

# *Payment Transaction and Report Management Software Quality Assurance Plan*

---

SWEGroup5  
Version 1.0

Document Control Page

File Name	:	PTRM_SQA_Plan.doc
Template Version	:	02/08/2009

Version	Change Description			Remark
<1.0, 02/08/2009>	Original Version			
	Performed by / Date	Reviewed by / Date	Approved by / Date	
	Sirikanya Yomkoet 25/05/2009	Nattawut Riyasan 02/08/2009	Jirasith Srisupho 05/08/2009	

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	PURPOSE .....	1
1.2	SCOPE .....	1
1.3	DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	5
1.3.1	<i>Key Acronyms</i> .....	5
1.3.2	<i>Key Terms</i> .....	6
1.4	REFERENCES .....	6
<b>2.</b>	<b>SQA MANAGEMENT .....</b>	<b>7</b>
2.1	ORGANIZATION .....	7
2.1.1	<i>Project Manager</i> .....	7
2.1.2	<i>SQA Manager</i> .....	8
2.1.3	<i>SQA Team</i> .....	8
2.2	TASKS .....	8
2.3	RESPONSIBILITIES .....	9
2.3.1	<i>PTMR Responsibilities</i> .....	9
2.3.2	<i>Configuration Control Board (CCB)</i> .....	11
<b>3.</b>	<b>SQA DOCUMENTATION.....</b>	<b>12</b>
3.1	DEVELOPMENT, VERIFICATION AND VALIDATION, USE, AND MAINTENANCE.....	12
3.2	CONTROL .....	12
<b>4.</b>	<b>STANDARD AND PRACTICES.....</b>	<b>12</b>
4.1	CODING/DESIGN LANGUAGE STANDARDS .....	12
4.2	DOCUMENTATION STANDARDS .....	12
<b>5.</b>	<b>REVIEWS AND AUDITS. ....</b>	<b>13</b>
5.1	WALKTHROUGHS .....	13
5.1.1	<i>Overview</i> .....	13
5.1.2	<i>Procedures</i> .....	13
<b>6.</b>	<b>CONFIGURATION MANAGEMENT.....</b>	<b>15</b>
<b>7.</b>	<b>TESTING.....</b>	<b>15</b>
7.1	SYSTEM TESTING .....	15
7.2	INTEGRATION TESTING.....	15
7.3	UNIT TESTING .....	15
7.4	TEST DOCUMENTATION.....	15
<b>8.</b>	<b>SOFTWARE METRICS .....</b>	<b>15</b>
<b>9.</b>	<b>PROBLEM TRACKING .....</b>	<b>15</b>
<b>10.</b>	<b>RECORDS COLLECTION, MAINTENANCE, AND RETENTION.....</b>	<b>16</b>
<b>11.</b>	<b>TRAINING .....</b>	<b>16</b>
<b>12.</b>	<b>RISK MANAGEMENT.....</b>	<b>17</b>



# 1. Introduction

เนื่องจากปัจจุบัน หน่วยงาน Consumer Payment Management มีรายการรับชำระเข้ามาเป็นจำนวนมาก จากช่องทางการรับชำระหลายช่องทาง ทำให้เจ้าหน้าที่ไม่สามารถทำการตรวจสอบรายการรับชำระ ได้ทัน เพราะไม่มีระบบรองรับการทำงาน เจ้าหน้าที่ต้องเปิดรายงานการรับชำระของแต่ละช่องทางมาตรวจสอบกับรายงานสรุปการรับชำระ ของแต่ละบริษัทว่าแต่ละช่องทางนั้นจำนวนรายการรับชำระ และ จำนวนเงิน ตรงกันหรือไม่ และจากหากมีรายการรับชำระผิดปกติเช่น มีการรับชำระจากลูกค้าแต่ไม่มีการตัดยอดในระบบ เจ้าหน้าที่ต้องใช้เวลานานกว่าจะหาได้ว่ารายการรับชำระรายการใดที่ผิดปกติ ซึ่งต้องใช้คนเป็นจำนวนมากในการตรวจสอบ จากปัญหานี้จะส่งผลให้ลูกค้าถูกตัดบริการ และสร้างความไม่พอใจแก่ลูกค้า และกระทบต่อภาพพจน์ขององค์กร ทางหน่วยงานจึงมีความต้องการสร้างระบบรองรับการทำงานให้ทันตามเวลาที่กำหนด และสามารถเก็บข้อมูลไปใช้ทำรายงานสรุปประจำวัน และประจำเดือน ซึ่งรายงานที่ได้จะนำไปใช้เป็นข้อมูลสำหรับประกอบการตัดสินใจ

## 1.1 Purpose

เอกสารฉบับนี้จัดทำขึ้นเพื่อใช้สำหรับจัดทำแผนงานคุณภาพสำหรับโครงการ กำหนดวิธีการตรวจสอบและควบคุมคุณภาพของผลิตภัณฑ์ หรือ ซอฟต์แวร์ ที่พัฒนาขึ้น โดยควบคุมระบบของนโยบาย วิธีการปฏิบัติ และอธิบายวิธีการประเมินคุณภาพ เพื่อให้ผลิตภัณฑ์มีคุณภาพและตรงตามความต้องการของลูกค้าและแผนงานที่บริษัทได้กำหนดไว้

## 1.2 Scope

ขอบเขตสำหรับจัดทำแผนงานควบคุมคุณภาพสำหรับโครงการครอบคลุม ขั้นตอนต่างๆดังนี้

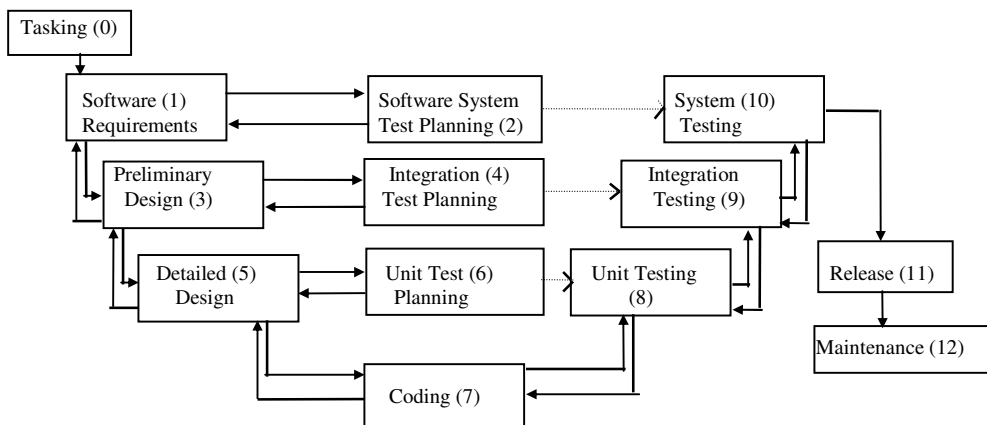


Figure 1.2-1. Software Development Life cycle

- (0) Tasking
- (1) Software Requirements. Software Project/Development Planning
- (2) Software System Test Planning
- (3) Preliminary Design
- (4) Integration Test Planning
- (5) Detailed Design
- (6) Unit Test Planning
- (7) Coding
- (8) Unit Testing
- (9) Integration Testing
- (10) System Testing
- (11) Release
- (12) Maintenance

