

บทที่ 4

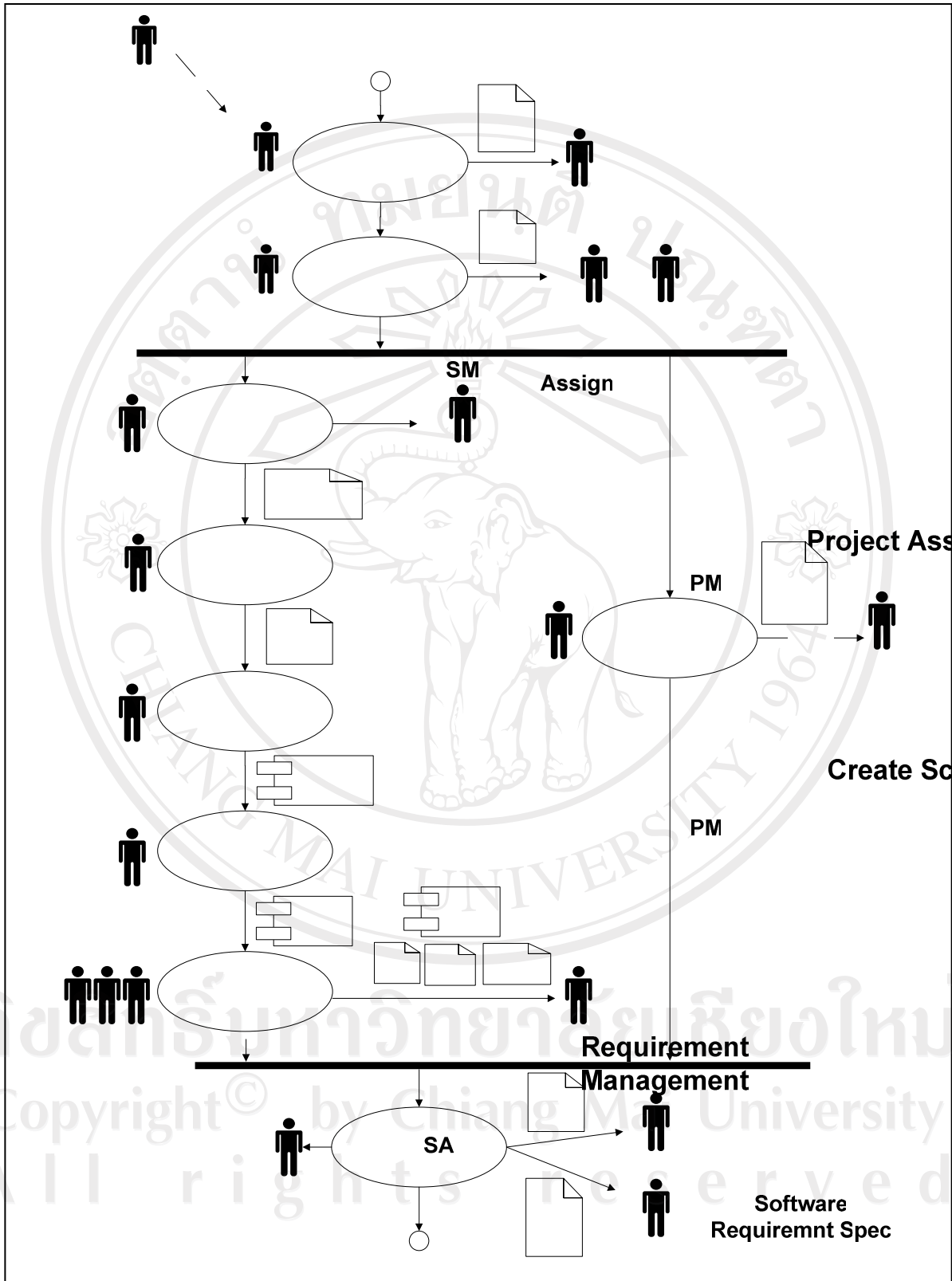
การออกแบบกระบวนการพัฒนาซอฟต์แวร์

การออกแบบกระบวนการพัฒนาซอฟต์แวร์ของบริษัท เมินชา ซอฟต์แวร์ แอนด์ คอนซัลติ้ง จำกัด โดยอาศัยผลการประเมินสมรรถนะของบริษัทก่อนการทำ CMM ทำให้ได้กระบวนการพัฒนาซอฟต์แวร์ของบริษัทที่อิงกับมาตรฐาน CMM Level 2 ซึ่งในที่นี้จะเรียกว่า MSSC Standard Process โดยได้แบ่งออกเป็น 6 ส่วน คือ ส่วนของ Software Development Life Cycle (SDLC) ของบริษัท, Software Project Planning, Software Project Tracking and Oversight, Requirement Management, Software Quality Assurance, Software Configuration Management

4.1 Software Development Life Cycle (SDLC)

Software Development Life Cycle หรือวงจรการพัฒนาระบบ ของบริษัทเมินชา ซอฟต์แวร์ แอนด์ คอนซัลติ้ง ได้มีการปรับเปลี่ยนให้ขั้นตอนการทำงานในวงจรการพัฒนาระบบมีขั้นตอนการทำงานและเอกสารที่ชัดเจนและเป็นแบบแผนมากขึ้น

โดย SDLC แบบเดิม สามารถนำมาเขียนขั้นตอนการทำงานได้ดังรูป 4.1



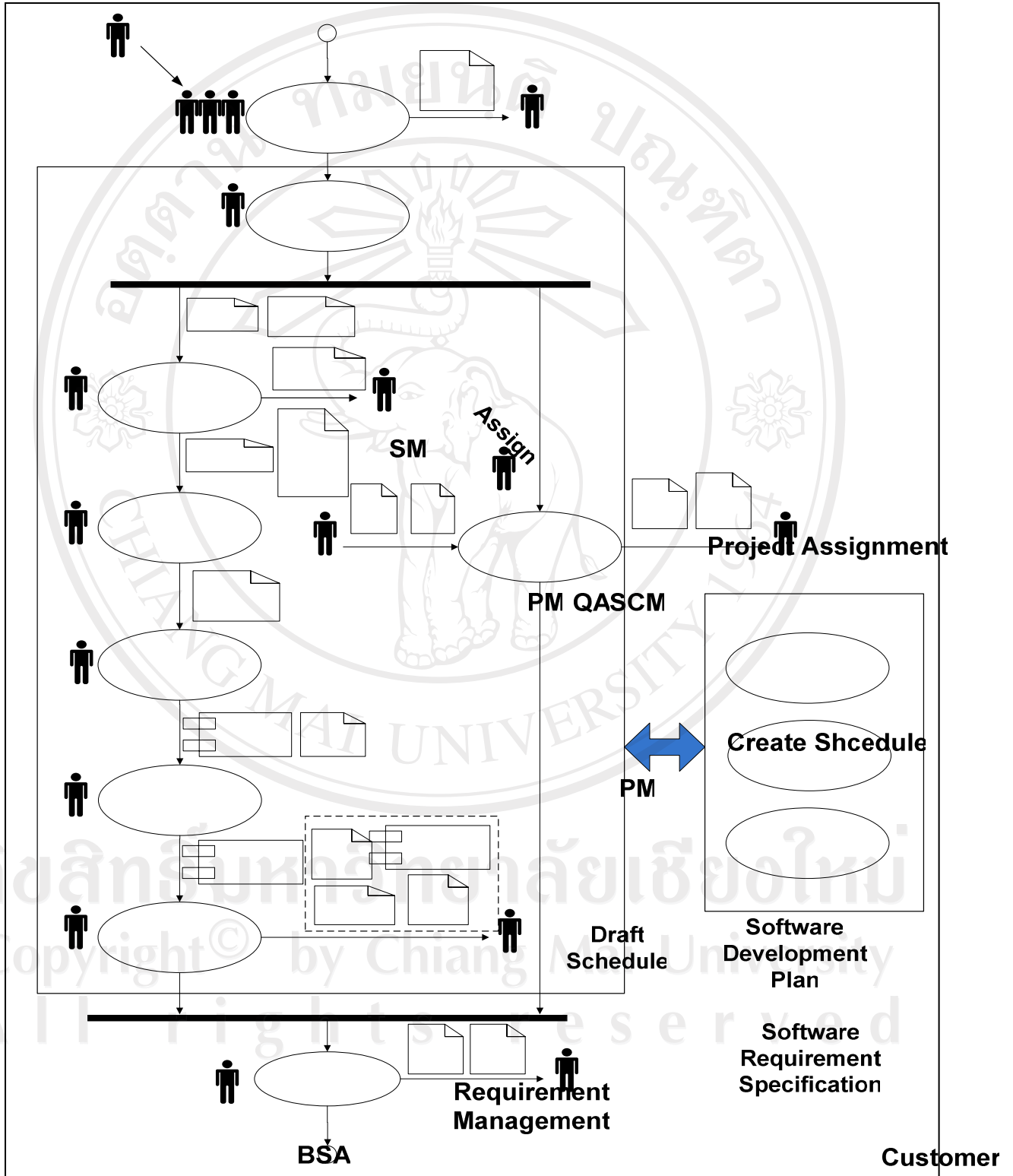
รูป 4.1 Software Development Life Cycle (รูปแบบเดิม)

Analysis & Design

SA

Analysis
Design
Document

และส่วนของ SDLC แบบใหม่ที่ได้ทำการปรับปรุงแล้ว สามารถเขียนขั้นตอนการทำงานได้ดังรูป 4.2



รูป 4.2 Software Development Life Cycle (รูปแบบใหม่)
Use-Case Specification

จากรูป 4.2 จะเห็นได้ว่าการเพิ่มกระบวนการทำงานในส่วน of Software Quality Assurance, Software Configuration Management เข้ามาในระบบการทำงานเพื่อให้เป็นไปตามมาตรฐานของ CMM Level 2 ซึ่งจะได้อธิบายในหัวข้อที่ 4.5 และ 4.6 ต่อไปตามลำดับ

