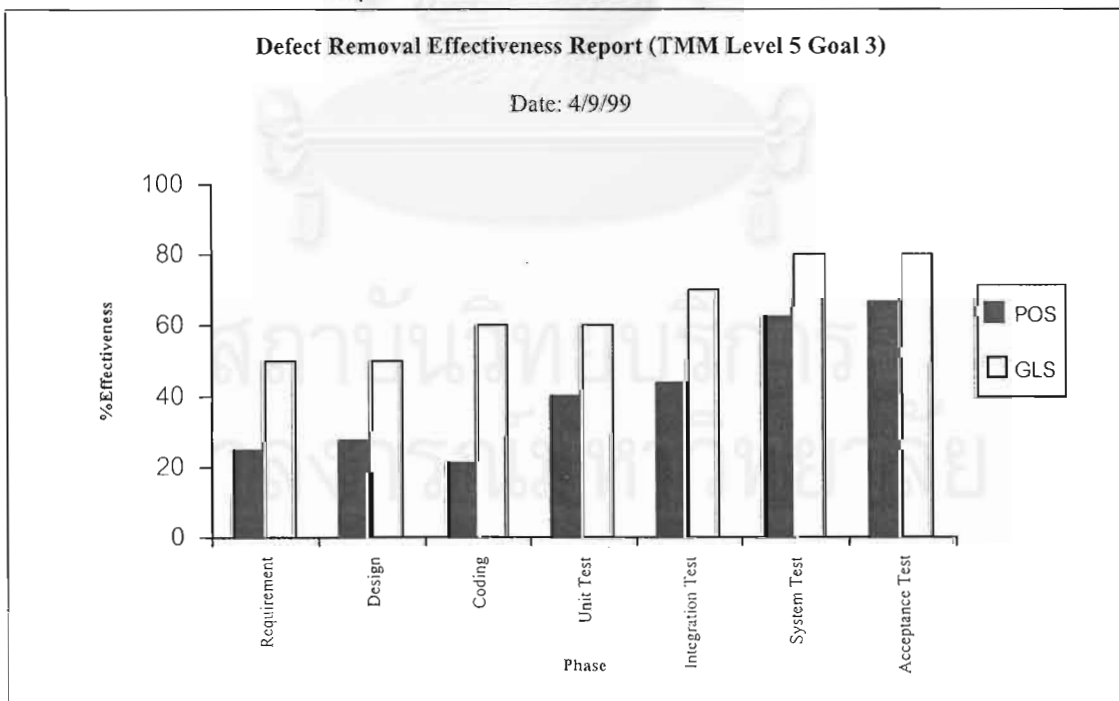
- รายงานสัญญาณ้กระบวนการทดสอบ

วัตถุประสงค์ เพื่อติดตามดูผลของกิจกรรมการปรับปรุงกระบวนการทดสอบ โดยการดูสัญญาณ้ที่ถึงคุณภาพกระบวนการทดสอบ เช่น ค่าใช้จ่ายในการทดสอบ ประสิทธิภาพในการจัดซื้ออุปกรณ์ ดังตัวอย่างรายงานในรูปที่ 4.100 แสดงให้เห็นถึงความพยายามที่ใช้ในการทดสอบต่อหนึ่งพันบรรทัดคำสั่งในแต่ละเดือน ถ้าเครื่องมือหรือเทคโนโลยีการทดสอบที่นำมาใช้ได้ผล ความพยายามที่ใช้ในการทดสอบควรจะลดลง โดยประสิทธิภาพในการจัดซื้ออุปกรณ์ไม่ลดลง



รูปที่ 4.100 ตัวอย่างรายงานความพยายามในการทดสอบในแต่ละเดือน

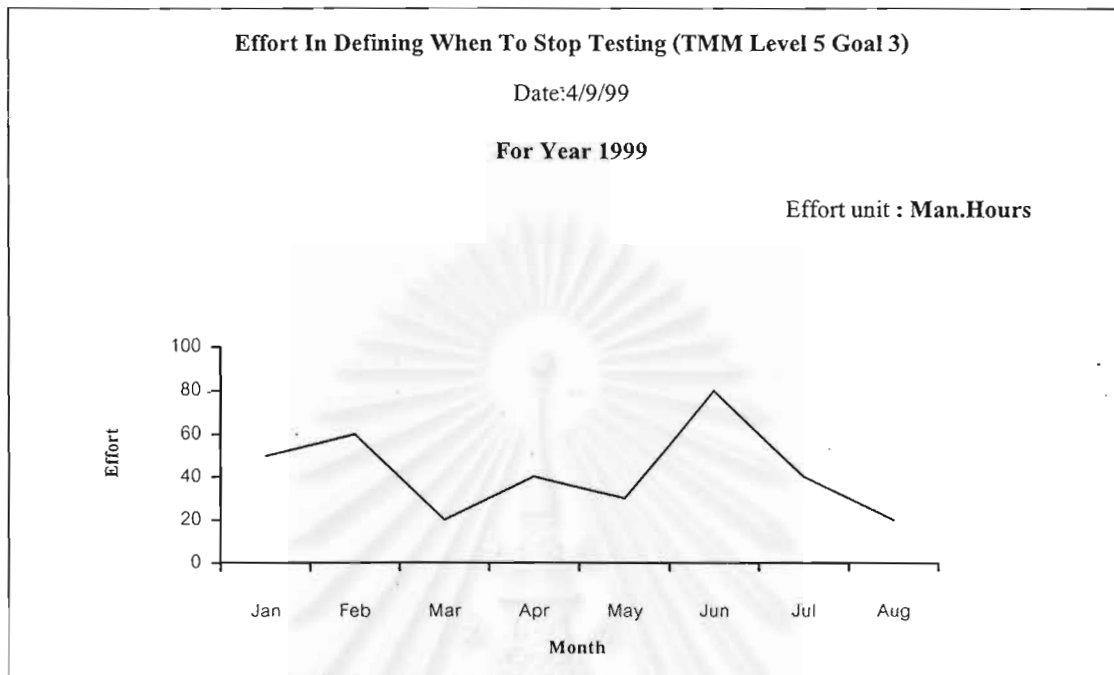
ตัวอย่างรายงานในรูปที่ 4.101 แสดงให้เห็นถึงการเปรียบเทียบประสิทธิผลในการขจัดข้อบกพร่องระหว่างโครงการว่ามีการเปลี่ยนที่ดีขึ้นหรือไม่ จะได้หาจุดบกพร่องของกระบวนการทดสอบและหาทางปรับปรุง



รูปที่ 4.101 ตัวอย่างรายงานประสิทธิผลในการขจัดข้อบกพร่องระหว่างโครงการ

- รายงานความพยายามในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ

วัตถุประสงค์ เพื่อติดตามสถานะกิจกรรมในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ ดังตัวอย่างรายงานในรูปที่ 4.102 แสดงให้เห็นถึงความพยายามที่ใช้แต่ละเดือนในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ

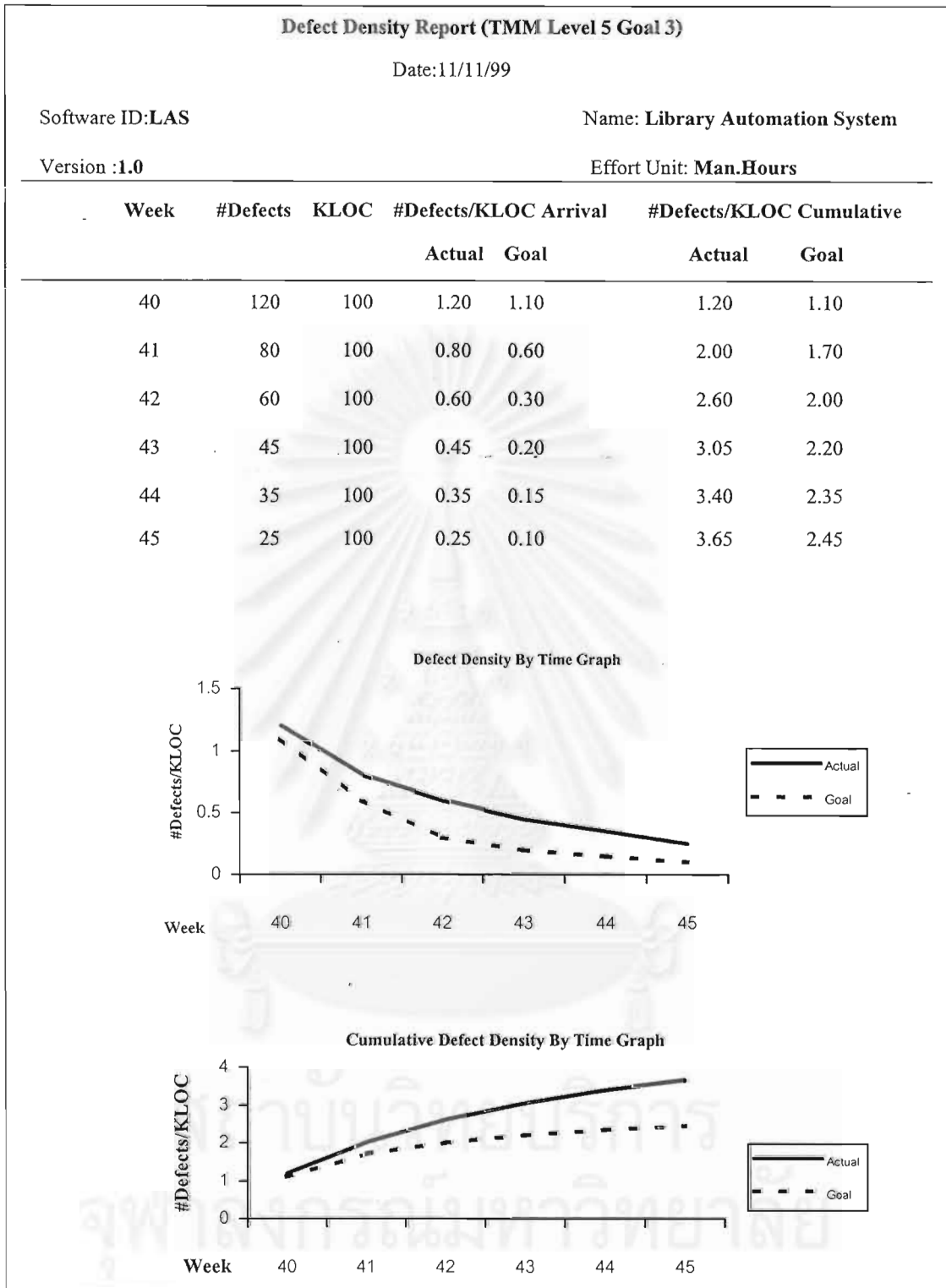


รูปที่ 4.102 ตัวอย่างรายงานความพยายามที่ใช้ในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ

- รายงานความหนาแน่นข้อบกพร่องตั้งแต่การทดสอบระบบ

วัตถุประสงค์ เพื่อใช้ในการกำหนดว่าเมื่อใดควรจะหยุดทดสอบและนำซอฟต์แวร์ไปใช้งานจริงได้ ดังตัวอย่างรายงานในรูปที่ 4.103 แสดงให้เห็นถึงความหนาแน่นข้อบกพร่องที่เกิดขึ้นและสะสมตั้งแต่การทดสอบระบบเทียบกับเป้าหมายที่ต้องการ ถ้าค่าที่เกิดขึ้นจริงใกล้เคียงกับเป้าหมายที่ตั้งไว้ ก็อาจจะเพียงพอที่จะระบุได้ว่าควรจะหยุดทดสอบ แต่ถ้าไม่ก็ควรที่จะทดสอบต่อไปอีก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.103 ตัวอย่างรายงานความหนาแน่นข้อบกพร่องตั้งแต่การทดสอบระบบ

4.3 การทดสอบเครื่องมือ

4.3.1 สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือที่พัฒนาขึ้นมีดังต่อไปนี้

- 1) เครื่องไมโครคอมพิวเตอร์ เพนเทียม 133 เมกะเฮิร์ตซ์ หน่วยความจำ 16 เมกะไบต์
- 2) ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ 95
- 3) เครื่องมือทำงานบนไมโครซอฟต์แอสเซส 97

4.3.2 วิธีการทดสอบ

- 1) จำลองการพัฒนาซอฟต์แวร์ขึ้นมา 2 โครงการ ซึ่งถูกพัฒนาโดยองค์กรที่อยู่ในระดับ 5 ของทีเอ็มเอ็ม
- 2) จำลองข้อมูลที่เกี่ยวข้องกับการทดสอบของซอฟต์แวร์ 2 โครงการดังกล่าว ซึ่งข้อมูลเป็นมาตรวัดในแต่ละระดับของทีเอ็มเอ็ม
- 3) ทดสอบการป้อนข้อมูลเข้าไปในทุกๆ หน้าจอของเครื่องมือโดยทำการเพิ่มรายการ ลบรายการและแก้ไขรายการ
- 4) ทดสอบการออกรายงานต่างๆ รายงานในแต่ละระดับของทีเอ็มเอ็มที่ได้ออกแบบไว้

4.3.3 ผลการทดสอบ

- 1) การป้อนข้อมูลเข้าไปในหน้าจอต่างๆ ปรากฏว่าเครื่องมือสามารถทำการเพิ่มรายการ ลบรายการ และแก้ไขรายการได้อย่างถูกต้อง
- 2) การออกรายงานต่างๆ ปรากฏว่าเครื่องมือสามารถออกรายงานตามที่ออกแบบไว้ได้อย่างถูกต้อง

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้เสนอมาตรวัดของการทดสอบสำหรับทดสอบมาทัวริตีโมเดล (ทีเอ็มเอ็ม) ในแต่ละระดับตั้งแต่ระดับ 2 ถึง 5 มาตรวัดสะท้อนให้เห็นถึงกระบวนการทดสอบขององค์กร ซึ่งข้อมูลมาตรวัดที่องค์กรเก็บจะเป็นประโยชน์ในการวิเคราะห์ และปรับปรุงกระบวนการทดสอบ มาตรวัดที่เสนอได้มาจากวิธีเป้าหมาย/คำถาม/มาตรวัด หรือจิกิวเอ็ม โดยที่เป้าหมายได้มาจากเป้าหมายในแต่ละระดับของทีเอ็มเอ็ม งานวิจัยได้ตั้งคำถามเพื่อให้ได้มาตรวัดที่สะท้อนให้เห็นถึงเป้าหมายนั้นๆ มาตรวัดที่ได้ถูกจัดแบ่งเป็น 2 ประเภทคือ มาตรวัดขั้นต่ำที่จำเป็นต้องมี และมาตรวัดที่จะมีหรือไม่มีก็ได้ ซึ่งมาตรวัดขั้นต่ำเพียงพอที่จะประเมินกระบวนการทดสอบได้ในระดับหนึ่งอันจะเป็นประโยชน์ในการปรับปรุงกระบวนการทดสอบ อย่างไรก็ตามงานวิจัยเสนอเพียงมาตรวัดที่องค์กรควรจะมีเท่านั้น ไม่ได้กำหนดถึงวิธีการที่จะเก็บข้อมูล และความเหมาะสมของค่าของมาตรวัด

งานวิจัยนี้ได้ทำการออกแบบและพัฒนาเครื่องมือเพื่อจะทำการเก็บข้อมูลมาตรวัดที่กำหนดขึ้นและออกรายงานที่จะเป็นประโยชน์ในการติดตามและวิเคราะห์กระบวนการทดสอบ เครื่องมือเก็บข้อมูลทั้งมาตรวัดขั้นต่ำและมาตรวัดที่จะมีหรือไม่มีก็ได้ การออกแบบเครื่องมือประกอบไปด้วย การออกแบบฐานข้อมูล การออกแบบอินพุต และการออกแบบเอาต์พุต การออกแบบฐานข้อมูลใช้หลักการออกแบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งตารางและเขตข้อมูลต่างๆ ได้มาจากมาตรวัดที่กำหนดขึ้น สำหรับการออกแบบอินพุตและเอาต์พุตได้จัดเมนูเป็นระดับตั้งแต่ระดับ 2 ถึง 5 เพื่อให้ง่ายต่อองค์กรเมื่ออยู่ในระดับใดจะได้สนใจเฉพาะข้อมูลในระดับนั้น ระดับที่สูงขึ้นจะต้องเก็บข้อมูลในระดับที่ต่ำกว่าด้วย การออกแบบหน้าจอในการป้อนข้อมูลคำนึงถึงมาตรวัดที่ต้องเก็บในแต่ละเป้าหมายของทีเอ็มเอ็มเพื่อนำเข้าเก็บในฐานข้อมูล สำหรับการออกแบบเอาต์พุตในการออกรายงานต่างๆ คำนึงถึงข้อมูลที่องค์กรต้องใช้ในการติดตามการทำการกิจกรรมการทดสอบเพื่อบรรลุเป้าหมายในแต่ละระดับของทีเอ็มเอ็ม

งานวิจัยได้ทำการพัฒนาเครื่องมือตามที่ได้ออกแบบ โดยใช้ไมโครซอฟต์แอ็คเซส 97 เครื่องมือรับข้อมูลมาตรวัดในแต่ละระดับเข้าเก็บในฐานข้อมูล โดยผู้ใช้ป้อนข้อมูลผ่านทางหน้าจอ เครื่องมือใช้ข้อมูลในฐานข้อมูลในการออกรายงานทั้งรายละเอียดในระดับ โครงการและองค์กรในการประเมินและวิเคราะห์กระบวนการทดสอบ

หลังจากพัฒนาเครื่องมือเสร็จได้ทำการทดสอบเครื่องมือ โดยจำลองการพัฒนาซอฟต์แวร์ขององค์กรที่อยู่ในระดับ 5 ของทีเอ็มเอ็ม และจำลองข้อมูลที่เกี่ยวข้องกับการทดสอบตามมาตรวัดในแต่ละ

ระดับของทีเอ็มเอ็ม ซึ่งผลการทดสอบปรากฏว่าการป้อนข้อมูลทุกหน้าจอเป็นไปอย่างถูกต้องและ เครื่องมือสามารถออกรายงานทุกรายงานตามที่ออกแบบไว้ได้อย่างถูกต้อง

5.2 ข้อเสนอแนะ

5.2.1 มาตรฐานที่เสนอสำหรับคุณสมบัติของซอฟต์แวร์ในทีเอ็มเอ็มระดับ 4 ยังวัดคุณสมบัติของซอฟต์แวร์ไม่ครอบคลุมเพียงพอ เช่นคุณสมบัติความสามารถในการทำงานที่งานวิจัยเสนอเพียงมาตรวัดเดียว อย่างไรก็ตามยังมีอีกหลายมาตรวัดที่จะวัดคุณสมบัตินี้ ซึ่งก็ขึ้นอยู่กับองค์กรต่างๆ ในการให้นิยามคุณสมบัติของซอฟต์แวร์ที่จะทำการวัดด้วย ดังนั้นควรจะหาวิธีการในการกำหนดมาตรวัดและหามาตรวัดที่ครอบคลุมคุณสมบัติในหลายๆ แง่มุม