1. ขั้นตอนของแนวคิดซอฟต์แวร์และการเริ่มต้น

SQA ควรจะเกี่ยวข้องทั้งการเขียนและการตรวจสอบ การจัดการวางแผนเพื่อจะรับประกันซึ่งกระบวนการขั้นตอน และมาตรฐานที่บ่งบอกในแผนการ คือ การจัดสรร ชัดเจน เจาะจง และสามารถตรวจสอบในระหว่างขั้นตอน SQA ให้ QA ในส่วนการจัดการวางแผน

2. ขั้นตอนการกำหนดความต้องการซอฟต์แวร์ และการวางแผนโครงการ

ในกระบวนการการกำหนดความต้องการซอฟต์แวร์ SQA มีหน้าที่ให้ความมั่นใจว่า ความต้องการซอฟต์แวร์นั้นมีความครบถ้วนสมบูรณ์ สามารถทดสอบได้ และตรงความต้องการ

Software Requirements Specification (SRS) เป็นเอกสารที่ใช้ในการกำหนด ความต้องการทางด้านซอฟต์แวร์ อย่างเป็นทางการ โดย ความต้องการของลูกค้าควรจะอยู่ในรูปของ ข้อกำหนดของการทำงาน ขั้นตอนการทำงาน หรือ การบรรยายละเอียดของระบบซอฟต์แวร์ Software System Specification (SSS).

SRS จะถูกตรวจทานโดย Project Leader, Development Manager และ CCB และ SRS ควรจะมีการลงนามโดย Project Manager และ ลูกค้า หลังจากที่ได้ทำการตรวจทานร่วมกันแล้ว

ในกระบวนการการวางแผน สิ่งที่ต้องการคือเอกสารที่ใช้สำหรับการจัดการหมายกำหนดการของโครงการทรัพยากรที่ต้องใช้ รวมไปถึงเวลา และจำนวนบุคลากร ที่ต้องใช้ในการพัฒนา โดยจะถูกนำเสนอใน Software

Project Plan ซึ่ง จะถูกตรวจทานโดย Manager ทีมอื่นๆที่เกี่ยวข้องกับการพัฒนาระบบ รวมไปถึง CCB (Change Control Board)

### 3. ขั้นตอนของการออกแบบสถาปัตยกรรมซอฟต์แวร์เบื้องต้น(

เป็นขั้นตอนของการวิเคราะห์และออกแบบระบบ เอกสารที่ได้จากกระบวนการนี้คือ อ Software Design Document (SDD) และถูกตรวจทานโดย Project Leader, Development Manager และ CCB โดยหลักในการควบคุมคุณภาพประกอบด้วย

- ควบคุมให้อยู่ในมาตรฐานการออกแบบที่กำหนด ตามที่กำหนดในแผนการจัดการ
- ควบคุมให้ซอฟต์แวร์ ออกแบบภายใต้ความต้องการที่ลูกค้ากำหนด และอยู่ใน SRS
- ควบคุมให้ส่วนติดต่อกันอยู่ในข้อตกลงกับมาตรฐานที่กำหนด และตรงตาม SRS
- การตรวจสอบ เอกสาร PRD ( ตรวจสอบรูปแบบเบื้องต้น(และ การรับประกันทุกรายการที่กระทำจะตัดสินใจแล้ว

### 4. ขั้นตอนของรายละเอียดซอฟต์แวร์ออกแบบ

- ควบคุมว่าได้ออกแบบตามมาตรฐานการออกแบบที่กำหนดไว้
- การรับประกันว่าจัดสรรส่วนประกอบรวมอยู่แล้วในรายละเอียดการออกแบบ
- การรับประกันผลลัพธ์ของการตรวจการออกแบบจะรวมอยู่ในการออกแบบ

### 5. ขั้นตอนการทำซอฟต์แวร์ให้สำเร็จ

- ผลลัพธ์ของการเขียนโปรแกรมและออกแบบ รวมถึงแผนงานบรรจุอยู่ในการวางแผนพัฒนาซอฟต์แวร์
- สถานะของงานที่สามารถส่งได้ทั้งหมด
- กิจกรรมการจัดการโครงสร้างและการเก็บรวมการพัฒนาโปรแกรม
- รายงานที่ไม่สอดคล้องกันและระบบการกระทำวิธีที่ถูกต้อง

### 6. ขั้นตอนของการรวบรวมและทดสอบโปรแกรม

- ควบคุมการเตรียมพร้อมสำหรับการทดสอบที่สามารถส่งมอบได้ทั้งหมดทุกเรื่อง
- ควบคุมการทดสอบทั้งหมดจะรันอยู่กับแผนการทดสอบและขั้นตอน ความไม่สอดคล้องใด ๆ ถูกรายงานและตัดสินใจ

- ควบคุมว่ารายงานการทดสอบนั้นเสร็จสิ้นและถูกต้อง
- การได้รับการยอมรับว่า การทดสอบเสร็จสิ้น ซอฟต์แวร์และการจัดเอกสารพร้อมสำหรับส่งมอบ
- การมีส่วนร่วมในการทดสอบความพร้อม ตรวจสอบ และรับประกันการกระทำทั้งหมดจะเสร็จสิ้นสมบูรณ์

7. ขั้นตอนการรับและการส่งมอบซอฟต์แวร์

ในจำนวนที่น้อยที่สุด กิจกรรม SQA ระหว่างขั้นตอนการรับและการส่งมอบซอฟต์แวร์ รวมถึงรับประกันการกระทำของการตรวจสอบโครงสร้างสุดท้ายเพื่อแสดงทั้งหมดทุกเรื่องที่สามารถส่งพร้อมสำหรับการส่ง

8. ขั้นตอนของวิศวกรรมการสนับสนุนซอฟต์แวร์และการปฏิบัติ

ในขั้นตอนนี้ จะเป็นวงจรการพัฒนาเล็ก ๆ เพื่อเพิ่มหรือแก้ไขซอฟต์แวร์ รวมถึงกระบวนการสำหรับการแก้ไขปรับปรุง ดังนี้