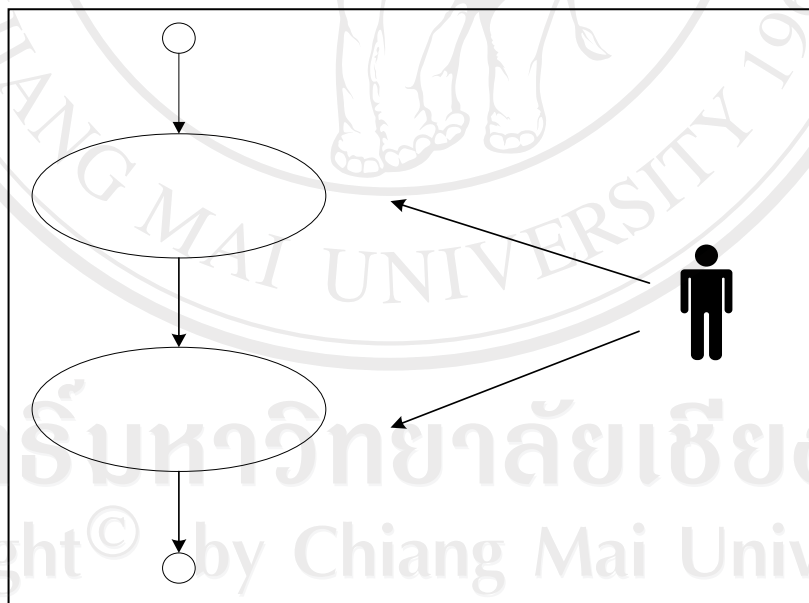
ส่วนขั้นตอน Project Assignment, Create Schedule และ Close-Out Project นั้นเป็นส่วนหนึ่งของ Software Project Planning และ Tracking and Monitoring เป็นส่วนหนึ่งของ Software Project Tracking and Oversight ซึ่งจะได้อธิบายในหัวข้อที่ 4.2 และ 4.3 ต่อไปตามลำดับ

ส่วนขั้นตอน Requirement Management เป็นส่วนของ Requirement Management ซึ่งจะได้อธิบายในหัวข้อที่ 4.4 ต่อไป

ส่วนขั้นตอน Analysis & Design, Implement, Test, Deployment จะอธิบายกระบวนการทำงานดังต่อไปนี้

4.1.1 Analysis & Design

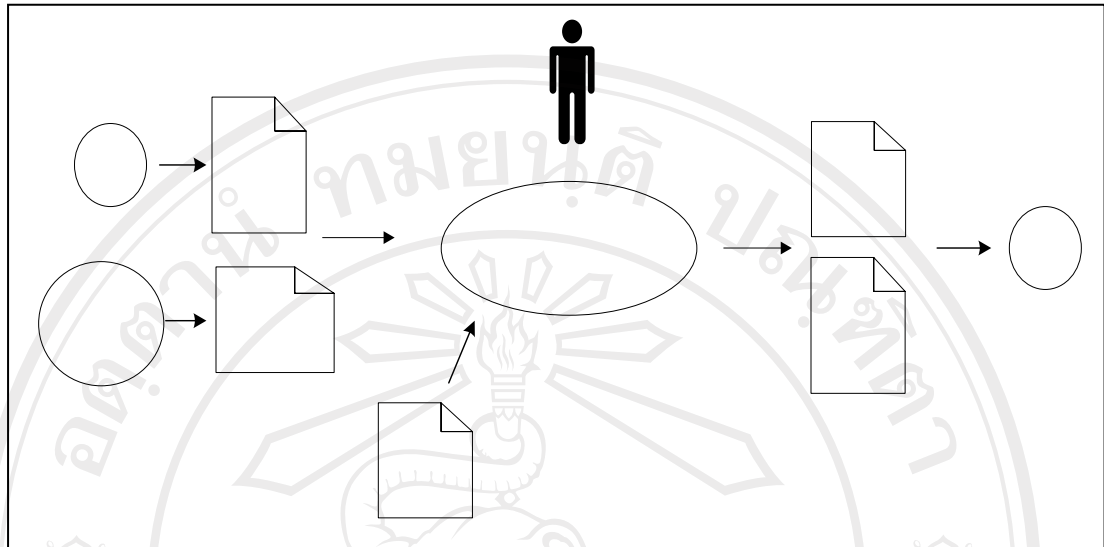
ส่วนนี้จะอธิบายถึงการทำงานของ SA (System Analyst) ในการวิเคราะห์ ออกแบบระบบตามความต้องการของลูกค้า และส่งข้อมูลการออกแบบให้กับ Programmer ทำการ Implement ต่อไป ซึ่งมีขั้นตอนการทำงานดังรูป 4.3



รูป 4.3 Analysis & Design Workflow

จากรูป 4.3 สามารถอธิบายรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.4-4.5

Define a Completed Architecture and Analyze

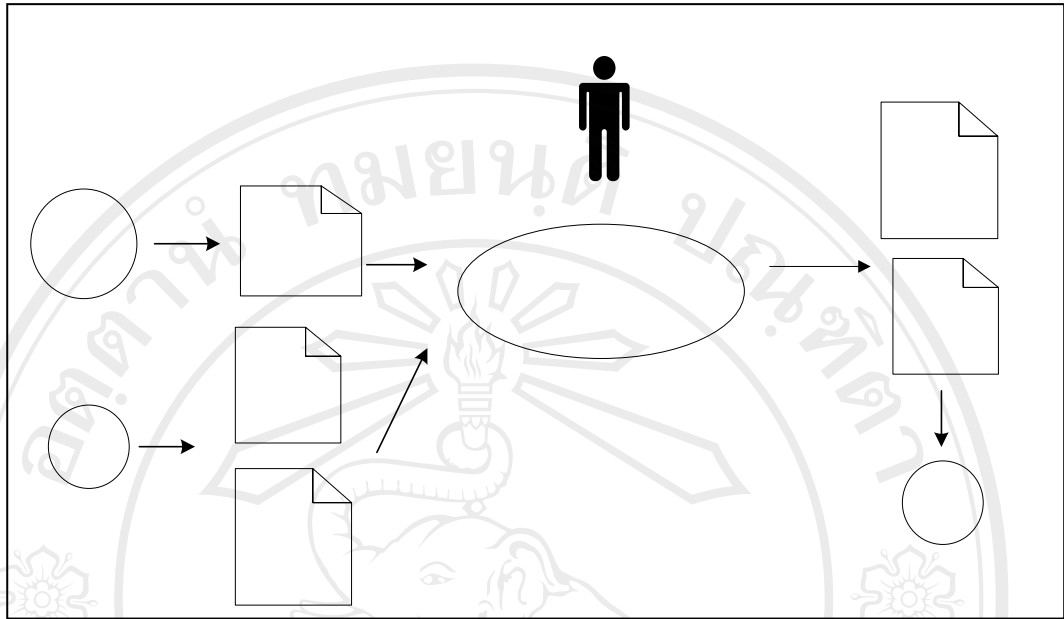


รูป 4.4 Define a Completed Architecture and Analyze Artifacts Flow

จากรูป 4.4 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนมีจุดประสงค์เพื่อร่างสถาปัตยกรรมของระบบ โดยการกำหนดส่วนประกอบสำคัญและนำมาวิเคราะห์และกำหนดโครงสร้างของระบบ และเพื่อกำหนด class ที่สำคัญจาก Use-case เพื่อทำการวิเคราะห์พฤติกรรม (Class, Properties) ของระบบ ซึ่งมีขั้นตอนการทำงานดังนี้

1. ขั้นตอนนี้ SA นำเอา Software Architecture Document ที่ได้จาก **Process: Analyze/Reanalyze the Project Baseline Use Case Specification** มาใช้ในการกำหนดสถาปัตยกรรมระบบ ซึ่งจะเป็นการเพิ่มเติมรายละเอียดในส่วนของ Logical View และ Process View
2. SA ทำ Analysis & Design ที่เหมาะสมกับ Project โดยใช้วิธีการ Applying UML and Patterns
3. SA นำเอกสาร Software Architecture Document มาทำการสร้าง Analysis and Design Document ให้เสร็จในส่วนของการ **Analysis & Design Document Template**
4. SA ส่ง Analysis and Design Document และ Software Architecture Document ให้ PM ทำการ Review
5. SA ส่ง Software Architecture Document ให้ SCM ทำการจัดเก็บและควบคุม

Design



รูป 4.5 Design Artifacts Flow

จากรูป 4.5 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการ
 ออกแบบส่วนประกอบต่างๆ ของระบบ (Class, Interface, Subsystem) ซึ่งมีขั้นตอนการทำงาน
 ดังนี้

1. SA นำเอกสาร Use-Case Specification มาทำการเพิ่มเติม Analysis and Design Document ให้เสร็จ ในส่วนของหัวข้อ Phase Design ให้เสร็จสมบูรณ์
2. ถ้าระบบมีการเกี่ยวข้องกับฐานข้อมูลก็ต้องการสร้าง Data Model และความสัมพันธ์ต่างๆ ใน Analysis and Design Document
3. SA ตั้ง Software Architecture Document และ Analysis and Design Document ให้ PM ทำการ Approve
4. SA ตั้ง Software Architecture Document และ Analysis and Design Document ให้ SQA ทำการ Review
5. SA ตั้ง Software Architecture Document และ Analysis and Design Document ให้ SCM ทำการจัดเก็บและควบคุม
6. PM ขอ Setup Baseline กับ SCCB โดยจะประกอบไปด้วย
 - Configuration Items ที่ถูก Setup Baseline ล่าสุด
 - Software Architecture Document

Baseline

Use-Case Specification

Analysis & Design Document (Analysis)