5.2.2 มาตรวัดและเครื่องมือจากงานวิจัยควรจะนำไปทดลองใช้งานในองค์กรจริง เพื่อจะประเมินผลของมาตรวัดและเครื่องมือว่าช่วยในการปรับปรุงกระบวนการทดสอบได้ดีเพียงใด จะได้ปรับปรุงมาตรวัดและเครื่องมือให้เหมาะสมกับความต้องการขององค์กรต่างๆ ได้ดียิ่งขึ้น

5.2.3 เครื่องมือต้นแบบที่พัฒนาขึ้นเป็นแบบผู้ใช้คนเดียว ควรจะมีการปรับปรุงให้สามารถใช้งานในแบบผู้ใช้หลายคน (Multi-user)

5.2.4 เครื่องมือที่พัฒนายังไม่มีความสามารถในการนำข้อมูลเข้า (Import) และส่งข้อมูลออก (Export) กับโปรแกรมอื่นๆ ได้ เช่น โปรแกรมประเภทการทดสอบด้านความครอบคลุมการทดสอบ โปรแกรมประเภทการบริหารโครงการ ควรจะมีการปรับปรุงให้สามารถติดต่อกับโปรแกรมอื่นๆ ได้เพื่อลดการป้อนข้อมูลที่ซ้ำซ้อนและนำข้อมูลออกไปวิเคราะห์หรือออกรายงานในรูปแบบที่ต้องการได้เช่น โปรแกรมทางสถิติ

5.2.5 เครื่องมือที่พัฒนายังไม่มีความสามารถในการหาความสัมพันธ์ของมาตรวัดต่างๆ ได้ ซึ่งจะมีประโยชน์ต่อการไปใช้ในการประมาณการและควบคุมกระบวนการทดสอบ เช่น มาตรวัดความพยายามในการทดสอบมีความสัมพันธ์ในรูปของสมการคณิตศาสตร์กับขนาดของซอฟต์แวร์อย่างไร ถ้าทราบ องค์กรก็สามารถประมาณการความพยายามในการทดสอบได้จากขนาดของซอฟต์แวร์ เพื่อจะช่วยในการวางแผนการทดสอบได้ ดังนั้นควรจะปรับปรุงให้เครื่องมือสามารถสรุปความสัมพันธ์ของมาตรวัดต่างๆ

จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Roger S. Pressman. Software Engineering: A Practitioner 's Approach. 3rd ed. Singapore: McGraw-Hill, 1992.
2. Taratip Suwannasart. Towards Development of A Testing Maturity Model. Dissertation, Illinois Institute of Technology, May 1996.
3. Daich, G. Price, B. Ragland, and M. Dawood. Software Test Technologies Report. Software Technology Support Center, Hill Air Force Base, Utah, August 1994.
4. Mark C. Paula, Bill Curtis, Mary Beth Chrissis and Charles V. Weber. Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon University, February 1993.
5. Burnstein, I., Homyen, A., Grom, R., Carlson, C.R. A Model to Assess Testing Process Maturity. Crosstalk, The Journal of Defense Software Engineering, November 1998. pp. 26-30.
6. Kearney, J.K., R.L. Sedlmeyer, W.B. Thompson, M.A. Gray, and M.A. Adler. Software Complexity Measurement. Communications of the ACM, vol.29, no.11, November 1986.
7. Donald R. McAndrews. Establishing a Software Measurement Process. Technical Report CMU/SEI-93-TR-16, Software Engineering Institute, Carnegie Mellon University, July 1993.
8. Basili, V.R., and D.M. Weiss. A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering, Vol.SE-10, 1984. pp.728-738.
9. Stephen H. Kan. Metrics and Models in Software Quality Engineering. United States of America: Addison-Wesley Publishing Company, 1995.
10. John Gaffney, Robert Cruickshank, Richard Werling, and Henry Felber. Software Productivity Consortium. The Software Measurement Guidebook. United States of America: International Thomson Computer Press, 1995.



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

บทความวิชาการที่ได้รับการตีพิมพ์

A Set of Measurements to Improve Software Testing Process

Taratip Suwannasart¹ and Prapass Srichaivattana²
 Department of Computer Engineering, Faculty of Engineering
 Chulalongkorn University

Abstract: A set of measurements related to test-related data is developed to be included in the Testing Maturity Model (TMM). The objective of measurements is to track testing practices status in each level of the TMM with the exception of level one. We use the Goal/Question/Metric paradigm to derive the set of measurements. The set of measurements will help software development organizations improve their testing processes. Future research will be focused on developing a tool to collect the measurements and evaluating the usefulness of the measurements.

Key Words: Software testing, TMM, Measurement, Metrics

1. Introduction

Developing quality software is a goal of every software development organization. To achieve the goal, an organization must have a mature testing process. Since testing is key activities related to software quality. A Testing Maturity Model (TMM) has been developed to help software development organizations evaluate, and improve their testing processes. The TMM guides the organizations improve their testing processes by recommending a set of testing practices at each level of maturity. The details of TMM development can be found in [1].

In this paper we propose a set of recommended measurements to be included in the TMM. The proposed set of measurements assures that the organizations perform the set of recommended practices. Moreover, the measurements will show the status of the testing practices performed. A major goal of the measurements is to guide which test-related data to be collected at each level of the TMM (except level one). These quantitative data will be serve as a foundation for managing and tracking testing process, and a baseline for collecting organization's historical test-related data.

This paper we discuss an overview of the TMM. How to define the set of recommended metrics is then described. The set of recommended measurements of each maturity level is discussed. Summary and future research are then outlined.

2. An Overview of The TMM

The TMM development was guided by four sources: the Capability Maturity Model (CMM) version 1.1 [2], Gelperin and Hetzel's evolutionary testing model [3], industry testing

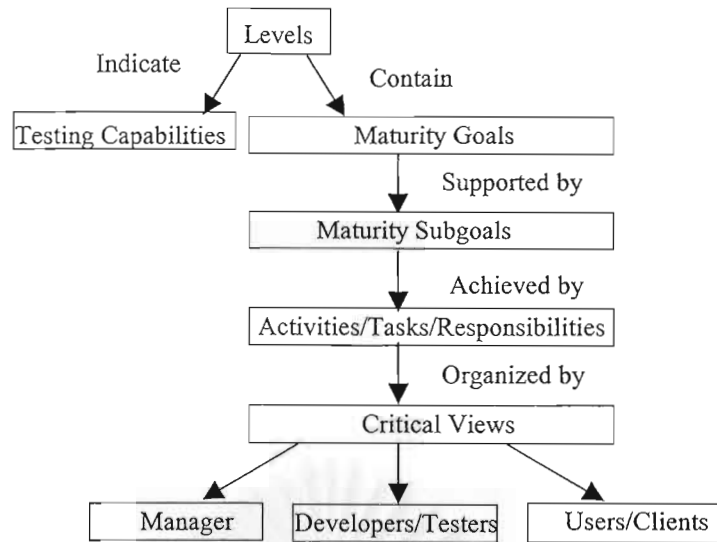
practices [4], and Beizer's progression phases of a tester's mental model [5]. The CMM is a comprehensive process improvement model developed by the Software Engineering Institute (SEI) that has been accepted by the software industry. The Gelperin and Hetzel's testing model serves as the foundation for level differentiation in the TMM. The industry testing practices illustrates the testing practices and environments in the software industry. The Beizer's model gives the individual tester's thinking process.

The TMM version 1.0 has 2 major components[6,7]: a set of levels and an assessment model. The details of each component are described as follows:

2.1 A Set of Levels

The TMM is characterized as five maturity levels. The TMM levels define a position in a testing maturity hierarchy. Characteristics of each level are described in terms of testing capabilities and organization goals. With the exception of level one, each level has the framework of maturity goals, maturity subgoals, activities, tasks, and responsibilities. The structure of the TMM is shown in figure 1.

The set of maturity goals identifies key process areas that must be addressed to improve testing capability. The maturity goals of each level (except level one) are illustrated in figure 2. The maturity subgoals support one or more maturity goals and define the scope, boundaries, and needed accomplishments for a particular level. Activities, tasks, and responsibilities are organized by three critical views that represent each of the key participants involved in the testing process: managers, developers and testers, as well as users and clients.