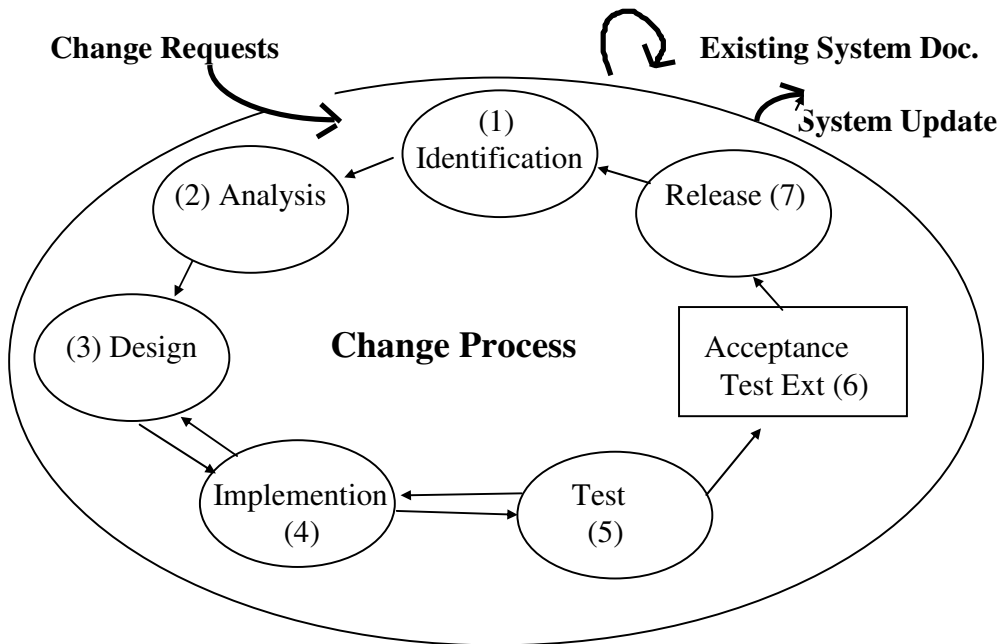


Figure 2.1-2 Software Maintenance Life Cycle



## **1.3 Definitions, Acronyms, and Abbreviations**

### ***1.3.1 Key Acronyms***

BCR -	Baseline Change Request
CCB -	Configuration Control Board
CM -	Configuration Management
CRC -	Change Request Coordinator
CSU -	Computer Software Unit
ITR -	Internal Test Report
PL -	Project Lead
PM -	Program Manager
PVCS -	PC Based source code control system
SCM -	Software Configuration Manager
SCR -	Software Change Request
SE -	Software Engineer
SEPG -	Software Engineering Process Group
SOW -	Statement Of Work
SPR -	Software Problem Report
SQA -	Software Quality Assurance
SQAM-	Software Quality Assurance Manager
SQAP-	Software Quality Assurance Plan
SRS -	Software Requirements Specification
SSS -	Software System Specification
TPR -	Test Problem Report
VDD -	Version Description Document

### 1.3.2 Key Terms

Configuration Management	การควบคุมการเปลี่ยนแปลงโครงการของระบบอย่างเป็นระบบเพื่อคงความบูรณภาพของระบบและสามารถติดตามการเปลี่ยนแปลงของโครงแบบของระบบได้ตลอดวงจรชีวิต
Project deliverables	สิ่งที่จะส่งมอบให้ลูกค้า ซึ่งได้จากการดำเนินโครงการ โดยทั่วไปจะส่งมอบงานเมื่อเสร็จสิ้นงานในแต่ละระยะของโครงการ
Quality assurance	ระบบของนโยบาย วิธีการปฏิบัติ และเครื่องชี้หน้าที่ผู้บริหารใช้รองรับเพื่อให้สินค้าและบริการเป็นไปตามมาตรฐานเฉพาะเจาะจงสอดคล้องกับคุณภาพผลิตภัณฑ์ตามที่กำหนดไว้
Software life cycle	วงจรชีวิตการพัฒนาระบบ
Software quality Management	วิธีการที่จะทำให้มั่นใจได้ว่าผลิตภัณฑ์ซอฟต์แวร์ที่ได้นั้นมีคุณภาพในระดับที่พอใจ

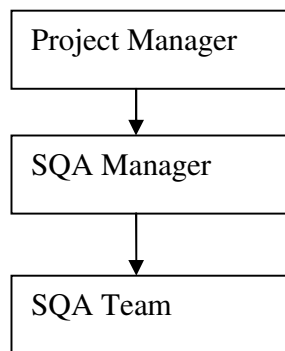
### 1.4 References

- DOD-STD-2167A DOD Std 2167A, Defense System Software Development, 29 Feb 88
- DOD-STD-2168 DOD Std 2168, Defense System Software Quality Program, 29 Apr 88
- MIL-STD-498 Military Standard Software Development and Documentation, 5 Dec 94
- IEEE1028 IEEE Std 1028-1997, Standard for Software Reviews and Audits
- IEEE610 IEEE Std 610.12-1990, Standard Glossary of Software Engineering Terminology (ANSI)
- IEEE730 IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans
- IEEE730.1 IEEE Std 730.1-1995, Guide For Software Quality Assurance Planning
- AFDTC Handbook for Technical Report Preparation, 31 May 92
- Pamphlet 83-2
- [http://wacki.2xprime.com/QMS/Templates/3\\_Defined/RKM/Risk%20Management%20Plan.doc](http://wacki.2xprime.com/QMS/Templates/3_Defined/RKM/Risk%20Management%20Plan.doc)
- กิตติ ภัคดีวัฒนกุล, พนิดา พานิชกุล, วิศวกรรมซอฟต์แวร์ (Software Engineering), สำนักพิมพ์ เค ที พี

## 2. SQA MANAGEMENT

เพื่อให้ SQA มีความเป็นอิสระจากองค์กรอื่น และสามารถจัดเก็บปัญหาทั้งหลายที่มีผลกระทบต่อระดับโครงการ ถ้าคุณภาพของผลิตภัณฑ์ใดที่กำลังมีความเสี่ยงก็จะถูกรายงานความเป็นไปได้โดยตรงไปสู่ผู้บริหารชั้นสูงอีกระดับหนึ่ง SQA สามารถดำเนินการไปในระดับของโครงการที่สูงกว่า งานของ SQA อาจจะได้รับจากบุคคลใดบุคคลหนึ่ง หรือจากองค์กรใดองค์กรหนึ่ง หรือ อาจจะได้รับจากหลายบุคคลหรือหลายองค์กร SQA Manager เมื่อได้รับการมอบหมายอำนาจหน้าที่เรียบร้อยแล้ว จะต้องทำการพิจารณาการจัดตั้งองค์กรของ SQA ขึ้น โดย แบ่ง อำนาจหน้าที่ตามลำดับชั้น