Software Architecture Document

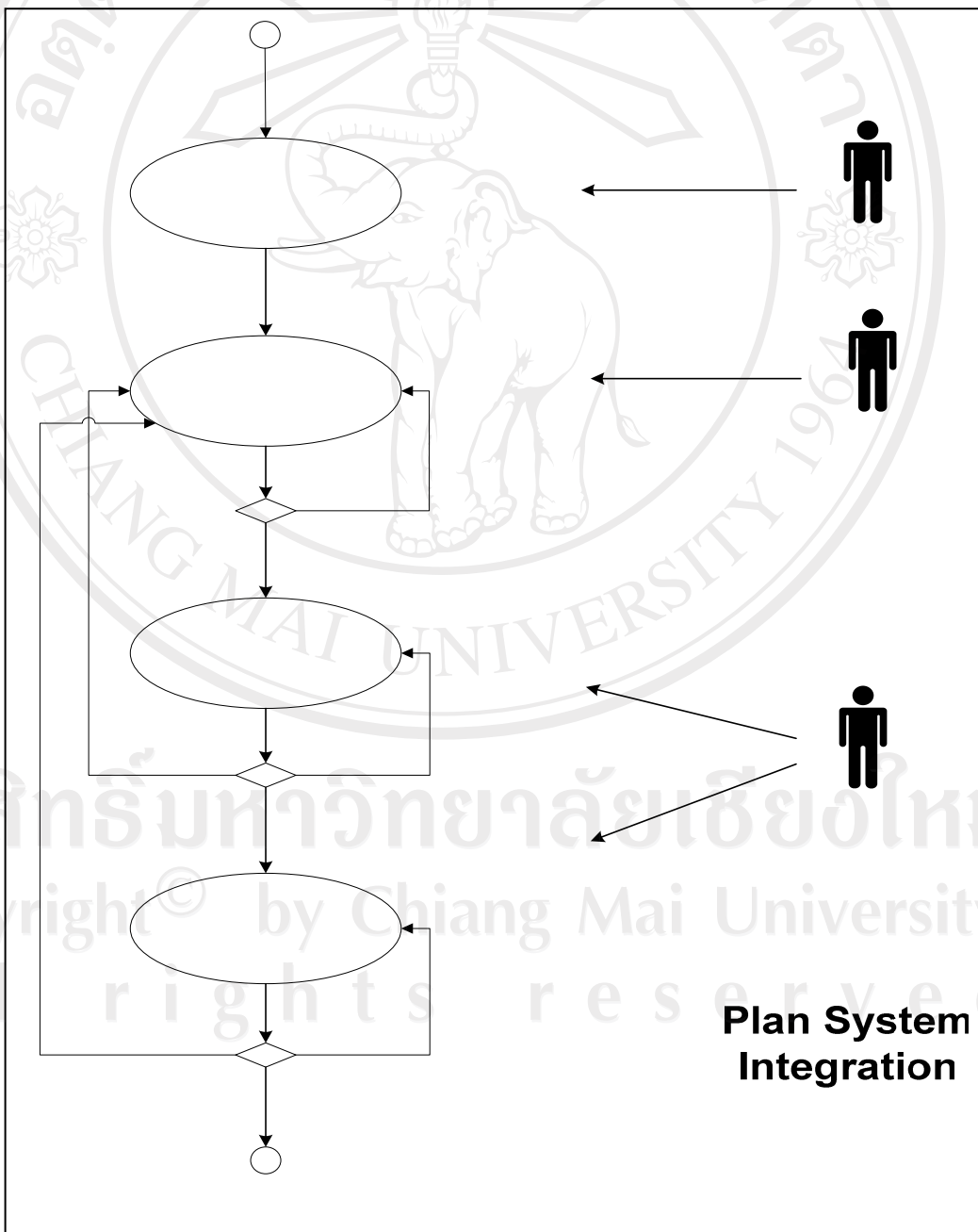
SCM

ลิขสิทธิ์ © by Chiang Mai University
 All rights reserved

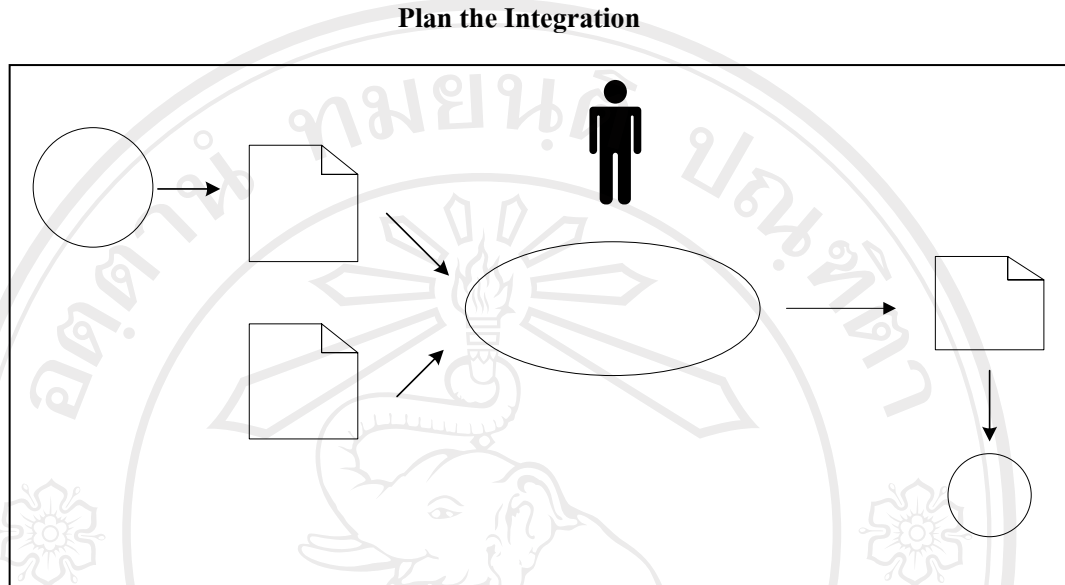
- Analysis and Design Document
- Deployment Plan (จาก **Process: Create/Revise Deployment Plan**)

4.1.2 Implement

ส่วนนี้จะอธิบายถึงการทำงานของ Programmer ในการ Coding โปรแกรมตามที่ SA ได้ทำการออกแบบระบบไว้ จากนั้น Programmer จะทำการส่งโปรแกรมให้กับ Tester ทำการ Test ต่อไป ซึ่งมีขั้นตอนการทำงานดังรูป 4.6



จากรูป 4.6 สามารถอธิบายรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.7-4.10

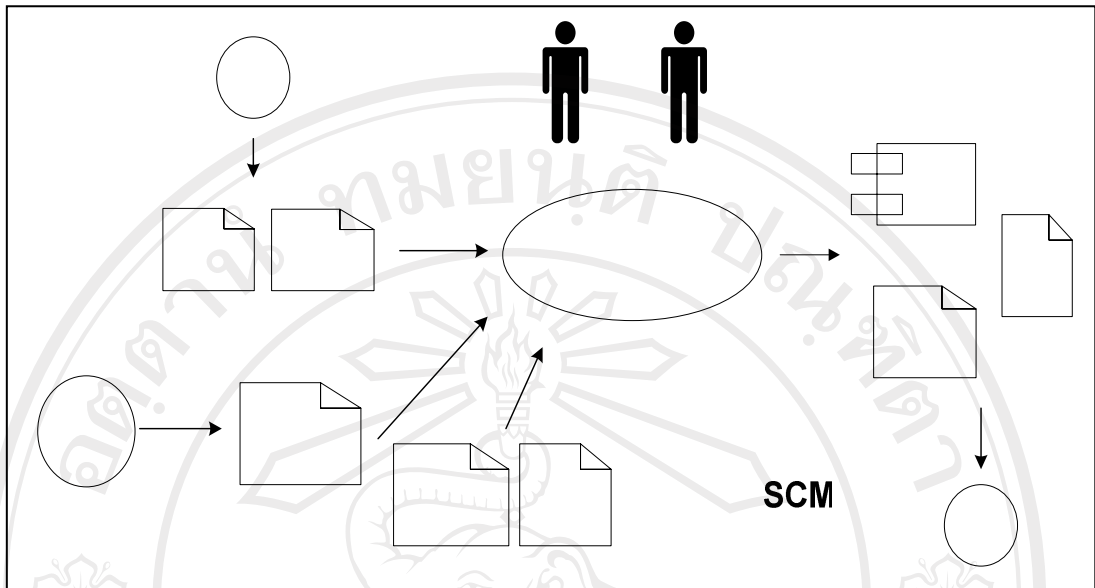


รูป 4.7 Plan the Integration Artifacts Flow

จากรูป 4.7 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อวางแผนว่า subsystem ใดควรจะถูก implement และนำมาประกอบเข้าด้วยกัน ซึ่งมีขั้นตอนการทำงานดังนี้

1. SA ทำการวางแผนสำหรับการ Integrate System โดยดูข้อมูลจาก **Analysis & Design Document** จะได้ **Integration Build Plan** โดยใช้ **Integration Build Plan Template**
2. SA ส่ง **Integration Build Plan** ให้ PM ทำการ Approve
3. SA ส่ง **Integration Build Plan** ให้ SQA ทำการ Review
4. SA ส่ง **Integration Build Plan** ให้ SCM เพื่อทำการจัดเก็บและควบคุม
5. ในกรณี Project นั้นต้องใช้ Database SA นำเอา **Analysis and Design Document** มาทำการสร้าง Database เตรียมไว้ให้ Programmer