^s
Figure 1: The Structure of the TMM

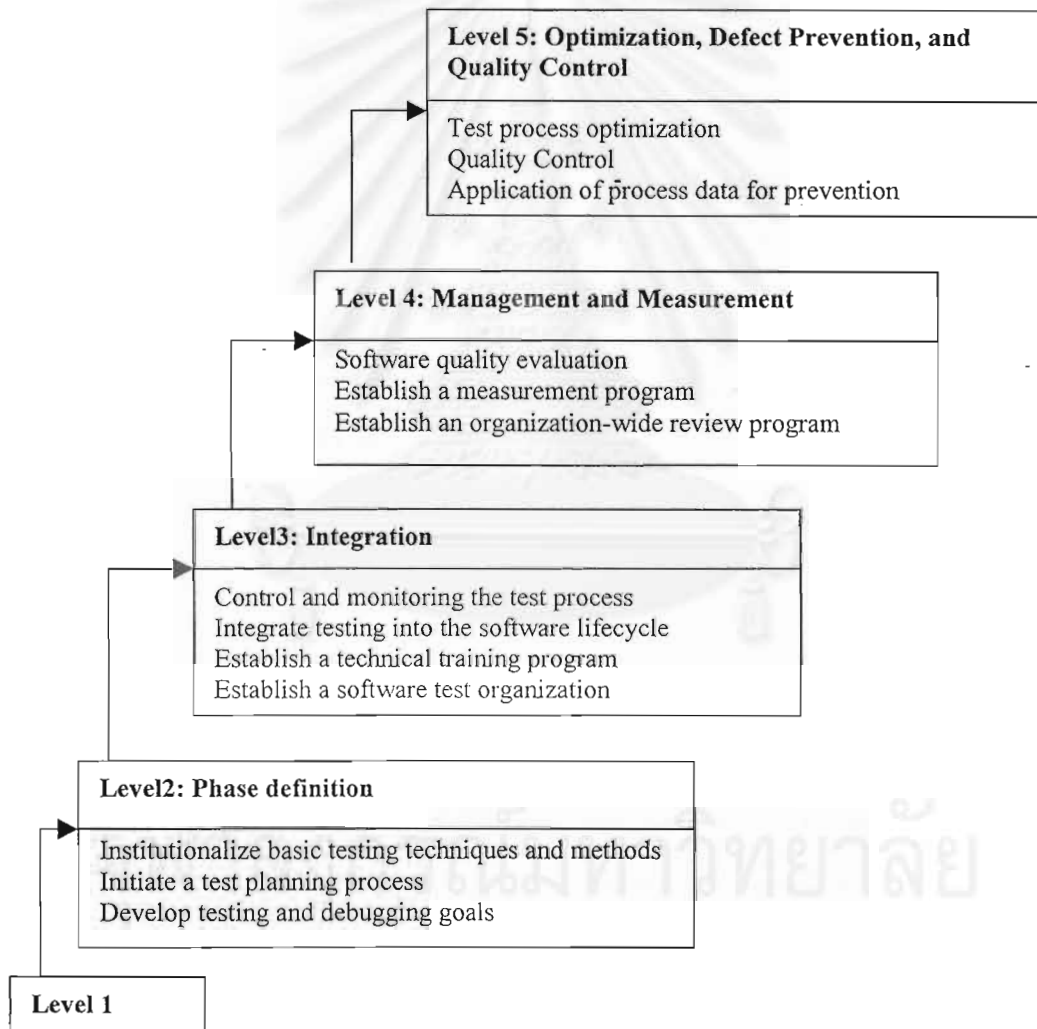


Figure 2: The Maturity Goals by Level

2.2 The Assessment Model

The TMM Assessment Model (TMM-AM) [8] can be used by software organizations to

assess and improve their testing process capabilities. The TMM-AM consists of three components:

- The assessment procedure: provides a series of steps to be followed by an organization that needs to carry out a testing process self assessment.
- A questionnaire: is the assessment instrument that helps collect and record assessment information.
- Assessment training and team selection criteria: The assessment training provide a preparation of the assessment team to conduct the self-assessment. The assessment team members are selected based on their understanding of the assessment goals, knowledge, experiences, skills, and commitments to test process improvement.

Following the TMM assessment procedure, an organization obtains the outputs of the assessment as follows:

- TMM level (on a scale from 1 to 5)
- The organization testing process strengths and weaknesses
- The Test process profile (a summary of the organization's testing process)
- Action plans for testing process improvement
- Assessment record

3. How do we derive A Set of Measurements?

We use Goal/Question/Metric (GQM) paradigm [9] to derive a set of measurements for each level of the TMM. The GQM is developed to counter the problem that many software metric programs have failed because they have poorly defined, or even non-existent objectives. Originally, it was used for evaluating defects for a set of projects in the NASA/GSFC environment. Because of its intuitive nature, the approach has gained widespread appeal. According to the GQM, there are three stages as follows:

1. Set goals specific to needs. Goals may be defined for any object, for a variety of reasons, with respect to various models of quality, from various points of view, relative to a particular environment.
2. Refine the goals into quantifiable questions that are tractable.
3. Deduce the metrics and data to be collected to answer the questions.

In this research, goals will be the maturity goals of each level. Questions will be formulated to achieve the goals. Then we derive the set of measurements to answer those questions. The details of each question can be found in [10]. In this paper, we present only the maturity goals and what to be measured in order to achieve the goals.

4. A Set of Measurements

As we discussed in the section 3 that we use the GQM paradigm to derive a set of measurements for the TMM. This section we describe brief characteristics of each level of the TMM with the exception of level 1, maturity goals along with their brief practices, and a set of recommended measurements.

4.1 Level 2: Phase Definition

The testing goal at this level is to demonstrate that the software product satisfies its specifications and requirements. Testing and debugging concepts are still not clearly defined. Execution is considered as a primary testing activity. There is some test planning, but it is done after the code is developed. This level consists of 3 maturity goals, which are described below. A set of measurements of this level is shown in Table 1.

Goal 2.1: Develop testing and debugging goal

In order to distinguish the differences between testing and debugging process, an organization should separate their cost.

Goal 2.2: Initiate a test planning process

The project manager should track status of developing the test plan and define testing activities, i.e. cost and schedule.

Goal 2.3: Institutionalize Basic Testing Techniques and Methods

In order to know the status of using basic testing techniques and methods, the organization may track outputs from their uses, i.e. test cases.

Goals	A Set of Measurements
2.1	Testing cost Debugging cost
2.2	Cost in developing test plans Test schedule
2.3	Numbers of test cases derived from test techniques: black-box, white-box

Table 1: A Set of Measurement at level 2

4.2 Level 3: Integration

An organization at this level has as its primary testing goal: the ability to detect faults in order to demonstrate that the program is error-free and satisfies its requirements. The concept of testing and debugging are clearly defined. Tests are designed early to force the organization to be concerned with testing at the earlier phases of the software development process. The maturity goals of this level are described below and a set of measurements is shown in Table 2.

Goal 3.1: Establishing the Software Test Organization

In order to know the status of the test organization, its manpower and cost should be tracked.

Goal 3.2: Establish a Technical Training Program

The organization should provide a suitable quantity of test training to testers and SQA groups, i.e. the number of test-related courses per year.

Goal 3.3: Integration of Testing into the Software Life Cycle

In order to track the results of integration of testing into the software development life cycle, the organization may use defects removed (found), defects injected (origin) and defects escaped during testing in software development life cycle in order to derive the effectiveness of integration.

Goal 3.4: Controlling and Monitoring the Testing Process

For activities defined in test plans, the organization should control and monitor to see if they are performed as planned. Test cases are very important in the testing process, so the organization should monitor their running status such as the number of test cases passed and failed. The organization should monitor the quality of unit testing by measuring the test coverage: statement, branch and basis path coverage. Moreover, the status of integration testing should be tracked

Goals	A Set of Measurements
3.1	Numbers of independent testers Cost of test organization
3.2	Numbers of test-related courses offered per year
3.3	Numbers of defects found by phase
3.4	Schedule slippage Numbers of test cases passed Numbers of test cases failed Numbers of tested modules % statement coverage % branch coverage % basis path coverage Numbers of integrated and tested modules

Table 2: A Set of Measurements at Level 3

4.3 Level 4: Management and Measurement

The primary goal of testing is to detect requirements, design, and implementation faults. The testing activities start at the earliest phases of software development life cycle. Testing is parallel, and not sequential to software implementation. The testing life cycle is fully coordinated with the software development life cycle and the review process. The maturity goals are described below and a proposed set of measurements at this level is shown in Table 3.

Goal 4.1: Establish an Organization-wide Review Program

The organization should control the effectiveness of the review process. The number of defects removed by reviews is measured to indicate the effectiveness.

Goal 4.2: Establish a Test Measurement Program

The TMM mentions that the organization should collect details about defects, such as their severity and type. Furthermore, the productivity of testers should be measured.

Goal 4.3: Software Quality Evaluation

The organization should define quality attributes of a software product and measure these attributes.

Goals	A Set of Measurements
4.1	Numbers of defects removed by reviews
4.2	Numbers of defects founded in each severity Numbers of defects found in each defect type Testers' productivity
4.3	Correctness Efficiency Flexibility Interoperability Maintainability Portability Reliability Reusability Testability Usability

Table 3: A Set of Measurements at Level 4

4.4 Level 5: Optimization/Defect Prevention and Quality Control

The primary goal of an organization's testing activities of this level is defect prevention. Prevention applies to requirements, design and implementation faults. At this level, we test to demonstrate that the software satisfies its specification, to detect faults, and to prevent faults. This level consists of 3 maturity goals, which are described below. The Table 4 shows a set of measurements at this level.

Goal 5.1: Application of Process Data for Defect Prevention

The organization should track activities of the defect prevention, i.e. cost of finding root causes of defects.

Goal 5.2: Quality Control

The organization should define the lower control limit (LCL) and the upper control limit (UCL) of metrics in the control chart to achieve statistical process control. Good indicators of testing process, that should be controlled, are as follows defect density, review effort rate, test effort rate, % statement coverage, % branch coverage, and % basis path coverage.

Goal 5.3: Test Process Optimization

The organization should track the status in evaluating new testing tools and technologies. The TMM mentions that the organization should determine when to stop testing. One possible way is to compare the goal and actual failure rate. If the differences are acceptable, the software is ready to be released.

Goals	A Set of Measurements
5.1	Total effort in finding root causes of defects
5.2	LCL and UCL for defect density, review effort rate, test effort rate, % statement coverage, % branch coverage, and % basis path coverage
5.3	Total effort in evaluating new testing tools and technologies Total effort in defining when to stop testing Failure rate goal Actual failure rate

Table 4: A Set of Measurements at Level 5

5 Summary

This paper presents a set of recommended measurements for each maturity goal of the TMM version 1.0. The set of measurements is derived by the GQM approach. It reflects an organization's testing process and product so that the organization can use the collected data to improve the testing process effectively. What we propose here is only a minimum set of measurements which we believe that they are enough to reflect an organization's testing process. We define only what to be collected, not how to collect. The characteristics of each measure and a full set of measurements can be found in [10].