### 2.1 Organization



:

#### 2.1.1 Project Manager

เป็นผู้มอบหมายอำนาจหน้าที่ให้กับ SQA Manager และคอยตรวจสอบดูว่า โครงการได้นำกระบวนการที่ SQA Team กำหนดไปใช้ในโครงการหรือไม่ มีหน้าที่รับผิดชอบดังนี้

- ร่วมในโครงการและประสานงานกับคณะกรรมการโครงการ
- ตรวจสอบงานโดยรวมของโครงการอย่างสม่ำเสมอ
- ตรวจสอบคุณภาพในการดำเนินงานโครงการ
- ควบคุมติดตามการปฏิบัติงานในโครงการให้สอดคล้องกับกระบวนการ Software
- เป็นที่ปรึกษาและร่วมหาแนวทางการแก้ไขปัญหากับทีมที่ปรึกษาโครงการ

### 2.1.2 SQA Manager

เป็นหัวหน้าทีมในการประสานงานและติดตามงานต่างๆ ทั้งจาก ทีมงาน SQA และจากหน่วยงานอื่นๆ ที่เกี่ยวข้อง กับกระบวนการการพัฒนา และติดตามกระบวนการการทำงานให้เป็นไปตามแผนที่กำหนดไว้

### 2.1.3 SQA Team

ทำหน้าที่ในการตรวจสอบ และรายงานผลในการตรวจสอบ และติดตามผล รวมทั้งทำรายงานสรุป แก่ SQM และ หน่วยงานที่ถูกรับการตรวจสอบ

## 2.2 Tasks

Project Manager ควรเข้าใจในแผนการและ Process ที่มีอยู่เพื่อให้พร้อมในการเตรียม Software Product ให้ตรงตามความต้องการและสัญญาที่ได้ให้ไว้กับลูกค้า

SQA จะตรวจสอบ Software Life Cycle ตามรายการต่อไปนี้

- a. เมื่อสิ้นสุดการ Software Requirements and Software System Test Planning จะต้องทำการตรวจสอบ Requirements Review ตาม 5.1.2.1 ของเอกสารนี้
- b. เมื่อสิ้นสุดการ Preliminary Design จะต้องทำการตรวจสอบ Preliminary Design Review ตาม 5.1.2.2
- c. เมื่อสิ้นสุดการ Detailed Design จะต้องทำการตรวจสอบ Critical Design Review ตาม Section 5.1.2.2
- d. เมื่อสิ้นสุดการ Unit Testing จะต้อง walkthrough in accordance ตาม Section 5.1.2.3
- e. เมื่อสิ้นสุดการ Integration Testing จะต้อง Test Review ตาม Section 5.1.2.
- f. เมื่อสิ้นสุดการ System Testing จะต้อง Test Review and an audit ตาม Section 5.1.2.3
- g. เมื่อ Release จะต้อง SQA review โดยใช้ Appendix C.

SQA จะตรวจสอบ Maintenance Life Cycle ตามรายการต่อไปนี้

- a. เมื่อสิ้นสุดการ Analysis จะต้อง Requirements Review ตาม Section 5.1.2.1
- b. เมื่อสิ้นสุดการ Design จะต้อง Preliminary Design Review ตาม Section 5.1.2.2
- c. เมื่อสิ้นสุดการ Implementation จะต้องดำเนินการตาม System Test Plan ตาม Section 5.1.2.3

d. เมื่อสิ้นสุดการ Release จะต้องดำเนินการตาม SQA review using the release checklists ตาม Appendix C.

e. Acceptance Testing ต้องได้รับการยอมรับ ยินยอมลูกค้า ต้องเป็นไปตามเอกสารที่ได้สร้างขึ้นใน phase ต่าง ๆ ของ Project (i.e., the SRS, VDD, User's documentation , etc.).

## 2.3 Responsibilities

มีการกำหนดความรับผิดชอบตามขั้นตอนการพัฒนาาระบบดังนี้

### 2.3.1 PTMR Responsibilities

Table 2.2-1: Software Development Life Cycle

<b>Life Cycle Item:</b>	<b>Review by:</b>	<b>Product:</b>
(0) Tasking	CCB	Minutes Of Meeting
(1) Software Requirements	Payment Team, PM, PL, CCB	SRS SRS Review
(2) Software System Test Planning	Payment Team PM, PL, CCB	Software System Test Plan Software System Test Plan Review
Software Project Plan	Payment Team PM, PL, CCB	Software Project Plan
(3) Preliminary Design	Payment Team PM, PL, CCB	DRAFT Design Document DRAFT Design Document Review
(4) Integration Test Planning	Payment Team PM, PL, CCB	Software Test Plan Software Test Plan Review

(5) Detailed Design	Payment Team PM, PL, CCB	Software Project Plan SPP Review
(7) Coding	PL, SQAM	Audit Documentation
	PL, SQAM	Walkthrough Documentation
(8 - 10) Unit, Integration, System Testing	Payment Team PM, PL, CCB	Software System Test Plan Software System Test Plan Review Documentation
(11) Release	Project Lead	Baseline release to Configuration Control Inspection Report
(12) Maintenance	Refer Below	Refer Below

Table 2.2-2: Software Maintenance Life Cycle

Life Cycle Item:	Review by:	Product:
(1) Identification	PL, PM	CER categorization SCM Audit
(2) Analysis	PL, PM, CCB, Payment Team	CER scheduling Functional Test Review
(3) Design	PL, PM, CCB, Payment Team	Design Review
(4) Implement	PL, PM, SQAM	WalkThrough Report Audit Documentation
(5) Test	PL, PM, SQAM	Test Report Audit Documentation
(7) Release	PL, PM	Staff Meeting Minutes

### 2.3.2 Configuration Control Board (CCB)

รับผิดชอบต่อการอนุมัติ หรือ ปฏิเสธ รวมไปถึง ทบทวน การขอการเปลี่ยนแปลงความต้องการ การเปลี่ยนแปลงกระบวนการในการทำงาน รวมไปถึงการเปลี่ยนแปลงต่างๆ ของโครงการ