Implement Components



รูป 4.8 Implement Components Artifacts Flow

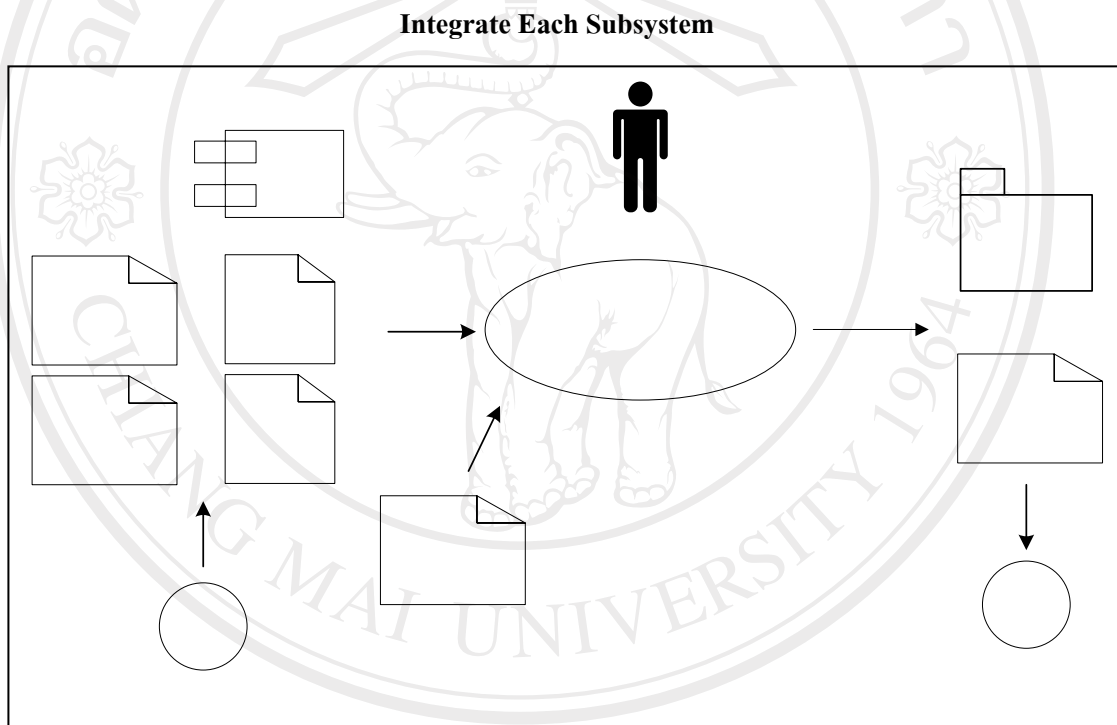
จากรูป 4.8 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการ Coding และ Unit Test ซึ่งมีขั้นตอนการทำงานดังนี้

1. Programmer นำเอาเอกสาร Analysis and Design **Test Log** จาก **Baseline** มาทำการ Coding และ Programmer ทำการ Coding Component ตามลำดับก่อนหลัง ตามที่ระบุไว้ใน **Integration Build Plan** เพื่อให้ได้ Component ออกมา โดย Component จะถูกเก็บอยู่ที่ CM Repository ตามที่ SCM ได้จัดเตรียมไว้ให้
2. Programmer ทำ Unit Test กับ Component ที่ถูกเก็บไว้ใน CM Repository โดยนำ Test Log และ Unit Test Template มาประกอบการทำ Unit Test Programmer จะบันทึกผลลัพธ์จากการทำ Unit Test ลงใน **Baseline**
3. Programmer ทำการสร้าง Delivery Document จาก Delivery Document Template เพื่อแจ้งให้ SA ทราบว่ามี Component ใดบ้างที่ถูกทำการ Implement เสร็จและทำ Unit Test เรียบร้อยแล้วพร้อมให้ SA ทำการ Review Component ที่อยู่ใน CM Repository
4. Programmer ส่ง Delivery Document และ Unit Test ให้ SA เพื่อทำการ Review
5. SA ได้รับ Delivery Document และ Unit Test จาก Programmer SA ทำการตรวจ Source Code ของ Component ที่ถูกจัดเก็บใน CM Repository และนำเอา Test Log มาประกอบการตรวจ และบันทึกผลลัพธ์จากการตรวจลงใน Unit Test

Analysis & Design Document

Delivery Document Template

6. ในกรณีที่พบ Bug เกิดขึ้น SA จะส่ง Unit Test ให้ Programmer ทำการแก้ไข Component นั้น โดย Programmer จะนำ Component จาก CM Repository มาทำการแก้ไข
7. Programmer ทำการ Implement Component ให้เสร็จจนสามารถ นำมาทำการ Integrate เป็น Subsystem ได้ จากนั้นจึงไปทำ **Process: Integrate each Subsystem** ต่อไป โดย Component ที่จะนำไปทำการ Integrate เป็น Subsystem นั้นจะถูกจัดเก็บใน CM Repository ถ้ายังทำการ Implement Component ยังไม่เสร็จให้กลับไปทำ Implementation Components ให้เสร็จ



รูป 4.9 Integrate Each Subsystem Artifacts Flow

จากรูป 4.9 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการ Implement Component ให้เป็น Subsystem ซึ่งมีขั้นตอนการทำงานดังนี้

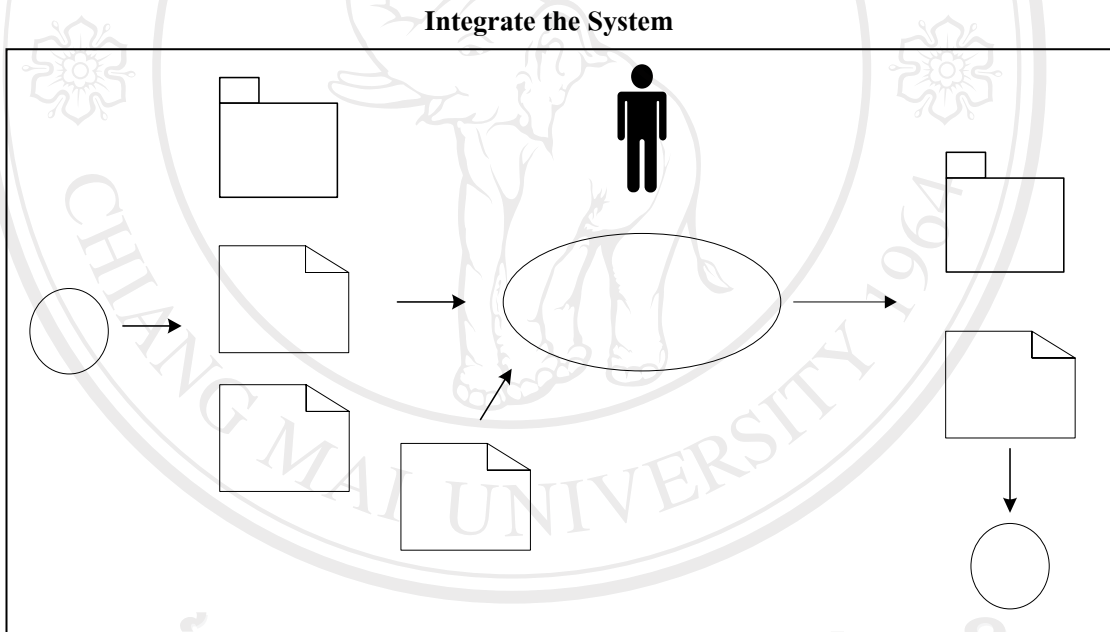
1. SA นำเอา Component จาก CM Repository มาทำการ Integrate เป็น Subsystem
2. ถ้า SA ไม่สามารถทำการ Integrate เป็น Subsystem ได้ SA ส่ง Unit Test คืนให้ Programmer และ Programmer จะนำเอา Component จาก CM Repository มาทำการแก้ไข ย้อนกลับไปทำ **Process: Implement Components**

Component

**Delivery
Document**

Unit Test

3. SA นำ Delivery Document Template มาสร้าง Delivery Document เพื่อแสดงว่าตอนนี้ได้ทำการ Integrate Subsystem เรียบร้อยแล้ว
4. SA ส่ง Delivery Document ให้ PM Approved
5. SA ส่ง Delivery Document ให้ Tester เพื่อแจ้งให้ Tester ทราบว่า SA ได้ Integrate Subsystem เรียบร้อยแล้ว
6. ในกรณีที่ Tester ทำการ Test Subsystem ไม่ผ่าน จาก **Process: Execute Test in Subsystem** SA จะได้รับ Test Log จาก Tester SA จะนำเอา Test Log ส่งให้ Programmer Programmer นำเอา Component ที่อยู่ใน CM Repository มาทำการแก้ไข ย้อนกลับไปทำ **Process: Implement Components**



รูป 4.10 Integrate the System Artifacts Flow

จากรูป 4.10 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการ Integrate Subsystem เป็น System ซึ่งมีขั้นตอนการทำงานดังนี้