For future research, we propose the following:

1. Characteristics of software quality metrics in the level 4 are not clearly defined. A Framework to measure them needs to be developed.
2. A tool for gathering data sources to calculate each measure will be developed. The tool will help an organization to analyze and report test-related data. It also should be developed to be able to interface with other testing tools such as test coverage tool, test case data generator tool, and project management tool.
3. The proposed set of measurements should be used and evaluated by software development organizations. Feedback from them will help us to improve the TMM.

6 References

- [1] Burnstein, I., Suwannasart, T., Carlson, C.R., "Developing a Testing Maturity Model for Software Test Process Evaluation and Improvement," IEEE International Test Conference, Washington D.C., October 20-25, 1996, pp. 581-589.
- [2] Paulk, M., et al. "Key Practices of the Capability Maturity Model, Version 1.0," Technical Report CMU/SEI-93-TR-25, Software Engineering Institute, Pittsburgh, Pa., 1993.
- [3] Gelperin, D., and Hetzel, B., "The Growth of Software Testing," Communication of the ACM, Vol. 31, No. 6, 1988, pp. 687-695.
- [4] Durant, J., Software Testing Practices Survey Report, TR5-93, May 1993.
- [5] Beizer, B., Software System Techniques, 2nd ed., Van Nostrand Reinhold, New York, 1990.
- [6] Burnstein, I., Suwannasart, T., Carlson, C.R., "Developing a Testing Maturity Model: Part I," Crosstalk, The Journal of Defense Software Engineering, August 1996, pp. 21-24.
- [7] Burnstein, I., Suwannasart, T., Carlson, C.R., "Developing a Testing Maturity Model: Part II," Crosstalk, The Journal of Defense Software Engineering, September 1996, pp. 19-26.
- [8] Burnstein, I., Homyen, A., Grom, R., Carlson, C.R., A Model to Assess Testing Process Maturity, Crosstalk, The Journal of Defense Software Engineering, November 1998, pp. 26-30.
- [9] Basili, V.R., and Weiss, D.M., A Methodology for Collecting Valid Software Engineering Data, IEEE Transactions on Software Engineering, Vol. SE-10, 1984, pp. 728-738.
- [10] Srichaivattana, P., "The Set of TMM Metrics," Technical Report, Software Engineering Lab, Department of Computer Engineering, Chulalongkorn University, 1999.

ภาคผนวก ข
ตารางต่างๆ ในฐานข้อมูล

ชื่อตาราง : AcceptanceTest

วัตถุประสงค์ : เพื่อเก็บข้อมูลในขั้นตอนการทดสอบเพื่อยอมรับ

เขตข้อมูล (Field)	PK ¹	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์ ²
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType
TestActivityID	X	หมายเลขกิจกรรมการทดสอบเพื่อยอมรับ	Text	5	TestPlan
Date	X	วันที่ทำกิจกรรมการทดสอบ	Date	8	
LOC		จำนวนบรรทัดคำสั่งของซอฟต์แวร์ ที่มาทดสอบ	Single	4	
Manpower		จำนวนคนที่ทำการทดสอบ	Integer	2	
Effort		ความพยายามที่ใช้ในการทดสอบ	Single	4	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType
#Defects		จำนวนข้อบกพร่องที่เกิดขึ้น	Integer	2	
Note		บันทึกข้อความเพิ่มเติม	Memo	- ³	

หมายเหตุ: 1. PK = กุญแจหลัก(Primary Key)

2. ตารางที่สัมพันธ์หมายถึงตารางที่เขตข้อมูล ไปอ้างอิงกับเขตข้อมูลของตารางนั้น

3. - หมายถึงขนาดไม่จำกัด

ชื่อตาราง : AcceptanceTestPlan

วัตถุประสงค์ : เพื่อเก็บข้อมูลความต้องการที่ทดสอบในแผนงานการทดสอบเพื่อยอมรับ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestActivityID	X	หมายเลขกิจกรรมการทดสอบเพื่อยอมรับ	Text	5	TestPlan
Requirement IDTested	X	หมายเลขความต้องการที่ทำการ ทดสอบ	Text	5	

ชื่อตาราง : CasualMeeting

วัตถุประสงค์ : เพื่อเก็บรายละเอียดในการประชุมหาสาเหตุรากของข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
CasualMeetingID	X	หมายเลขการประชุมหาสาเหตุราก ของข้อบกพร่อง	Text	10	
PlannedDate		วันที่ทำการประชุมตามแผน	Date	8	
ActualDate		วันที่ทำการประชุมจริง	Date	8	
PlannedDuration		จำนวนชั่วโมงที่ใช้ในการประชุม ตามแผน	Single	4	
ActualDuration		จำนวนชั่วโมงที่ใช้ในการประชุมจริง	Single	4	
PlannedManpower		จำนวนคนที่เข้าร่วมประชุมตามแผน	Integer	2	
ActualManpower		จำนวนคนที่เข้าร่วมประชุมจริง	Integer	2	
PlannedEffort		ความพยายามในการหาสาเหตุราก ตามแผน	Single	4	
ActualEffort		ความพยายามในการหาสาเหตุรากจริง	Single	4	
NumberOfActions		จำนวนของกิจกรรมในการป้องกัน ข้อบกพร่องที่ได้จากการประชุม	Integer	2	
Note		บันทึกข้อความเพิ่มเติม	Memo	-	

ชื่อตาราง : DefectCauseCategory

วัตถุประสงค์ : เพื่อเก็บประเภทของสาเหตุของข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย (Description)	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
CauseCode	X	รหัสของสาเหตุ	Text	5	
Description		รายละเอียดของสาเหตุของข้อบกพร่อง	Text	50	

ชื่อตาราง : DefectDebuggingEffort

วัตถุประสงค์ : เพื่อเก็บความพยายามในการแก้ไขข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย (Description)	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
DefectID	X	หมายเลขข้อบกพร่อง	Text	5	Defects
PersonID	X	หมายเลขบุคลากรที่ทำการแก้ไข	Text	5	Persons
Date	X	วันที่ทำการแก้ไข	Date	8	
DebuggingEffort		ความพยายามในการแก้ไข	Single	4	

ชื่อตาราง : DefectPreventionActivity

วัตถุประสงค์ : เพื่อเก็บรายละเอียดกิจกรรมการป้องกันข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
ActivityID	X	หมายเลขกิจกรรม	Text	5	
Description		รายละเอียดกิจกรรม	Text	200	
PlannedStart Date		วันที่เริ่มทำกิจกรรมตามแผนงาน	Date	8	
ActualStartDate		วันที่เริ่มทำกิจกรรมจริง	Date	8	
PlannedEndDate		วันที่สิ้นสุดกิจกรรมตามแผนงาน	Date	8	
ActualEndDate		วันที่สิ้นสุดกิจกรรมจริง	Date	8	
PlannedEffort		ความพยายามที่ใช้การทำกิจกรรมตาม แผน	Single	4	
PlannedCost		ค่าใช้จ่ายที่ใช้ในการทำกิจกรรมตาม แผน	Single	4	
ActualCost		ค่าใช้จ่ายที่ใช้ในการทำกิจกรรมจริง	Single	4	
ExpectedResult		ผลลัพธ์ที่คาดหวัง	Memo	-	
ActualResult		ผลลัพธ์ที่ได้จริง	Memo	-	

ชื่อตาราง : DefectPreventionActivityEffort

วัตถุประสงค์ : เพื่อเก็บความพยายามในการทำกิจกรรมการป้องกันข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
ActivityID	X	หมายเลขกิจกรรม	Text	5	Defect Prevention Activity
PersonID	X	รายละเอียดกิจกรรม	Text	-	Persons
Date	X	วันที่เริ่มทำกิจกรรมตามแผนงาน	Date	8	
Effort		ความพยายามในการทำกิจกรรม	Single	4	

ชื่อตาราง : Defects

วัตถุประสงค์ : เพื่อเก็บรายละเอียดข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
DefectID	X	หมายเลขข้อบกพร่อง	Text	5	
DefectDetail		รายละเอียดข้อบกพร่อง	Text	200	
FoundDate		วันที่พบข้อบกพร่อง	Date	8	
ClosedDate		วันที่แก้ไขข้อบกพร่องเสร็จ	Date	8	
FoundPhase		ขั้นตอนที่พบข้อบกพร่อง	Integer	2	Software Development Phase
OriginPhase		ขั้นตอนที่ข้อบกพร่องเกิดขึ้น	Integer	2	Software Development Phase
ModuleIDFound		หมายเลขโมดูลที่พบข้อบกพร่อง	Text	10	
Testing Technique		เทคนิคการทดสอบที่ใช้ในการ ตรวจพบข้อบกพร่อง	Text	2	Testing Techniques
DefectSeverity		ระดับความรุนแรงของข้อบกพร่อง	Text	2	DefectSeverity
DefectType		ประเภทของข้อบกพร่อง	Text	5	DefectType

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
CauseCode		รหัสของสาเหตุของข้อบกพร่อง	Text	5	DefectCause Category

ชื่อตาราง : DefectSeverity

วัตถุประสงค์ : เพื่อเก็บระดับความรุนแรงของข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
DefectSeverity	X	ระดับความรุนแรงของข้อบกพร่อง	Text	2	
Description		รายละเอียดความรุนแรงของข้อบกพร่อง	Text	50	

ชื่อตาราง : DefectType

วัตถุประสงค์ : เพื่อเก็บประเภทของข้อบกพร่อง

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
DefectType	X	ประเภทของข้อบกพร่อง	Text	5	
Description		รายละเอียดประเภทของข้อบกพร่อง	Text	100	