### 3. SQA DOCUMENTATION

#### 3.1 Development, Verification and Validation, Use, and Maintenance

- Development (การพัฒนาระบบ) งานหลักก็คือการเขียนโปรแกรมให้ตรงตาม SRS
- Verification (การตรวจสอบความถูกต้อง) เป็นการตรวจสอบความถูกต้องของงานหลังการทำแต่ละขั้นตอนว่าทำถูกต้องหรือไม่ เช่น การตรวจสอบ requirement specification การตรวจสอบความถูกต้องของการเขียนโปรแกรม Verification มักจะหมายถึงเทคนิคที่ไม่ใช้ร่วมกับการรัน
- Validation (การทวนสอบความถูกต้องของระบบกับความต้องการ) เป็นการทดสอบในขั้นสุดท้ายก่อนส่งมอบให้ลูกค้า โดยจุดประสงค์คือการตรวจสอบว่าซอฟต์แวร์มีคุณสมบัติตรงตามความต้องการที่กำหนด (Specification)
- Maintenance การบำรุงรักษาซอฟต์แวร์ ครอบคลุมการปรับปรุง พัฒนาซอฟต์แวร์เพื่อให้สามารถใช้ซอฟต์แวร์ให้เกิดประโยชน์ ตรงตามความต้องการที่เปลี่ยนไปตามกาลเวลา ให้ซอฟต์แวร์ใช้งานได้นานที่สุด

#### 3.2 Control

การควบคุมความถูกต้องของงานเวอร์ชันต่าง ๆ ระหว่างการพัฒนาซอฟต์แวร์ ชิ้นงานจำนวนมากจะถูกสร้างและนำไปประกอบรวมกันเพื่อสร้างเป็นซอฟต์แวร์เวอร์ชันต่าง ๆ การกำหนดเวอร์ชันให้กับชิ้นงาน การควบคุมการแก้ไขปรับปรุงชิ้นงานที่ทีมพัฒนาใช้ร่วมกันเพื่อให้ชิ้นงานมีความถูกต้อง การควบคุมเหล่านี้ จะถูกกำหนดไว้ใน PTMR\_SCM\_Plan.doc

### 4. STANDARD AND PRACTICES

#### 4.1 Coding/Design Language Standards

การพัฒนาระบบ จะต้องอยู่ภายใต้ข้อกำหนด ที่กำหนดไว้ในเอกสาร PTMR\_SCM\_Plan.doc

#### 4.2 Documentation Standards

เอกสารสำหรับกระบวนการ SQA จะต้องอยู่ภายใต้ข้อกำหนด ที่กำหนดไว้ในเอกสาร PTMR\_SCM\_Plan.doc



## 5. Reviews and Audits.

### 5.1 Walkthroughs

#### 5.1.1 Overview

กระบวนการ Reviews and Audits เป็นกระบวนการหนึ่งที่ใช้ในการติดตามความก้าวหน้าในการพัฒนาซอฟต์แวร์ ทาง SQA ได้เลือกที่จะทำการ Review กิจกรรมในแต่ละ Process ที่ทำ โดยมีการใช้ Unitest เข้ามาตรวจสอบการทำงาน รวมถึงการรายงานผลกระทบจากการทำงานในแต่ละ process ว่าสามารถ ทำตามมาตรฐานที่กำหนดไว้หรือไม่

#### 5.1.2 Procedures

##### 5.1.2.1 Requirements Walkthroughs

- ตรวจสอบ (review) เอกสาร URD เพื่อดูว่าเอกสารได้อธิบายแต่ละขั้นของการทำงานของซอฟต์แวร์ตามที่ลูกค้าต้องการ (URD = User Requirement Document)
- ตรวจสอบเอกสาร URD ว่าได้ระบุคุณลักษณะของซอฟต์แวร์ที่ต้องการ ลำดับความสำคัญของคุณลักษณะ และอ้างอิงถึงที่มาของข้อมูล
- ตรวจสอบว่า URD เขียนอธิบายถึงข้อบังคับที่ลูกค้าต้องการ (เช่น Usability, Availability)
- ตรวจสอบ (Audit) กิจกรรมการเก็บข้อกำหนดกับลูกค้าตามกระบวนการที่กำหนด
- ตรวจสอบ SR (Software Requirement) ว่ามีโครงสร้างและเนื้อหาตามมาตรฐาน
- ตรวจสอบ SR ว่าสามารถสอบกลับได้ถึง User Requirements
- ตรวจสอบว่า SR สามารถทวนสอบ (Verify) ได้
- เข้าร่วมการ review เอกสารข้อกำหนดเพื่อดูว่าการ Review เป็นไปตามกระบวนการที่กำหนด

##### 5.1.2.2 Design Walkthrough

- เข้าไปตรวจติดตามเพื่อดูว่า Design ได้ถูกออกแบบตามเทคนิคและวิธีการที่กำหนด
- เข้าไปตรวจติดตามเพื่อดูว่าพนักงานออกแบบได้รับการฝึกอบรมในเทคนิคการออกแบบตามแผนงานการฝึกอบรมของโครงการ

- เข้าไปตรวจสอบ ADD (Architecture Design Document) ว่าได้ระบุคอมโพเนนต์หลักพร้อมทั้ง Function, Inputs และ Outputs ของคอมโพเนนต์หลักเหล่านั้น
- เข้าไปตรวจสอบดูว่า AD อธิบายถึงโครงสร้างของการเชื่อมต่อระหว่างคอมโพเนนต์
- เข้าไปตรวจสอบเอกสาร Detail Design ว่าได้มีการออกแบบเป็นส่วนขยายอย่างมีตรรกะ จากโครงสร้างซอฟต์แวร์ที่ระบุไว้ใน ADD
- เข้าไปตรวจสอบ Design ว่าสามารถสอบกลับไปที่ข้อกำหนดได้
- เข้าไปตรวจสอบเอกสาร Design ว่าได้มีการระบุ Computer Resources ที่ต้องการ
- เข้าร่วมการ Review เอกสาร Design เพื่อดูว่าการ Review เป็นไปตามกระบวนการที่กำหนด

### 5.1.2.3 Test Walkthrough

- เข้าไป review ว่า Unit test level สอดคล้องกับมาตรฐานความปลอดภัยของซอฟต์แวร์
- เข้าไปตรวจติดตามว่า Unit Test แรกเป็น White Box Test
- เข้าไป review Unit Test Plan ว่าครอบคลุมการทดสอบ Branch Coverage
- เข้าไปตรวจติดตามเพื่อดูว่า System Test Specification ได้มีการจัดทำขึ้น
- เข้าไปตรวจติดตามว่า ได้มีการออกแบบ Integration Test สำหรับทุก Interface
- เข้าไป Review ว่าได้มีการกำหนด Test procedure สำหรับแต่ละ Test Case
- เข้าไป Review ว่า System Test ได้รับการทวนสอบเทียบกับ System objectives
- เข้าไปตรวจติดตามเพื่อดูว่า Failed Test ได้รับการทำซ้ำและได้มีการแก้ไขปัญหา