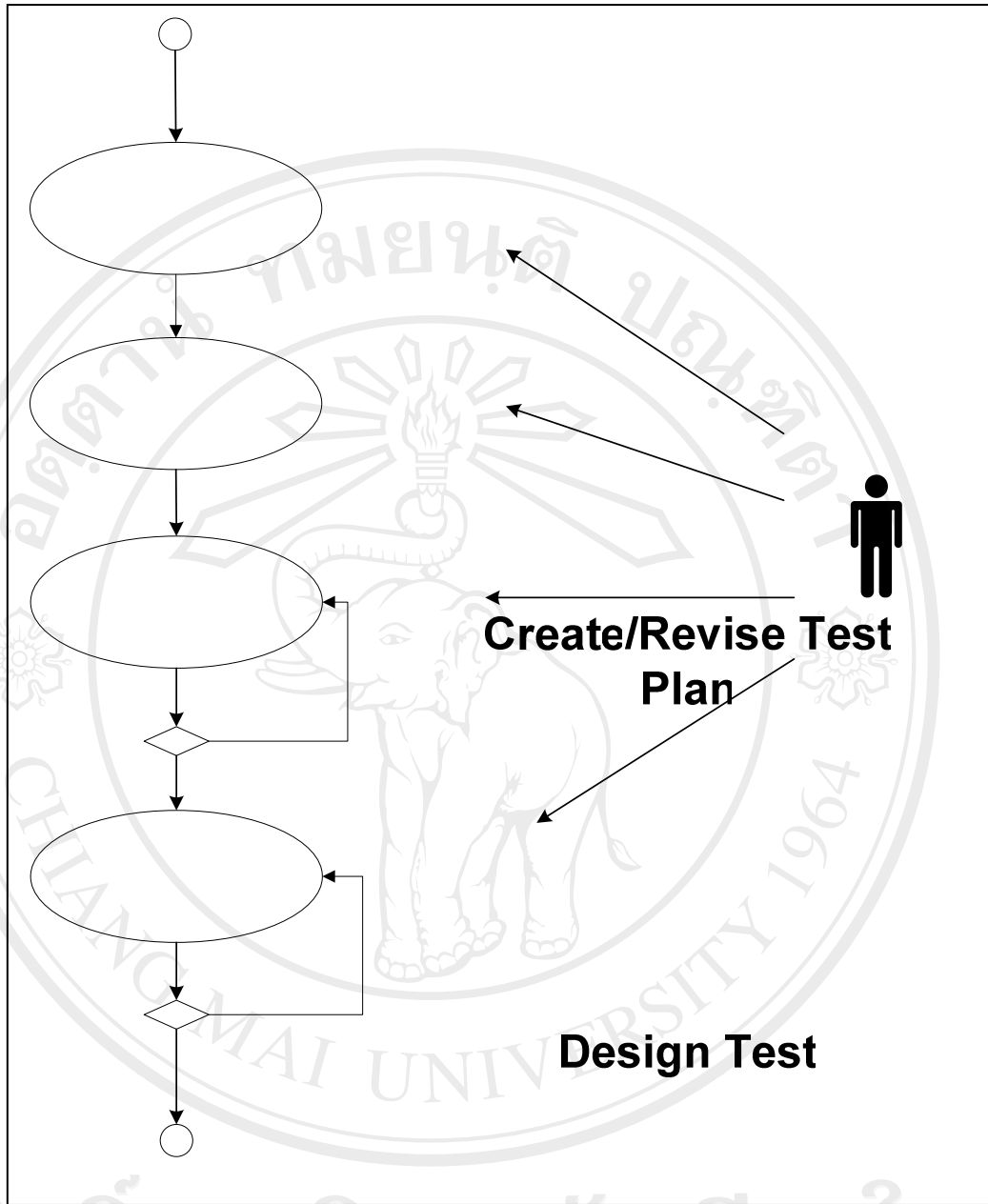
1. SA จะได้รับ Test Log จาก Tester ซึ่ง Test Log จะระบุว่า Subsystem ได้ผ่านการ Test Subsystem แล้ว
2. SA นำเอา Subsystem ที่ผ่านการ Test จาก SCM และ นำ Integration Build Plan มาทำการ Integrate เป็น System ถ้า SA สามารถทำการ Integrate Subsystem เป็น System

3. แต่ถ้า SA ไม่สามารถทำการ Integrate เป็น System ได้ SA จะส่ง Test log คืนให้ Programmer และ Programmer จะนำเอา Component จาก CM Repository มาทำการแก้ไข ย้อนกลับไปทำ **Process: Implement Components**
4. SA นำ Delivery Document Template มาสร้าง Delivery Document เพื่อแสดงว่า ตอนนี้ได้ทำการ Integrate System เรียบร้อยแล้ว
5. SA ส่ง Delivery Document ให้ PM Approved
6. SA ส่ง Delivery Document ให้ Tester เพื่อแจ้งให้ Tester ทราบว่า SA ได้ Integrate System เรียบร้อยแล้ว
7. ถ้า System ผ่านการ test เรียบร้อยแล้ว SA จะส่ง Delivery Document และ Unit Test ให้กับ SCM เพื่อทำการจัดเก็บและควบคุม
8. PM ขอ Setup Baseline กับ SCCB โดยจะประกอบไปด้วย Integration Build Plan

4.1.3 Test

ส่วนนี้จะอธิบายถึงการทำงานของ Tester ในการ Test โปรแกรมที่ได้รับจาก Programmer โดย Test ตาม Test plan ที่ได้ออกแบบไว้ เมื่อทำการ Test จนไม่พบ Bug หรือ Error แล้วจะทำการส่ง Test Log ให้กับ SA ทราบต่อไป ซึ่งขั้นตอนการทำงาน Test ดังรูป

4.11



รูป 4.11 Test Workflow

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

จากรูป 4.11 สามารถอธิบายรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.12-4.15

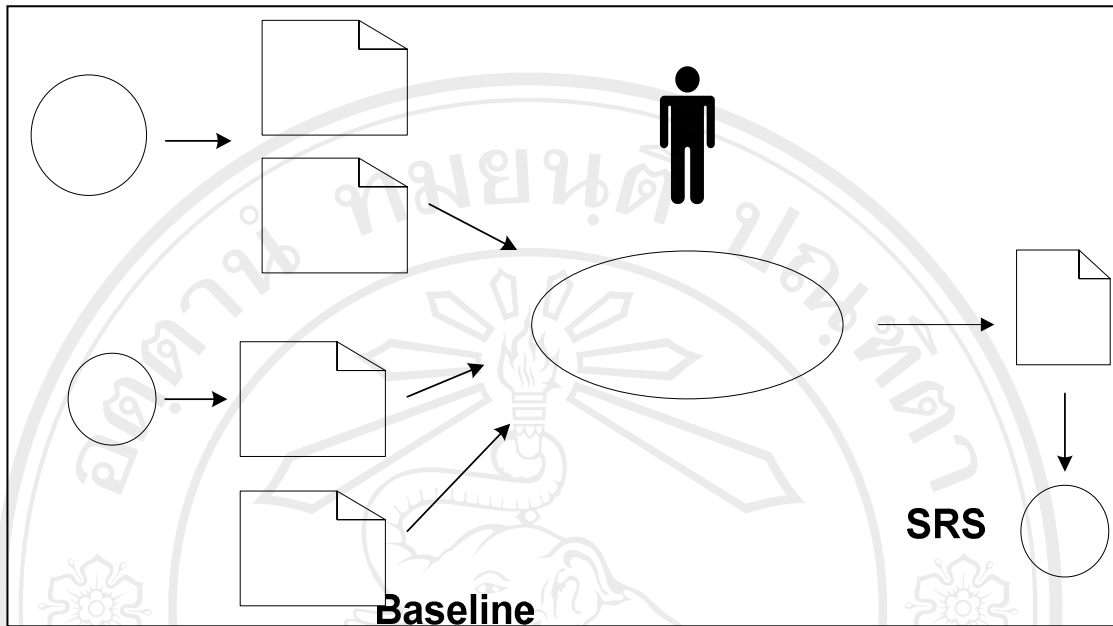
Copyright © by Chiang Mai University

All rights reserved

Execute Test in Subsystem

[Redo Test]

Create/Revise Test Plan



รูป 4.12 Create/Revise Test Plan Artifacts Flow

จากรูป 4.12 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการสร้าง Test Plan ซึ่งมีขั้นตอนการทำงานดังนี้

1. Test Leader ทำการสร้าง Test Plan จาก Test Plan Template โดยนำเอา SRS, Use Case Specification, Test Guideline มาประกอบการพิจารณาในการทำ Test Plan
2. Test Leader ส่ง Test Plan ให้ PM เพื่อเอาไปรวมเป็น SPP จากนั้นเข้าสู่ **Process:**

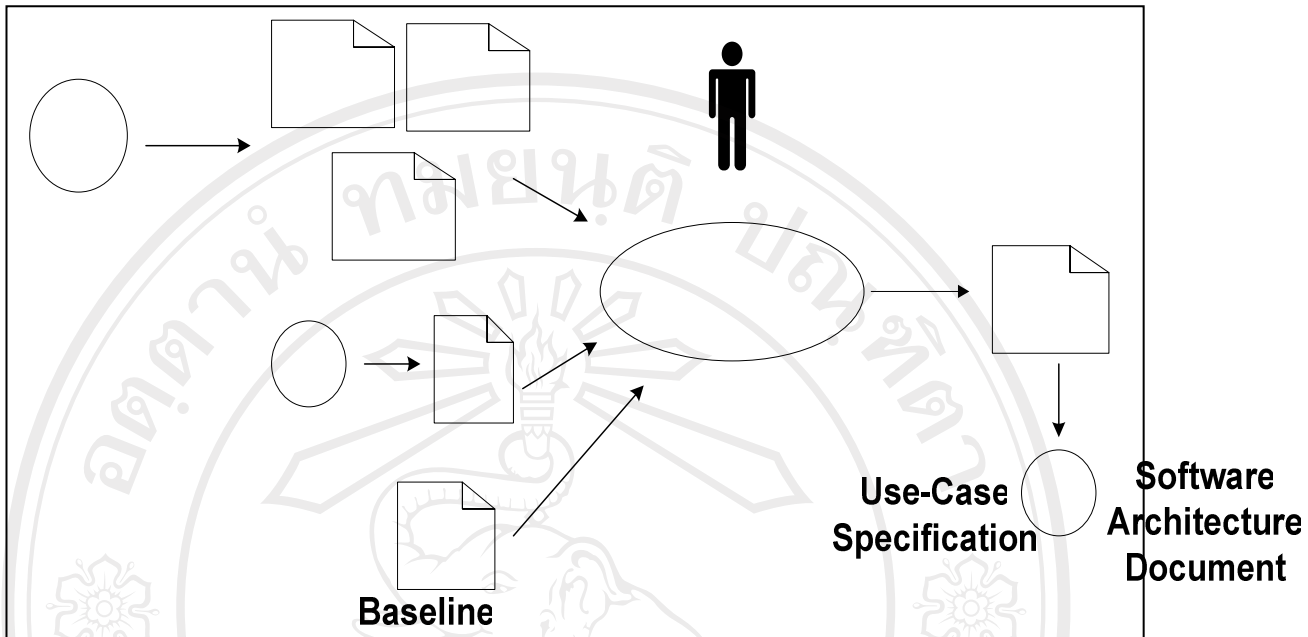
Create/Revise Software Project Plan

SCM

Test Guideline

Test Plan Template

Design Test



รูป 4.13 Design Test Artifacts Flow

จากรูป 4.13 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการสร้าง Test Procedure และ Test Case ลงในเอกสาร Test Log ซึ่งมีขั้นตอนการทำงานดังนี้