ชื่อตาราง : EffortUnit

วัตถุประสงค์ : เพื่อเก็บหน่วยของความพยายาม

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
EffortUnit	X	หน่วยของความพยายาม	Text	2	
Description		รายละเอียดของหน่วยของความพยายาม	Text	20	

ชื่อตาราง : IntegrationTest

วัตถุประสงค์ : เพื่อเก็บข้อมูลในขั้นตอนการทดสอบเบ็ดเสร็จ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType
TestActivityID	X	หมายเลขกิจกรรมการทดสอบ	Text	5	TestPlan
Date	X	วันที่ทำกิจกรรมการทดสอบ	Date	8	
LOC		จำนวนบรรทัดคำสั่งของซอฟต์แวร์ ที่มาทดสอบ	Single	4	
Manpower		จำนวนคนที่ทำการทดสอบ	Integer	2	
Effort		ความพยายามที่ใช้ในการทดสอบ	Single	4	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType
#Defects		จำนวนข้อบกพร่องที่เกิดขึ้น	Integer	2	-
Note		บันทึกข้อความเพิ่มเติม	Memo	-	

ชื่อตาราง : IntegrationTestPlan

วัตถุประสงค์ : เพื่อเก็บรายละเอียดของโมดูลต่างๆที่นำมารวมกันในแผนงานการทดสอบเบ็ดเสร็จ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestActivityID	X	หมายเลขกิจกรรมการทดสอบเบ็ดเสร็จ	Text	5	TestPlan
ModuleID	X	หมายเลขโมดูลที่นำมาทดสอบการ รวมกัน	Text	10	

ชื่อตาราง : ModuleIntegrationTest

วัตถุประสงค์ : เพื่อเก็บผลลัพธ์ของการทดสอบ โมดูลต่างๆที่นำมารวมกัน

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestActivityID	X	หมายเลขกิจกรรมการทดสอบเบ็ดเสร็จ	Text	5	TestPlan

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
ModuleID Integrated	X	หมายเลข โมดูลที่นำมาทดสอบการ รวมกัน	Text	10	
Date	X	วันที่ทำการทดสอบ	Date	8	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType

ชื่อตาราง : Persons

วัตถุประสงค์ : เพื่อเก็บข้อมูลของบุคลากรที่เกี่ยวข้องกับกระบวนการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
PersonID	X	หมายเลขบุคลากร	Text	5	
Name		ชื่อบุคลากร	Text	50	
Surname		นามสกุล	Text	50	
PersonType		ประเภทบุคลากร	Text	2	PersonType
InTestOrganization		บุคลากรอยู่ในหน่วยงานการทดสอบ หรือไม่	Yes/No	1 บิต	

ชื่อตาราง : PersonType

วัตถุประสงค์ : เพื่อเก็บข้อมูลประเภทบุคลากรที่เกี่ยวข้องกับกระบวนการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
PersonType	X	ประเภทบุคลากร	Text	2	
Description		รายละเอียดประเภทบุคลากร	Text	20	

ชื่อตาราง : QualityControlParameters

วัตถุประสงค์ : เพื่อเก็บค่าของพารามิเตอร์ที่ใช้ในการควบคุมกระบวนการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
Date		วันที่ป้อนข้อมูล	Date	8	

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบนารี)	ตารางที่ สัมพันธ์
ModuleDefect DensityLCL		ค่าขีดจำกัดล่างของความหนาแน่น ของข้อบกพร่องของโมดูล	Single	4	
ModuleDefect DensityUCL		ค่าขีดจำกัดบนของความหนาแน่น ของข้อบกพร่องของโมดูล	Single	4	
Requirement ReviewRateLCL		ค่าขีดจำกัดล่างของอัตราทบทวน ความต้องการ	Single	4	
Requirement ReviewRateUCL		ค่าขีดจำกัดบนของอัตราทบทวน ความต้องการ	Single	4	
DesignReview RateLCL		ค่าขีดจำกัดล่างของอัตราทบทวน การออกแบบ	Single	4	
DesignReview RateUCL		ค่าขีดจำกัดบนของอัตราทบทวน การออกแบบ	Single	4	
CodeReview RateLCL		ค่าขีดจำกัดล่างของอัตราทบทวน โปรแกรม	Single	4	
CodeReview RateUCL		ค่าขีดจำกัดบนของอัตราทบทวน โปรแกรม	Single	4	
TestDocument ReviewRateLCL		ค่าขีดจำกัดล่างของอัตราทบทวน เอกสารทดสอบ	Single	4	
TestDocument ReviewRateUCL		ค่าขีดจำกัดบนของอัตราทบทวน เอกสารทดสอบ	Single	4	
TestEffortRateLCL		ค่าขีดจำกัดล่างของอัตราความ พยายามในการทดสอบ	Single	4	
TestEffortRateUCL		ค่าขีดจำกัดบนของอัตราความ พยายามในการทดสอบ	Single	4	
StatementCoverage LCL		ค่าขีดจำกัดล่างของความครอบคลุม ข้อความสั่ง	Single	4	
StatementCoverage UCL		ค่าขีดจำกัดบนของความครอบคลุม ข้อความสั่ง	Single	4	

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
BranchCoverage LCL		ค่าขีดจำกัดล่างของความครอบคลุม กิ่ง	Single	4	
BranchCoverage UCL		ค่าขีดจำกัดบนของความครอบคลุม กิ่ง	Single	4	
BasisPathCoverage LCL		ค่าขีดจำกัดล่างของความครอบคลุม ทางเดิน	Single	4	
BasisPathCoverage UCL		ค่าขีดจำกัดบนของความครอบคลุม ทางเดิน	Single	4	

ชื่อตาราง : RequirementAcceptanceTest

วัตถุประสงค์ : เพื่อเก็บผลลัพธ์ของการทดสอบความต้องการในขั้นตอนการทดสอบเพื่อยอมรับ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestActivityID	X	หมายเลขกิจกรรมการทดสอบเพื่อ ยอมรับ	Text	5	TestPlan
RequirementID	X	หมายเลขความต้องการที่ทำการ ทดสอบ	Text	5	Acceptance TestPlan
Date	X	วันที่ทำการทดสอบ	Date	8	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType

ชื่อตาราง : RequirementSystemTest

วัตถุประสงค์ : เพื่อเก็บผลลัพธ์ของการทดสอบความต้องการในขั้นตอนการทดสอบระบบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestActivityID	X	หมายเลขกิจกรรมการทดสอบระบบ	Text	5	TestPlan
RequirementID	X	หมายเลขความต้องการที่ทำการ ทดสอบ	Text	5	SystemTest Plan

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
Date	X	วันที่ทำการทดสอบ	Date	8	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType

ชื่อตาราง : ResultType

วัตถุประสงค์ : เพื่อเก็บชนิดผลลัพธ์ของการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
Result	X	รหัสของผลลัพธ์	Text	2	
Description		ความหมายของผลลัพธ์	Text	10	

ชื่อตาราง : Review

วัตถุประสงค์ : เพื่อเก็บรายละเอียดในการทบทวน

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
ReviewID	X	หมายเลขการทบทวน	Text	5	ReviewPlan
Date	X	วันที่ทำการทบทวน	Date	8	
QuantityOfWork ProductReviewed		ปริมาณของผลผลิตที่นำมาทบทวน	Single	4	
NumberOf Reviewers		จำนวนคนที่เข้าทำการทบทวน	Integer	2	
PreparationTime		เวลาที่ใช้ในการเตรียมการทบทวน	Single	4	
InspectionTime		เวลาที่ใช้ในการตรวจ	Single	4	
ReviewResult		ผลลัพธ์ของการทบทวน	Text	2	ReviewResult
NumberOfDefects		จำนวนข้อบกพร่องที่เกิดขึ้น	Integer	2	
Note		บันทึกข้อความเพิ่มเติม	Memo	-	

ชื่อตาราง : ReviewedWorkProductType

วัตถุประสงค์ : เพื่อเก็บชนิดของผลผลิตที่นำมาทบทวน

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
ReviewedWork ProductType	X	ชนิดของผลผลิตที่นำมาทบทวน	Text	5	
Description		รายละเอียดของผลผลิตที่นำมาทบทวน	Text	20	
UnitOfWork Product		หน่วยของผลผลิตที่นำมาทบทวน	Text	10	

ชื่อตาราง : ReviewPlan

วัตถุประสงค์ : เพื่อเก็บรายละเอียดของแผนงานการทบทวน

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
ReviewID	X	หมายเลขการทบทวน	Text	5	
Description		รายละเอียดการทบทวน	Text	100	
LastUpdatedDate		วันที่เปลี่ยนแปลงข้อมูลล่าสุด	Date	8	
ReviewedWork ProductType		ชนิดของผลผลิตที่นำมาทบทวน	Text	2	
PlannedQuantity OfWorkProduct Reviewed		ปริมาณของผลผลิตที่นำมาทบทวน ตามแผนงาน	Single	4	
PlannedReview Date		วันที่ทำการทบทวนตามแผนงาน	Date	8	
PlannedNumber OfReviewers		จำนวนคนที่เข้าทำการทบทวนตาม แผนงาน	Integer	2	
Planned PreparationTime		เวลาที่ใช้ในการเตรียมการทบทวน ตามแผนงาน	Single	4	
Planned InspectionTime		เวลาที่ใช้ในการตรวจตามแผนงาน	Single	4	

ชื่อตาราง : ReviewResult

วัตถุประสงค์ : เพื่อเก็บชนิดผลลัพธ์ของการทบทวน

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
ReviewResult	X	ผลลัพธ์ของการทบทวน	Text	2	
Description		ความหมายของผลลัพธ์	Text	20	