### 5.1.2.4 Coding Walkthroughs

- เข้าไป review ว่า Coding Standard ได้ถูกนำไปใช้ในการ Code
- เข้าไปให้คำแนะนำ เพื่อให้ฝ่ายพัฒนาคำนึงถึงความสำคัญของ Reused software
- ทำการตรวจติดตาม Quality Certification ของ COTS ว่าเป็นไปตามมาตรฐานของโครงการ
- ทำการตรวจติดตามว่า การโค้ด เป็นไปตามมาตรฐานของผลิตภัณฑ์ที่กำหนด
- ทำการตรวจติดตามว่า พนักงานพัฒนา ได้ทำ Code review และปัญหาได้รับการแก้ไข
- เข้าไปสังเกตการณ์การทำ Code review ว่าเป็นไปตามวิธีการและกระบวนการที่กำหนด
- เข้าไปตรวจติดตาม เพื่อดูว่า โค้ดได้ถูกแก้ไข ตามข้อกำหนดที่ได้รับการแก้ไขล่าสุด และสามารถสอบกลับประวัติการแก้ไขของโค้ดนั้นได้
- เข้าไป review โค้ดเพื่อดูว่า Software code สามารถสอบกลับไปที่ แบบและข้อกำหนดได้

## 6. CONFIGURATION MANAGEMENT

ในหัวข้อนี้จะอธิบายเกี่ยวกับ Configuration Management ซึ่งได้อธิบายไว้ในเอกสาร PTRM\_SCM\_Plan.doc และให้คนที่รับผิดชอบในส่วน SQA เข้ามาตรวจสอบผลผลิตของโครงการ รวมทั้งเอกสารของซอฟต์แวร์ก่อนหน้าด้วยสำหรับการให้เหตุผล หลังจากนั้น Project Manager จะเป็นคนตรวจสอบ baseline และเอกสารในการทำ Configuration Management Policies

## 7. TESTING

### 7.1 System Testing

การทดสอบระบบ คือ การทดสอบการทำงานจริงของระบบทั้งระบบในสภาวะแวดล้อมที่ผู้ใช้ปฏิบัติงานจริง

### 7.2 Integration Testing

การทดสอบการรวมระบบ คือ การทดสอบความถูกต้องว่าส่วนต่างๆของซอฟต์แวร์เมื่อทำงานร่วมกันแล้วทำงานได้ถูกต้อง

### 7.3 Unit Testing

การทดสอบส่วนย่อยของระบบ คือ การทดสอบการทำงานในแต่ละส่วนที่พัฒนา.

### 7.4 Test Documentation

PTMR System Developmental Test and Evaluation (DT&E) Test Plan [TBD];

PTMR System Follow-on Operational Test and Evaluation Test Plan [TBD]

## 8. SOFTWARE METRICS

ดูรายละเอียดได้ที่เอกสาร PTMR\_Software Measurement and Metrics.doc

## 9. PROBLEM TRACKING

ให้อ้างอิงกับเอกสาร PTRM\_SCM\_Plan.doc ซึ่งได้อธิบายถึงวิธีการและกระบวนการสำหรับ Reporting, Tracking และ Resolving Problems ที่ถูกระบุไว้ใน Software items และเชื่อมโยงไปยังกระบวนการพัฒนาบำรุงรักษา และความรับผิดชอบขององค์กร

สิ่งที่ขาดหายไป ใน SQA Plan ควรทำให้ถูกต้องโดยกระบวนการดังนี้

- 1) ระบุสิ่งที่ขาดหายไป โดยพนักงาน หรือ SQA Manager จะเป็นผู้ระบุในส่วนนี้
- 2) Project Manager จะระบุส่วนที่ขาดหายไปและวิธีการแก้ไขให้ถูกต้อง โดย Project Manager จะเข้าไปช่วยพนักงานที่ทำผิดหรือไม่ได้ทำตามขั้นตอน
- 3) เอกสาร SQA PLAN ฉบับล่าสุด จะต้องผ่านการตรวจแก้จากบริษัท และถูกเผยแพร่ไปยังทีมพัฒนาระบบ

## 10. RECORDS COLLECTION, MAINTENANCE, AND RETENTION

เอกสาร Software Developmental Library (SDL) สำหรับ สามารถดูได้ใน Appendix A เอกสารที่เกี่ยวข้องกับการพัฒนา configuration จะถูกเก็บรักษาไว้ ดูได้จาก ข้อ 3.1 และวิธีการเข้าถึงเอกสารทุกอย่าง จะถูกควบคุมโดย Project Manager

การเตรียมเอกสารจะเก็บรักษาไว้ในที่เก็บข้อมูลกลางที่กำหนดโดย PTRM\_SCM\_Plan.doc โดยจะมีการ backup ข้อมูลไว้ทุก ๆ สัปดาห์ และเอกสารเหล่านี้ จะถูกเก็บรักษาไว้ตั้งแต่เริ่มสัญญาสร้างระบบ Helpdesk System จนกระทั่งหมดสัญญาจ้าง

เอกสารส่วนนี้อธิบายถึงวิธีการดูแลรักษาระบบในช่วงระยะเวลาของการดูแลรักษาระบบ การบำรุงรักษา So ในส่วนของ server ต้องมีการเข้าไปตรวจสอบการทำงานว่ายังคงปกติหรือไม่เป็นประจำทุกวัน มีการตรวจสอบพื้นที่ว่างของ harddisk มีการ backup ทั้งระบบในส่วน server ทุกวันเก็บลงใน backup disk storage

## 11. TRAINING

PM จะเป็นผู้รับผิดชอบในการอนุญาตให้มีการอบรม เพื่อให้บรรลุผลตาม SQA goal และการพัฒนาโครงการ

จากนั้นก็ทำให้แน่ใจว่าทีมงานทุกคนได้อบรมตามกระบวนการ และ SQA Policies เพื่อนำไปใช้กับการทำงานในทีม ประกอบด้วย การอบรมขั้นต้นให้กับพนักงานใหม่ หรืออบรมตามความต้องการจนกระทั่งเปลี่ยนกระบวนการหรือเจ้าหน้าที่

ในส่วนของการอบรมการเพื่อดูแลบริหารจัดการระบบ Helpdesk มีการแบ่งออกเป็น 2 ส่วนคือ

- ฝึกอบรม user

ในส่วนนี้จะอบรมการใช้งานต่างๆ ที่จำเป็นต่อการใช้งานของ user ได้แก่ การแจ้งปัญหา การดูรายการปัญหาที่เคยส่ง การเปลี่ยนข้อมูลส่วนตัวของ user เป็นต้น