1. Tester ทำการสร้าง Test Procedure และ Test Case จาก Test Log Template โดยการนำเอา Use Case Specification, Test Plan, Analysis and Design Document และ Software Architecture Document เพื่อมาประกอบการพิจารณาในการทำ Test Log ในส่วนของ Test Procedure และ Test Case

2. Tester ส่ง Test Log ให้ Test Leader ทำการ Review

3. Tester ส่ง Test Log ให้ PM ทำการ Review

4. Tester ส่ง Test Log ให้ SQA ทำการ Review

5. Tester ส่ง Test Log ให้ SCM เพื่อทำการจัดเก็บและควบคุม

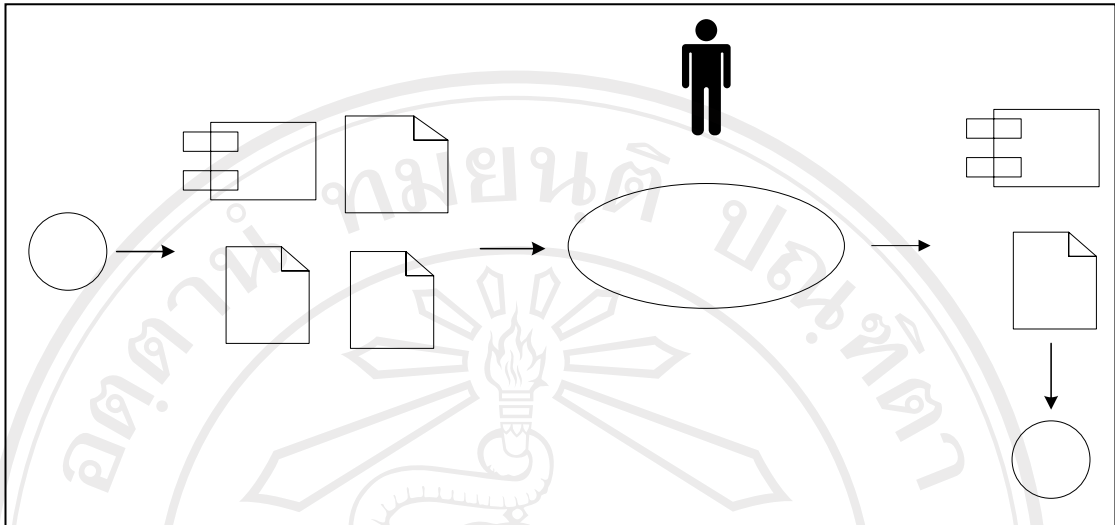
Analysis & Design Document

SCM

Test Plan

Test Log Template

Execute Test in Subsystem

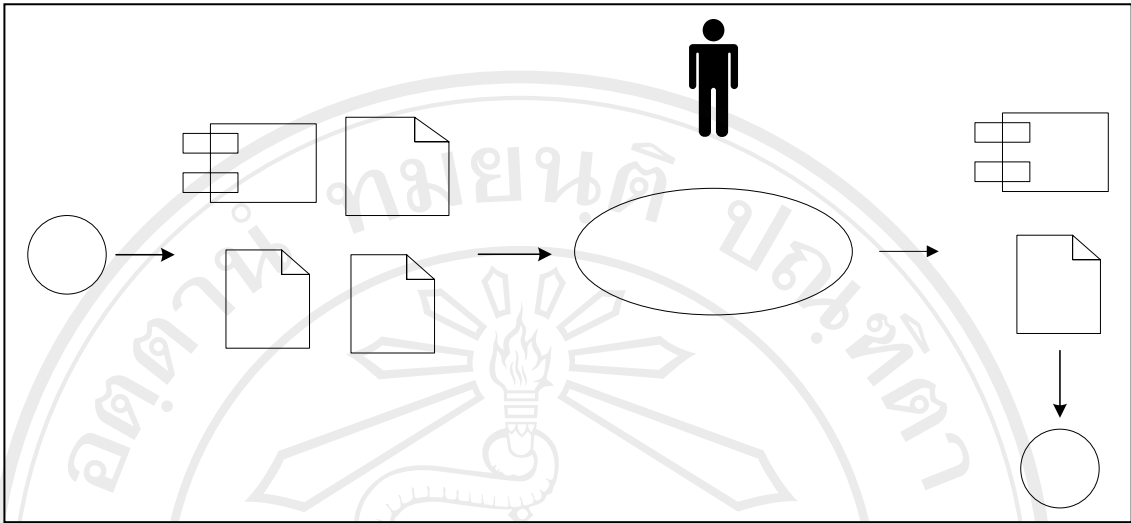


รูป 4.14 Execute Tests in Subsystem Artifacts Flow

จากรูป 4.14 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการ Test และเก็บผลลัพธ์ที่ได้จากการ Test ซึ่งมีขั้นตอนการทำงานดังนี้

1. Tester ได้รับ Delivery Document จาก **Process: Integrate Each Subsystem** Tester จะนำ Subsystem จาก CM Repository มาทำการ Test โดย Test ตาม Test Log และนำ Test Plan มาประกอบการทำ Test Subsystem
2. เมื่อ Tester ทำการ Test Subsystem โดย Tester จะบันทึกผลลัพธ์ที่ได้จากการ Test Subsystem ลงใน Test Log
3. ในกรณีที่ Test Subsystem ไม่ผ่าน Tester จะส่ง Test Log ให้กับ SA เพื่อแจ้งให้ทราบว่า Test Subsystem ไม่ผ่าน
4. Tester ส่ง Test Log ให้ Test Leader ทำการ Review
5. Tester ส่ง Test Log ให้ PM ทำการ Approve
6. Tester ส่ง Test Log ให้ SQA ทำการ Review

Execute Test in System



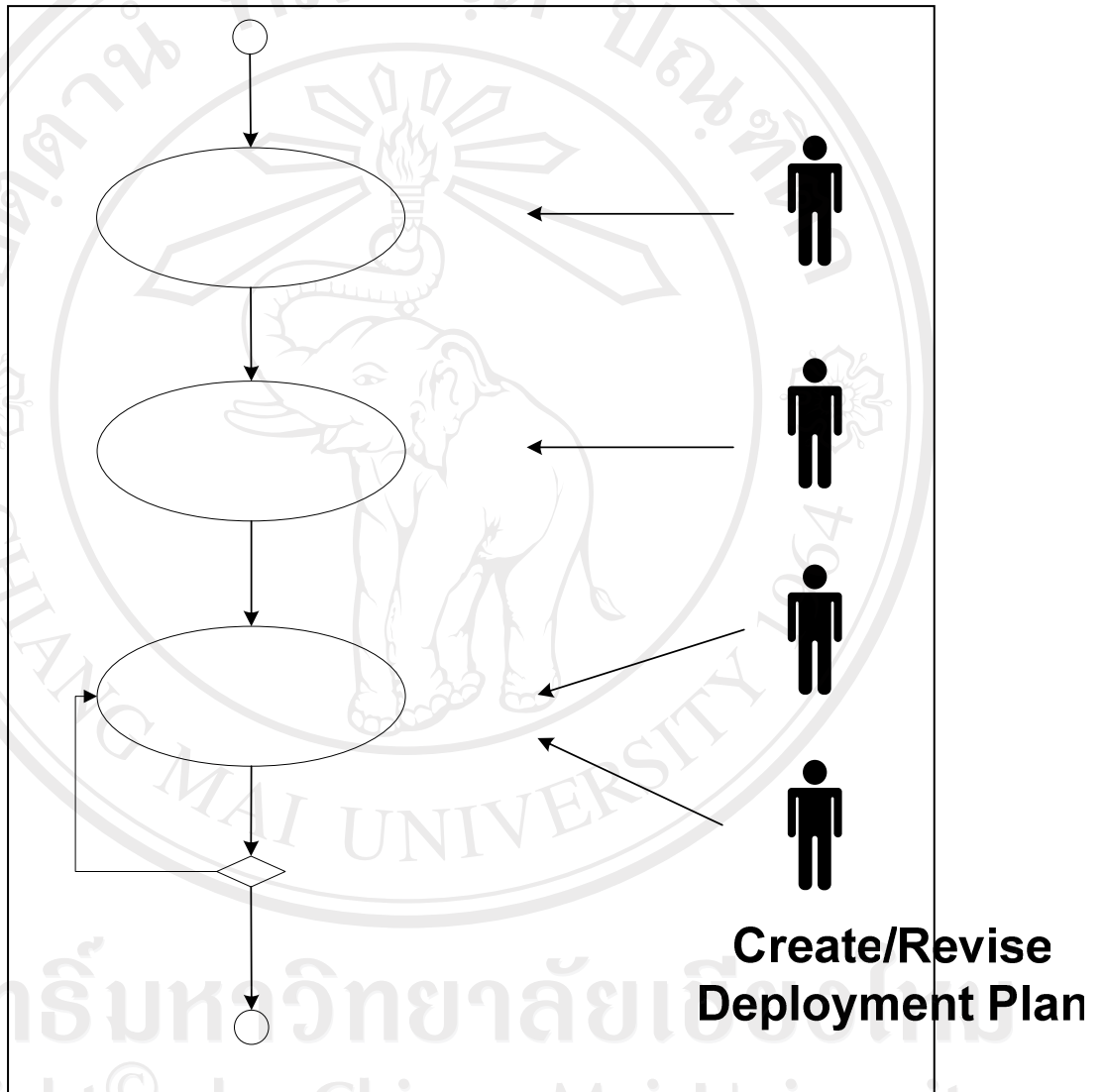
รูป 4.15 Execute Tests in System Artifacts Flow