ชื่อตาราง : Software

วัตถุประสงค์ : เพื่อเก็บรายละเอียดของซอฟต์แวร์ที่นำมาทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	
Name		ชื่อซอฟต์แวร์	Text	50	
Version		รุ่นของซอฟต์แวร์	Text	10	
TotalPagesOf Requirement Document		จำนวนหน้าของเอกสารความต้องการ	Integer	2	
TotalNumberOf Requirements		จำนวนความต้องการทั้งหมด	Integer	2	
TotalPagesOf DesignDocument		จำนวนหน้าของเอกสารการออกแบบ	Integer	2	
TotalModules		จำนวน โมดูลทั้งหมด	Integer	2	
TotalLinesOfCode		จำนวนบรรทัดคำสั่งของโปรแกรม	Single	4	
LinesOfCodeUnit		หน่วยของบรรทัดคำสั่งของโปรแกรม	Text	5	
TotalPagesOfUnit Test		จำนวนหน้าของเอกสารการทดสอบ โมดูล	Integer	2	
TotalPagesOf IntegrationTest		จำนวนหน้าของเอกสารการทดสอบ เบ็ดเสร็จ	Integer	2	
TotalPagesOf SystemTest		จำนวนหน้าของเอกสารการทดสอบ ระบบ	Integer	2	

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TotalPagesOf AcceptanceTest		จำนวนหน้าของเอกสารการทดสอบเพื่อยอมรับ	Integer	2	
EffortUnit		หน่วยของความพยายามสำหรับซอฟต์แวร์นี้	Text	2	EffortUnit

ชื่อตาราง : SoftwareDevelopmentPhase

วัตถุประสงค์ : เพื่อเก็บขั้นตอนในการพัฒนาซอฟต์แวร์

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
PhaseID	X	หมายเลขขั้นตอนในการพัฒนาซอฟต์แวร์	Integer	2	
Description		รายละเอียดของขั้นตอนในการพัฒนาซอฟต์แวร์	Text	50	

ชื่อตาราง : SoftwareQualityAttributes

วัตถุประสงค์ : เพื่อเก็บค่ามาตรวัดที่เกี่ยวข้องกับคุณสมบัติของซอฟต์แวร์

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	
Date	X	วันที่เก็บข้อมูล	Date	8	
NumberOfDefects RelativeTo RequirementAnd Standard		จำนวนข้อบกพร่องเกี่ยวกับความต้องการและมาตรฐาน	Integer	2	
NumberOfDefects RelativeToSecurity		จำนวนข้อบกพร่องเกี่ยวกับความปลอดภัย	Integer	2	
LinesOfCode		จำนวนบรรทัดคำสั่งของซอฟต์แวร์	Single	4	
ActualUtilization		การใช้ประโยชน์จริง	Single	4	
AllocatedUtilization		การใช้ประโยชน์ที่จัดสรร	Single	4	

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
AverageLaborDays ToChangeSoftware		ระยะเวลาโดยเฉลี่ยที่ใช้ในการ เปลี่ยนแปลง	Single	4	
AverageLaborDays ToFix		ระยะเวลาโดยเฉลี่ยที่ใช้ในการ บำรุงรักษา	Single	4	
AverageLaborHours ToUse		ระยะเวลาโดยเฉลี่ยที่ผู้ใช้ในการ การปฏิบัติงาน	Single	4	
EffortToTest		ความพยายามที่ใช้ในการทดสอบ	Single	4	
EffortToCouple		ความพยายามที่ใช้ในการทำงาน ร่วมกับระบบอื่น	Single	4	
EffortToDevelop		ความพยายามที่ใช้ในการพัฒนา	Single	4	
EffortToTransport		ความพยายามที่ใช้ในการย้ายไป ทำงานกับระบบอื่น	Single	4	
EffortToConvert		ความพยายามที่ใช้ในการปรับ เปลี่ยน	Single	4	

ชื่อตาราง : SystemTest

วัตถุประสงค์ : เพื่อเก็บข้อมูลในขั้นตอนการทดสอบระบบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType
TestActivityID	X	หมายเลขกิจกรรมการทดสอบ	Text	5	TestPlan
Date	X	วันที่ทำกิจกรรมการทดสอบ	Date	8	
LOC		จำนวนบรรทัดคำสั่งของซอฟต์แวร์ ที่มาทดสอบ	Single	4	
Manpower		จำนวนคนที่ทำการทดสอบ	Integer	2	
Effort		ความพยายามที่ใช้ในการทดสอบ	Single	4	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType
#Defects		จำนวนข้อบกพร่องที่เกิดขึ้น	Integer	2	

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
Note		บันทึกข้อความเพิ่มเติม	Memo	-	.

ชื่อตาราง : SystemTestPlan

วัตถุประสงค์ : เพื่อเก็บข้อมูลความต้องการที่ทดสอบในแผนงานการทดสอบระบบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestActivityID	X	หมายเลขกิจกรรมการทดสอบระบบ	Text	5	TestPlan
Requirement IDTested	X	หมายเลขความต้องการที่ทำการทดสอบ	Text	5	

ชื่อตาราง : TestCaseEffort

วัตถุประสงค์ : เพื่อเก็บความพยายามในการพัฒนากรณีทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestCaseID	X	หมายเลขกรณีทดสอบ	Text	5	TestCases
PersonID	X	หมายเลขบุคลากรที่ทำการพัฒนา	Text	5	Persons
Date	X	วันที่ทำการพัฒนา	Date	8	
DevelopmentEffort		ความพยายามในการพัฒนา	Single	4	

ชื่อตาราง : TestCasePlan

วัตถุประสงค์ : เพื่อเก็บข้อมูลแผนงานการนำกรณีทดสอบมาทดสอบในกิจกรรมการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TestActivityID	X	หมายเลขกิจกรรมการทดสอบ	Text	5	TestPlan
TestCaseID	X	หมายเลขกรณีทดสอบ	Text	5	TestCases

ชื่อตาราง : TestCases

วัตถุประสงค์ : เพื่อเก็บรายละเอียดกรณีทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestCaseID	X	หมายเลขกรณีทดสอบ	Text	5	
Description		รายละเอียดของกรณีทดสอบ	Text	100	
TestingTechnique		เทคนิคที่ใช้ในการพัฒนากรณี ทดสอบ	Text	2	Testing Techniques
PlannedStartDate		วันเริ่มการพัฒนากรณีทดสอบตาม แผน	Date	8	
ActualStartDate		วันเริ่มการพัฒนากรณีทดสอบจริง	Date	8	
PlannedEndDate		วันที่พัฒนากรณีทดสอบเสร็จตามแผน	Date	8	
ActualEndDate		วันที่พัฒนากรณีทดสอบเสร็จจริง	Date	8	
PlannedEffort		ความพยายามในการพัฒนาตามแผน	Single	4	

ชื่อตาราง : TestCaseStatus

วัตถุประสงค์ : เพื่อเก็บผลลัพธ์ของการนำกรณีทดสอบมาทำการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType
TestActivityID	X	หมายเลขกิจกรรมการทดสอบ	Text	5	TestPlan
TestCaseID	X	หมายเลขกรณีทดสอบ	Text	5	TestCases
Date	X	วันที่ทำการทดสอบ	Date	8	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType

ชื่อตาราง : TestMeasurementProgram

วัตถุประสงค์ : เพื่อเก็บรายละเอียดโปรแกรมการวัดกระบวนการทดสอบขององค์กร

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
ActivityID	X	หมายเลขกิจกรรม	Text	5	
Description		รายละเอียดกิจกรรม	Text	200	
PlannedStartDate		วันที่เริ่มทำกิจกรรมตามแผนงาน	Date	8	
ActualStartDate		วันที่เริ่มทำกิจกรรมจริง	Date	8	
PlannedEndDate		วันที่สิ้นสุดกิจกรรมตามแผนงาน	Date	8	
ActualEndDate		วันที่สิ้นสุดกิจกรรมจริง	Date	8	
PlannedCost		ค่าใช้จ่ายของกิจกรรมตามแผนงาน	Single	4	
ActualCost		ค่าใช้จ่ายของกิจกรรมจริง	Single	4	
PlannedManpower		จำนวนคนที่ทำกิจกรรมตามแผนงาน	Integer	2	
ActualManpower		จำนวนคนที่ทำกิจกรรมจริง	Integer	2	

ชื่อตาราง : TestingTechniques

วัตถุประสงค์ : เพื่อเก็บเทคนิคในการทดสอบขององค์กร

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TestingTechnique	X	เทคนิคในการทดสอบ	Text	2	
Description		รายละเอียดเทคนิคในการทดสอบ	Text	50	

ชื่อตาราง : TestOrganization

วัตถุประสงค์ : เพื่อเก็บรายละเอียดค่าใช้จ่ายของหน่วยงานการทดสอบขององค์กร

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TestCostID	X	หมายเลขอ้างอิงค่าใช้จ่าย	Text	10	
Description		รายละเอียดค่าใช้จ่าย	Text	100	
MonthYearUsed		เดือนที่เกิดค่าใช้จ่ายนี้	Date	8	
Budget		ค่าใช้จ่ายตามงบประมาณ	Single	4	
ActualCost		ค่าใช้จ่ายที่เกิดขึ้นจริง	Single	4	

ชื่อตาราง : TestPlan

วัตถุประสงค์ : เพื่อเก็บรายละเอียดแผนงานการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType
TestActivityID	X	หมายเลขกิจกรรมการทดสอบ	Text	5	
LastUpdatedDate		วันที่เปลี่ยนแปลงข้อมูลล่าสุด	Date	8	
Description		รายละเอียดของกิจกรรมการทดสอบ	Text	100	
PlannedStartDate		วันที่เริ่มทำกิจกรรมตามแผนงาน	Date	8	
PlannedEndDate		วันที่สิ้นสุดกิจกรรมตามแผนงาน	Date	8	
PlannedManpower		จำนวนคนที่ทำกิจกรรมตามแผนงาน	Integer	2	
PlannedEffort		ความพยายามที่ใช้ตามแผนงาน	Single	4	