- ฝึกอบรม admin

การฝึกอบรม admin จะอบรมถึงการจัดการระบบทั้งหมด การเข้าไปตั้งค่าให้แก่ตัวระบบเพื่อให้สามารถใช้งานได้ และการแก้ปัญหาเบื้องต้นเมื่อตัวระบบไม่สามารถทำงานได้

## 12. RISK MANAGEMENT

ความเสี่ยงที่อาจเกิดขึ้นเนื่องจากสาเหตุต่างๆ ดังนี้

- ความไม่เข้าใจในเทคนิคที่นำมาใช้ ส่งผลกระทบต่อโครงการปานกลาง โดยที่ทีมบริหารโครงการต้องจัดหลักสูตรอบรมเพิ่มเติม และกำหนดให้ทุกทีมต้องมีการจัดทำรายละเอียดย่อย เพื่อให้ทีมงานสามารถดำเนินการได้อย่างชัดเจนยิ่งขึ้น
- ความต้องการระบบของผู้ใช้เปลี่ยนไป
- ความผิดพลาดของการวิเคราะห์และออกแบบ สืบเนื่องมาจากปัญหาในข้อ .1 ด้วยประการหนึ่ง ทำให้ทีม Implementation ไม่สามารถพัฒนาโปรแกรมได้ ส่งผลกระทบต่อโครงการอย่างมาก ทีมบริหารต้องจัดทีมการทำ UML Diagram เป็นทีมเฉพาะเพื่อลดความขัดแย้ง และมีเวลาการกรออกแบบเต็มที่

## **APPENDIX A // PTRM Technical Documentation**

Each Project in development must have the appropriate life-cycle documentation. This documentation is to include:

- Tasking Statement(s)
- Software Requirements Specification
- Software Project Plan
- Software System Test Plan
- Preliminary Design Documentation
- Detailed Design Documentation
- Unit Test Plan

Please refer to the PTRM SCM Plan for information regarding deliverable documentation.

**APPENDIX B// Walkthrough and Inspection Forms**

**B.1.1) Requirements Walkthrough Form (v 1.1)**

**Software Project: PTRM**

**Software Item(s): SRS**

**Review Date: 08/08/2009**

**Reviewer(s): Sirikanya Yomkoet**

Moderator please respond to the following:

	Yes	No
1) Is there a clear understanding of the problem which this system is designed to solve?	<u>  X  </u>	<u>      </u>
2) Are the external and internal interfaces properly defined?	<u>  X  </u>	<u>      </u>
3) Are all the requirements traceable to the system level?	<u>  X  </u>	<u>      </u>
4) Is prototyping conducted for customer?	<u>      </u>	<u>  X  </u>
5) Is the performance achievable with constraints imposed by other system elements?	<u>  X  </u>	<u>      </u>
6) Are schedule, resources, and budget consistent with requirements?	<u>  X  </u>	<u>      </u>
7) Are validation criteria complete?	<u>  X  </u>	<u>      </u>

\* If answer 'No' to any of the above please attach problems found and suggestions to improve.

Please attach additional comments and copy of requirement items

**B.1.2) Software Project Plan Walkthrough Checklist**

**Software Project: PTRM**

**Software Item(s): Project Plan**

**Review Date: 09/08/2009**

**Reviewer(s): Thawatchai Yernyoob**

Moderator please respond to the following:

	Yes	No
1) Is the software scope unambiguously defined and bounded?	__X__	_____
2) Is the terminology clear?	__X__	_____
3) Are the resources adequate for scope?	__X__	_____
4) Are the resources readily available?	__X__	_____
5) Are the tasks properly defined and sequenced?	__X__	_____
6) Is task parallelism reasonable given the available resources?	__X__	_____
7) Have both historical productivity and quality data been used?	__X__	_____
8) Have any differences in estimates been reconciled?	__X__	_____
9) Are the pre-established budgets and deadlines realistic?	__X__	_____
10) Is the schedule consistent?	__X__	_____

\* If answer 'No' to any of the above please attach problems found and suggestions to improve.

Please attach additional comments and copy of requirement items



**B.1.3) Preliminary Design Walkthrough Checklist**

**Software Project: PTRM**

**Software Item(s):SRS**

**Review Date: 09/08/2009**

**Reviewer(s): Thawatchai Yernyoob**

Moderator please respond to the following:

	Yes	No
1) Are software requirements reflected in the software architecture?	__X__	_____
2) Is effective modularity achieved? Are modules functionally independent?	__X__	_____
3) Is the program architecture factored?	__X__	_____
4) Are interfaces defined for modules and external system elements?	__X__	_____
5) Is the data structure consistent with information domain?	__X__	_____
6) Is the data structure consistent with software requirements?	__X__	_____
7) Has maintainability been considered?	__X__	_____

\* If answer 'No' to any of the above please attach problems found and suggestions to improve.

Please attach additional comments and copy of requirement items

**B.1.4) Detailed Design Walkthrough Checklist**

**Software Project:**

**Software Item(s):**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Review Date:** \_\_\_\_\_

**Reviewer(s):**

_____	_____
_____	_____
_____	_____

Moderator please respond to the following:

	Yes	No
1) Does the algorithm accomplish the desired function?	_____	_____
2) Is the algorithm logically correct?	_____	_____
3) Is the interface consistent with the architectural design?	_____	_____
4) Is logical complexity reasonable?	_____	_____
5) Has error handling been specified and built-in?	_____	_____
6) Is local data structure properly defined?	_____	_____
7) Are structured programming constructs used throughout? _____	_____	_____
8) Is design detail amenable to the implementation language?	_____	_____
9) Which are used: operating system or language-dependent features?	_____	_____
10) Has maintainability been considered?	_____	_____

\* If answer 'No' to any of the above please attach problems found and suggestions to improve.

Please attach additional comments and copy of requirement items

**B.1.5) Program Code Walkthrough Checklist**

Note: the following checklist assumes that a design walkthrough has already been done.

**Software Project:**

**Software Item(s):**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Review Date:** \_\_\_\_\_

**Reviewer(s):**

_____	_____
_____	_____
_____	_____

Moderator please respond to the following:

	Yes	No
1) Is design properly translated into code?	_____	_____
2) Are there misspellings or typos?	_____	_____
3) Has proper use of language conventions been made?	_____	_____
4) Is there compliance with coding standards for language style, comments, module prologue?	_____	_____
5) Are incorrect or ambiguous comments present?	_____	_____
6) Are typing and data declaration proper?	_____	_____
7) Are physical contents correct?	_____	_____
8) Have all items on the design walkthrough checklist been reapplied (as required)?	_____	_____

\* If answer 'No' to any of the above please attach problems found and suggestions to improve.