จากรูป 4.15 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์ในการ Test System ซึ่งมีขั้นตอนการทำงานดังนี้

1. Tester ได้รับ Delivery Document จาก Process: Integrate the System Tester จะนำ System มาทำการ Test โดย Test ตาม Test Log และนำ Test Plan มาประกอบการทำ Test System
2. เมื่อ Tester ทำการ Test System โดย Tester จะบันทึกผลการ Test System ลงใน Test Log
3. ในกรณีที่ Test System ไม่ผ่าน Tester จะส่ง Test Log ให้กับ SA เพื่อแจ้งให้ทราบว่า Test System ไม่ผ่าน
4. Tester ส่ง Test Log ให้ Test Leader ทำการ Review
5. Tester ส่ง Test Log ให้ PM ทำการ Approve
6. Tester ส่ง Test Log ให้ SQA ทำการ Review
7. ถ้า Test System ผ่าน Tester จะส่ง Test Log ที่ผ่านการ Test เรียบร้อยแล้วให้กับ SA เพื่อให้ SA ส่ง Test Log ให้กับ SCM ทำการจัดเก็บและควบคุม

4.1.4 Deployment

ส่วนนี้จะอธิบายถึงการทำงานของ Support ในการวางแผนก่อนการส่งมอบระบบให้กับลูกค้า รวมถึงการสร้างเอกสาร, การ Training ต่างๆ โดย Support จะทำการทำการทดสอบระบบตามที่ลูกค้าได้ตกลงไว้ ซึ่งขั้นตอนการทำงาน Deployment ดังรูป 4.16

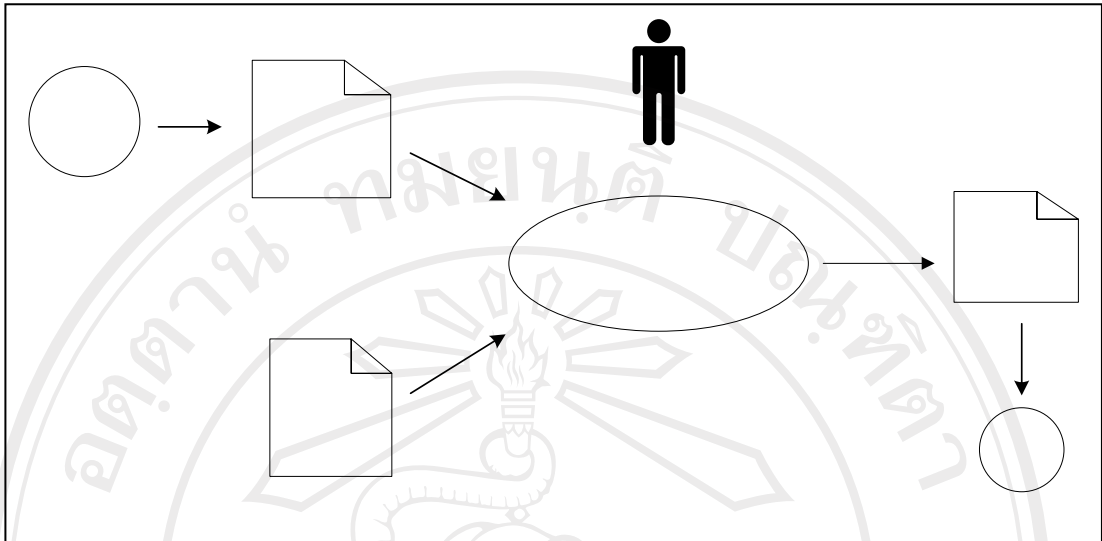


รูป 4.16 Deployment Workflow

จากรูป 4.16 สามารถอธิบายรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.17-4.19

**Develop Support
Material**

Create/Revise Deployment Plan



รูป 4.17 Create/Revise Deployment Plan Artifacts Flow

Baseline

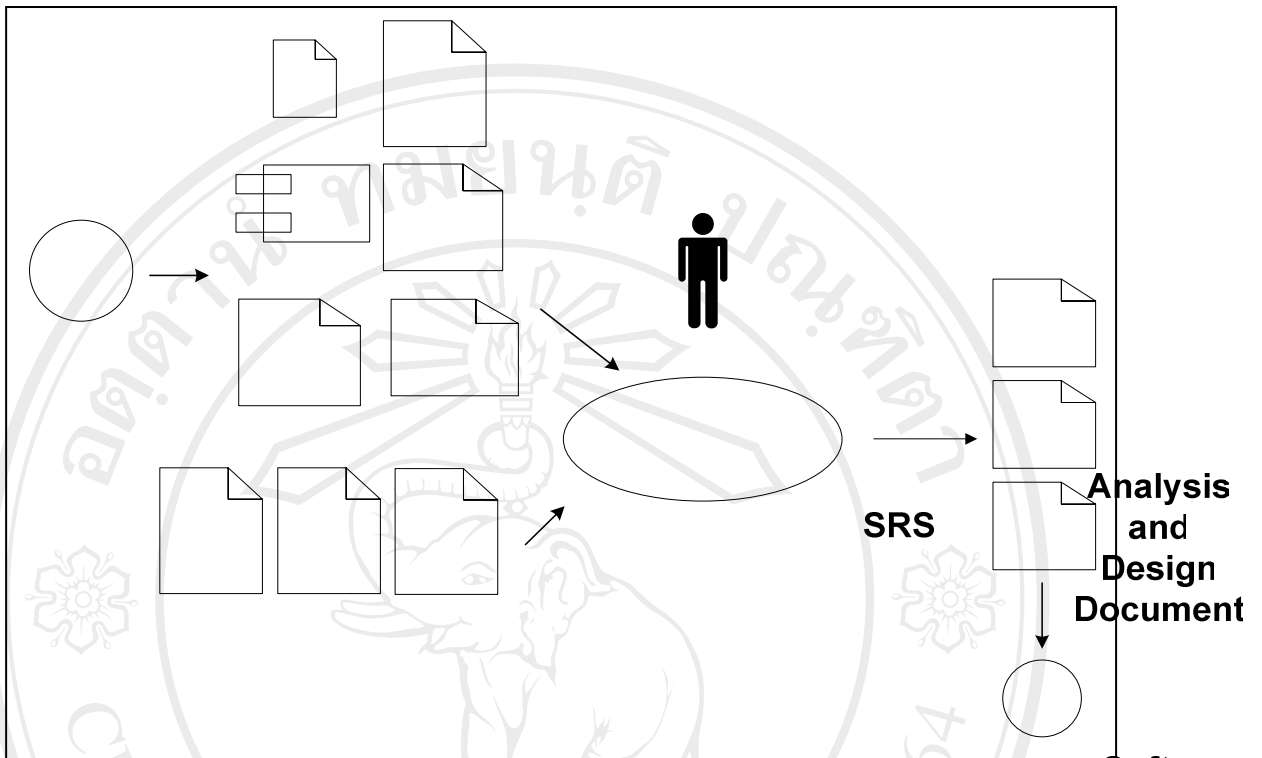
Software
Development
Plan

จากรูป 4.17 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีผู้เกี่ยวข้องส่งเพื่อทำการ Plan ในการทำ Deployment ให้กับลูกค้า ซึ่งมีขั้นตอนการทำงานดังนี้

1. Support ทำการสร้าง Deployment Plan จาก Deployment Plan Template โดยนำเอาเอกสาร Software Development Plan มาประกอบการสร้าง Deployment Plan
2. Support ส่ง Deployment Plan ให้ PM ทำการ Approve
3. Support ส่ง Deployment Plan ให้ SQA ทำการ Review
4. Support ส่ง Deployment Plan ให้ SCM ทำการจัดเก็บและควบคุม