ชื่อตาราง : TestPlanDevelopmentEffort

วัตถุประสงค์ : เพื่อเก็บความพยายามในการพัฒนาแผนงานการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType
PersonID	X	หมายเลขบุคลากรที่ทำการพัฒนา	Text	5	Persons
Date	X	วันที่ทำการพัฒนา	Date	8	
DevelopmentEffort		ความพยายามในการพัฒนา	Single	4	

ชื่อตาราง : TestProcessOptimizationActivity

วัตถุประสงค์ : เพื่อเก็บรายละเอียดการประเมินเทคโนโลยีหรือเครื่องมือการทดสอบในการปรับปรุงกระบวนการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TestTool TechnologyID	X	หมายเลขเทคโนโลยีหรือเครื่องมือ การทดสอบ	Text	10	

(มีต่อ)

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
Description		รายละเอียดเทคโนโลยีหรือเครื่องมือการทดสอบที่ประเมิน	Text	100	
Date		วันที่ทำการประเมิน	Date	8	
EffortUnit		หน่วยความพยายามที่ใช้ในการประเมิน	Text	2	EffortUnit
TestToolType		ประเภทของเครื่องมือการทดสอบ	Integer	2	TestToolType
ExpectedBenefit		ผลประโยชน์ในรูปตัวเงินที่คาดว่าจะได้รับ	Single	4	
ExpectedCost		ค่าใช้จ่ายในรูปตัวเงินที่คาดว่าจะเกิดขึ้น	Single	4	
ActualBenefit		ผลประโยชน์ในรูปตัวเงินที่ได้รับจริง	Single	4	
ActualCost		ค่าใช้จ่ายในรูปตัวเงินที่เกิดขึ้นจริง	Single	4	
EvaluationResult		ผลลัพธ์ของการประเมิน	Text	20	

ชื่อตาราง : TestProcessOptimizationEffort

วัตถุประสงค์ : เพื่อเก็บความพยายามในการประเมินหรือใช้เทคโนโลยีและเครื่องมือการทดสอบในการปรับปรุงกระบวนการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TestingTool-TechnologyID	X	หมายเลขเทคโนโลยีหรือเครื่องมือการทดสอบ	Long	4	TestProcessOptimizationActivity
PersonID	X	หมายเลขบุคลากรที่ทำการประเมินหรือใช้	Text	5	Persons
Date	X	วันที่ทำการประเมินหรือใช้	Date	8	
Effort		ความพยายามในการประเมินหรือใช้	Single	4	

ชื่อตาราง : TestProcedureEffort

วัตถุประสงค์ : เพื่อเก็บความพยายามในการพัฒนาขั้นตอนปฏิบัติการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestProcedureID	X	หมายเลขขั้นตอนปฏิบัติการทดสอบ	Text	5	Test Procedures
PersonID	X	หมายเลขบุคลากรที่ทำการพัฒนา	Text	5	Persons
Date	X	วันที่ทำการพัฒนา	Date	8	
DevelopmentEffort		ความพยายามในการพัฒนา	Single	4	

ชื่อตาราง : TestProcedures

วัตถุประสงค์ : เพื่อเก็บรายละเอียดขั้นตอนปฏิบัติการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestProcedureID	X	หมายเลขขั้นตอนปฏิบัติการทดสอบ	Text	5	
Description		รายละเอียดขั้นตอนปฏิบัติการทดสอบ	Text	100	
PlannedStartDate		วันเริ่มการพัฒนาขั้นตอนปฏิบัติการทดสอบตามแผน	Date	8	
ActualStartDate		วันเริ่มการพัฒนาขั้นตอนปฏิบัติการทดสอบจริง	Date	8	
PlannedEndDate		วันที่พัฒนาเสร็จตามแผน	Date	8	
ActualEndDate		วันที่พัฒนาเสร็จจริง	Date	8	
PlannedEffort		ความพยายามในการพัฒนาตามแผน	Single	4	

ชื่อตาราง : TestToolType

วัตถุประสงค์ : เพื่อเก็บชนิดของเครื่องมือการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TestingToolTypeID	X	หมายเลขเครื่องมือการทดสอบ	Integer	2	
Description		รายละเอียดเครื่องมือการทดสอบ	Text	50	

ชื่อตาราง : TestType

วัตถุประสงค์ : เพื่อเก็บประเภทของการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
TestType	X	ประเภทการทดสอบ	Text	2	
Description		รายละเอียดประเภทการทดสอบ	Text	50	

ชื่อตาราง : Training

วัตถุประสงค์ : เพื่อเก็บรายละเอียดการฝึกอบรมด้านการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
CourseID	X	หมายเลขหลักสูตรการฝึกอบรม	Text	5	
CourseName		ชื่อหลักสูตรการฝึกอบรม	Text	50	
PlannedStartDate		วันเริ่มการฝึกอบรมตามแผนงาน	Date	8	
ActualStartDate		วันเริ่มการฝึกอบรมจริง	Date	8	
PlannedDuration		ระยะเวลาการฝึกอบรมตามแผนงาน	Integer	2	
ActualDuration		ระยะเวลาการฝึกอบรมจริง	Integer	2	
PlannedCost		ค่าใช้จ่ายในการฝึกอบรมตามแผน	Single	4	
ActualCost		ค่าใช้จ่ายในการฝึกอบรมที่เกิดขึ้นจริง	Single	4	
PlannedNumberOfAttendances		จำนวนผู้เข้ารับการอบรมตามแผน	Integer	2	
ActualNumberOfAttendances		จำนวนผู้เข้ารับการอบรมจริง	Integer	2	

ชื่อตาราง : UnitTest

วัตถุประสงค์ : เพื่อเก็บข้อมูลในขั้นตอนการทดสอบโมดูล

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestType	X	ประเภทการทดสอบ	Text	2	TestType
TestActivityID	X	หมายเลขกิจกรรมการทดสอบ	Text	5	TestPlan
Date	X	วันที่ทำกิจกรรมการทดสอบ	Date	8	
LOC		จำนวนบรรทัดคำสั่งของโมดูล ที่มาทดสอบ	Single	4	
Manpower		จำนวนคนที่ทำการทดสอบ	Integer	2	
Effort		ความพยายามที่ใช้ในการทดสอบ	Single	4	
ExercisedStatements		ข้อความสั่งที่ถูกทดสอบ	Integer	2	
TotalStatements		ข้อความสั่งทั้งหมด	Integer	2	
ExercisedBranches		จำนวนกิ่งที่ถูกทดสอบ	Integer	2	
TotalBranches		จำนวนกิ่งทั้งหมด	Integer	2	
ExercisedBasisPaths		จำนวนทางเดินพื้นฐานที่ถูก ทดสอบ	Integer	2	
TotalBasisPaths		จำนวนทางเดินพื้นฐานทั้งหมด	Integer	2	
Result		ผลลัพธ์ของการทดสอบ	Text	2	ResultType
#Defects		จำนวนข้อบกพร่องที่เกิดขึ้น	Integer	2	
Note		บันทึกข้อความเพิ่มเติม	Memo	-	

ชื่อตาราง : UnitTestPlan

วัตถุประสงค์ : เพื่อเก็บข้อมูลแผนงานการทดสอบโมดูล

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
TestActivityID	X	หมายเลขกิจกรรมการทดสอบ	Text	5	TestPlan
ModuleIDTested	X	หมายเลขโมดูลที่นำมาทดสอบ	Text	10	

ชื่อตาราง : WhenToStopTestingDefineEffort

วัตถุประสงค์ : เพื่อเก็บความพยายามในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
PersonID	X	หมายเลขบุคลากรที่ทำการกำหนด	Text	5	Persons
Date	X	วันที่ทำการกำหนด	Date	8	
Effort		ความพยายามในการกำหนด	Single	4	

ชื่อตาราง : WhenToStopTestingCriteria

วัตถุประสงค์ : เพื่อเก็บเกณฑ์ที่ใช้ในการกำหนดว่าเมื่อใดควรจะหยุดการทดสอบ

เขตข้อมูล	PK	ความหมาย	ประเภท	ขนาด (ไบต์)	ตารางที่ สัมพันธ์
SoftwareID	X	หมายเลขซอฟต์แวร์	Text	10	Software
Week	X	สัปดาห์ตั้งแต่ขั้นตอนการทดสอบระบบ	Integer	2	
DefectDensityGoal		เป้าหมายของความหนาแน่นข้อบกพร่องตั้งแต่ขั้นตอนการทดสอบระบบ	Single	4	

ประวัติผู้เขียน

นาย ประภาส ศรีชัยวัฒน์ เกิดเมื่อวันที่ 18 มีนาคม พ.ศ. 2510 จบการศึกษาในระดับปริญญาตรีจาก คณะวิศวกรรมศาสตร์ สาขาอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปี 2531 และในระดับปริญญาโทจาก คณะพาณิชยศาสตร์และการบัญชี สาขาการเงิน มหาวิทยาลัยธรรมศาสตร์ ในปี 2538 และเข้าศึกษาต่อในหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิทยาศาสตร์คอมพิวเตอร์ เมื่อ พ.ศ. 2539 ปัจจุบันทำงานในตำแหน่งอาจารย์ประจำ คณะบริหารธุรกิจ สาขาคอมพิวเตอร์ธุรกิจ มหาวิทยาลัยเกริก



สถาบันวิทยาบริการ
จุฬาลงกรณ์มหาวิทยาลัย