Please attach additional comments and copy of requirement items

**B.1.6) Test Plan Walkthrough Checklist**

**Software Project:**

**Software Item(s):**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Review Date:** \_\_\_\_\_

**Reviewer(s):**

_____	_____
_____	_____
_____	_____

Moderator please respond to the following:

	Yes	No
1) Have major test activities been properly identified and sequenced?	_____	_____
2) Has traceability to validation criteria/requirements been established as part of software requirements analysis?	_____	_____
3) Are major functions demonstrated early?	_____	_____
4) Is the test plan consistent with the overall project plan?	_____	_____
5) Has a test schedule been explicitly defined?	_____	_____
6) Are test resources and tools identified and available?	_____	_____
7) Has a test record keeping mechanism been established?	_____	_____
8) Have test drivers and stubs been identified, and has work to develop them been scheduled?	_____	_____
9) Has stress testing for software been specified?	_____	_____

\* If answer 'No' to any of the above please attach problems found and suggestions to improve.

Please attach additional comments and copy of requirement items

**B.1.7) Walkthrough Report Summary Sheet**

Project Name:	Date:
Element Reviewed:	
Review Team:	
Problems Found:	
Proposed Solutions:	
Comments:	
Action Taken:	

## **B.2) Inspection Forms**

This section describes various forms provided for project use. Please modify them where necessary to be more meaningful to your particular software product. When modifying forms please contact the [PROJECT ABBREVIATION] SQA representative or [PROJECT ABBREVIATION] Program Manager for guidance.

### ***SQA Inspection Log***

The SQA Inspection Log form is used by an inspector during his/her review of the inspection package materials. It is also used by the recorder during an inspection. In preparing for an inspection, an inspector uses this form to identify all defects found. During an inspection, the recorder uses this form to record all official defects identified.

### ***SQA Inspection Report***

The SQA Inspection Report form is used by the moderator to officially close the inspection. Once the inspection team has agreed to a disposition on the inspection, the moderator completes this form and signs off on it.



## B.2.2) Inspection Log Description

Log Number - The log or tracking number assigned to this inspection

Date - During an inspection, the recorder enters the date that the inspection occurred. The Date block can be ignored by an inspector using this form during the pre-inspection review..

Software Item - The name of the software item being inspected

Inspector - The Inspector block can be left blank when used by the recorder during an inspection. When using this form during a pre-inspection review, the inspector name is entered in this block

Moderator - The moderator's name

Author - The name of the author responsible for the inspection package is entered in this block

Review Type - The type of inspection being conducted is entered in this block

Category - A category is a two character code that specifies the class and severity of a defect. A category has the following format - XY, where X is the severity (**M**ajor/ **m**Inor) and Y is the class.

Severity:

**M**ajor = a defect that would cause a problem in program operation

**m**Inor = all other defects

Class:

**M**(issing) = required item is missing from software element

**S**(tandards Compliance) = non-conformance to existing standards

**W**(rong) = an error in a software element

**E**(xtra) = unneeded item is included in a software element

**A**(mbiguous) = an item of a software element is ambiguous

**I**(nconsistent) = an item of a software element is inconsistent





#### **B.2.4) Inspection Log Defect Summary Description**

Log Number - The recorder will enter the log number assigned to this inspection

Date - The recorder will enter the date of the inspection in this block

Software Item - The recorder will enter the name of the software item that is being inspected

Moderator - The name of the inspection moderator

Inspection Type - The recorder will enter what type of inspection is being conducted in this block

Defects Found - The recorder will record counts of all defects found during review of the inspection package. The defect counts will be recorded in the table as follows:

- The Insp. column will contain name of inspector
- The Major Defects columns (M,S,W,E,A, and I) will hold identifiers that correspond to the number of major defects found for a specific type in one of the six possible classes. As an example: if the 'W' column held the number 3 then it would mean that there were 3 major defects found that were classed as Wrong.
- The first Total column will hold a total of all the major defects found for a type.
- The Minor Defects columns (M,S,W,E,A, and I) will hold numbers that correspond to the number of minor defects found for a specific type in one of the five possible classes. As an example: if the 'A' column held the number 3 then it would mean that there were 3 minor defects found that were classed as Ambiguous.
- The second Total column will hold a total of all the minor defects found for a type.

### B.2.5) Inspection Report

Log Number: \_\_\_\_\_ Date: \_\_\_\_\_

Software Item: \_\_\_\_\_

Inspection type: \_\_\_\_\_ Duration: \_\_\_\_\_

Number of inspectors: \_\_\_\_\_ Total prep time: \_\_\_\_\_

Size of materials inspected: \_\_\_\_\_

\_\_\_\_\_

Disposition: Accept \_\_\_\_ Conditional \_\_\_\_ Reinspect \_\_\_\_

Estimated rework effort: \_\_\_\_\_ (Hours)

Rework to be completed by : \_\_\_\_\_

Reinspection scheduled for: \_\_\_\_\_

Inspectors:

\_\_\_\_\_  
\_\_\_\_\_

Author(s):

\_\_\_\_\_

Recorder:

\_\_\_\_\_

Moderator:

\_\_\_\_\_

Moderator certification: \_\_\_\_\_ Date: \_\_\_\_\_

Additional comments:

\_\_\_\_\_

### **B.2.6) Inspection Report Description**

Log Number - The log or tracking number assigned to this inspection

Date - The date of the inspection

Software Item - The name of the software item being inspected

Inspection Type - The type of inspection being conducted

Duration - The duration of the inspection

Number of Inspectors - The number of inspectors present during the inspection

Total Prep Time - The total preparation time spent in preparation for the inspection

Size of Materials - The size of the materials inspected

Disposition - The disposition of the inspection

Estimated Rework Effort - The estimated rework effort time, in hours

Rework to Be Completed By - The estimated date of rework completion. If no rework is needed enter 'N/A'

Reinspection Scheduled For - The date for a reinspection of the software item to ensure rework completeness. It is the moderator's responsibility to decide if a reinspection is needed. If it is not needed this block contains 'N/A'

Inspectors - The names of the inspectors present at the inspection

Author(s) - The name(s) of the author(s) responsible for the software item

Recorder - The name of the recorder present at the inspection

Moderator - The moderators name in this block

Moderator Certification - The moderator will sign and date this block to signal that the inspection is concluded

Additional Comments - The moderator will enter additional comments as necessary for clarification.

Authorized By:  ( Sirikanya Yomkoet )	Received By:  ( Nattawut Riyasan )	Reviewed By:  ( Jirasith Srisupho )	Reviewed By:  ( Thawatchai Yernyoob )
Authorized Date: 02/08/2008	Received Date: 02/08/2008	Reviewed Date: 05/08/2008	Reviewed Date: 05/08/2008
*Project Leader	*Tester or CM	*Project Manager	*Configuration Manager