Deployment
Plan
Template

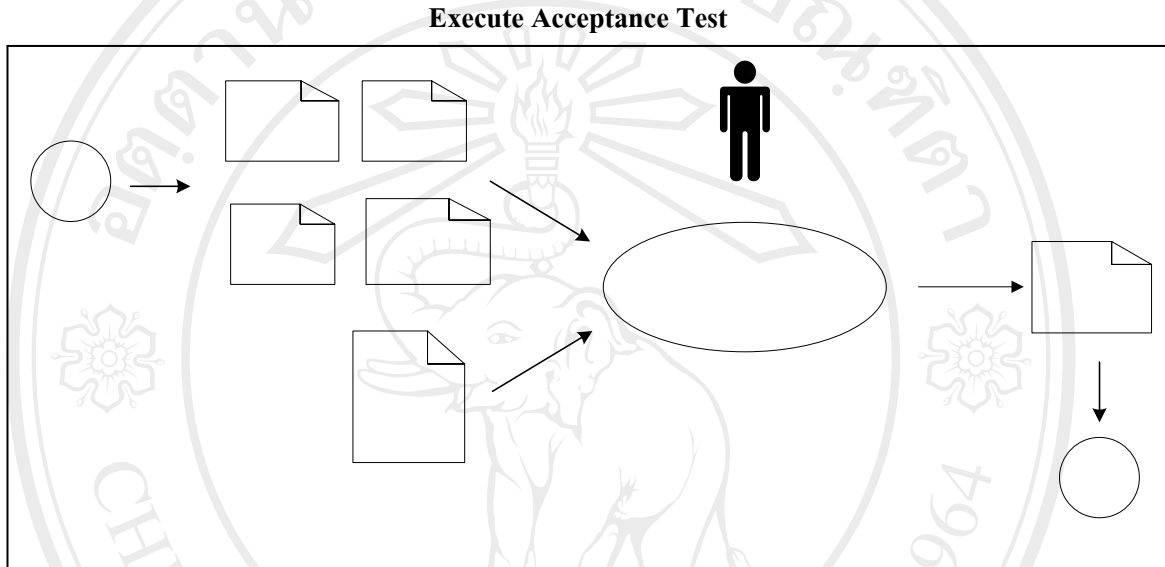
Develop Support Material



รูป 4.18 Develop Support Material Artifacts Flow

- จากรูป 4.18 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนั้นระบุวัตถุประสงค์เพื่อจัดทำ User Manual และ Training Outline สำหรับ Customer และจัดทำ Deliverable Item List สำหรับประกอบการทำ Deployment ให้กับ Customer ซึ่งมีขั้นตอนการทำงานดังนี้
1. Support ทำการสร้าง User Manual จาก User Manual Template โดยนำเอา SRS และ Analysis and Design Document มาทำการสร้าง User Manual เบื้องต้นและเมื่อได้ System ที่ผ่านการ Setup Baseline แล้ว Support จะทำการ Update User Manual ให้ถูกต้อง
 2. Support ทำการสร้าง Training Outline จาก Training Outline Template โดยใช้ System (Execute file), SRS และ User Manual มาประกอบการสร้าง Training Outline
 3. Support ทำการสร้าง Deliverable Item List จาก Deliverable Item List Template โดยนำเอา Software Development Plan, Integration Build Plan และ Deployment Plan มาประกอบการสร้าง Deliverable Item List
 4. Support ส่ง Deliverable Item List, User Manual และ Training Outline ให้กับ PM เพื่อทำการ Approve

5. Support ส่ง Deliverable Item List, User Manual และ Training Outline ให้กับ SQA เพื่อทำการ Review
6. Support ส่ง Deliverable Item List, User Manual และ Training Outline ให้ SCM เพื่อทำการจัดเก็บและควบคุม



รูป 4.19 Execute Acceptance Test Artifacts Flow

จากรูป 4.19 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการทดสอบ Software ตาม Acceptance Criteria ที่ได้ตกลงไว้กับ Customer ซึ่งมีขั้นตอนการทำงานดังนี้

1. Support สร้าง User Acceptance Form จาก User Acceptance Form Template โดยนำเอา Deployment Plan, Software Development Plan, Test Log และ Deliverable Item List มาประกอบการสร้าง User Acceptance Form
2. Support ส่ง User Acceptance Form ให้ PM Approve ให้ SQA Review และให้ SM Approve ก่อนที่จะทำ User Acceptance Test
3. Support ทำการเตรียม Environment และ Install software เพื่อใช้ในการทำ User Acceptance Test
4. การทำ User Acceptance Test อาจจะทำที่ Development Site หรือ Customer Site ก็ได้ ทั้งนี้ขึ้นอยู่กับความคิดเห็นของ PM

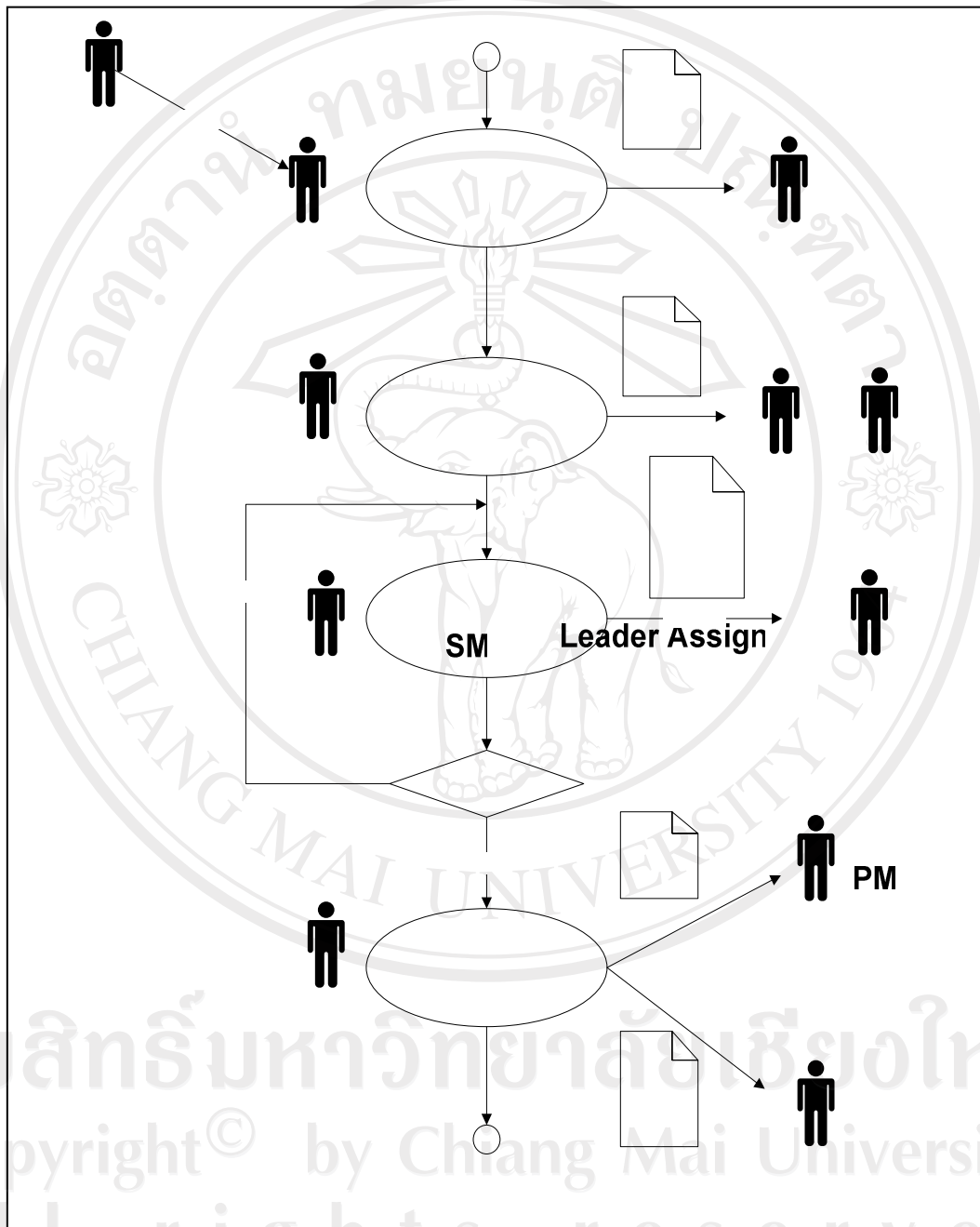
**User
Acceptance
Form
Template**

- กรณี ทำ User Acceptance Test ที่ Development Site PM จะแจ้งให้ Customer ทราบ เพื่อนัดวันที่ Customer ต้องเข้ามาทำ User Acceptance Test ที่ Development Site
 - กรณี ทำ User Acceptance Test ที่ Customer Site PM จะแจ้งให้ Customer ทราบ เพื่อนัดวันที่ Support จะเข้าไปทำ User Acceptance Test ที่ Customer Site
5. Support/Customer จะทำ User Acceptance Test โดยใช้ Test Log มาประกอบการทำ User Acceptance Test
6. Support ระบุผลจากการทำ User Acceptance Test ไว้ใน User Acceptance Form
- ถ้า Customer ยอมรับการ Test ครั้งนี้ จะลงชื่อไว้ใน User Acceptance Form และ Support ก็จะสามารถทำในขั้นตอนการ Deployment ต่อไป
 - ถ้า Customer ไม่ยอมรับ ให้ Support ระบุข้อผิดพลาด หรือถ้าพบ New/Change requirement ให้ Support กลับมาทำขั้นตอน Change Management Process โดยทำ Change Request Form และนำส่งให้ PM ทำการวิเคราะห์ และแก้ไขต่อไป
7. การทำ Deployment นั้น Support จะระบุผลไว้ใน User Acceptance Form
- ถ้าไม่พบปัญหาหรือข้อผิดพลาดในการทำ Deployment แล้ว Support ก็จะทำการส่งมอบ Deliverable Item ตามรายการที่ระบุไว้ Support จะส่ง User Acceptance Form ให้ Customer ลงชื่อ เพื่อบอกว่า User ยอมรับการทำ Deployment และได้รับ Deliverable Item แล้ว เมื่อทุกอย่างเรียบร้อยแล้ว Support จะส่ง User Acceptance Form นี้ให้ PM Review และ จัดเก็บที่ SCM
 - ถ้าพบปัญหาหรือข้อผิดพลาด Support จะต้องระบุปัญหาหรือข้อผิดพลาด จากการทำ Deployment ไว้ใน User Acceptance Form แล้วให้ Support กลับมาทำขั้นตอน Change Management Process โดยทำ Change Request Form และนำส่งให้ PM ทำการวิเคราะห์และแก้ไขต่อไป

4.2 Software Project Planning

Software Project Planning หรือการวางแผนโครงการซอฟต์แวร์ มีจุดประสงค์เพื่อสร้างแบบแผนและทิศทางการดำเนินงานของ Project ทั้งหมด ซึ่งจะต้องได้รับการออกแบบอย่างรอบคอบ ดังนั้นจึงได้มีการออกแบบกระบวนการทำงานในส่วน Software Project Planning ขึ้นมาใหม่เพื่อปรับปรุงกระบวนการทำงานแบบเดิม โดยอิงมาตรฐาน CMM Level 2

โดยกระบวนการทำงานแบบเดิม สามารถนำมาเขียนขั้นตอนการทำงานในส่วนของ Software Project Planning ได้ดังรูป 4.20

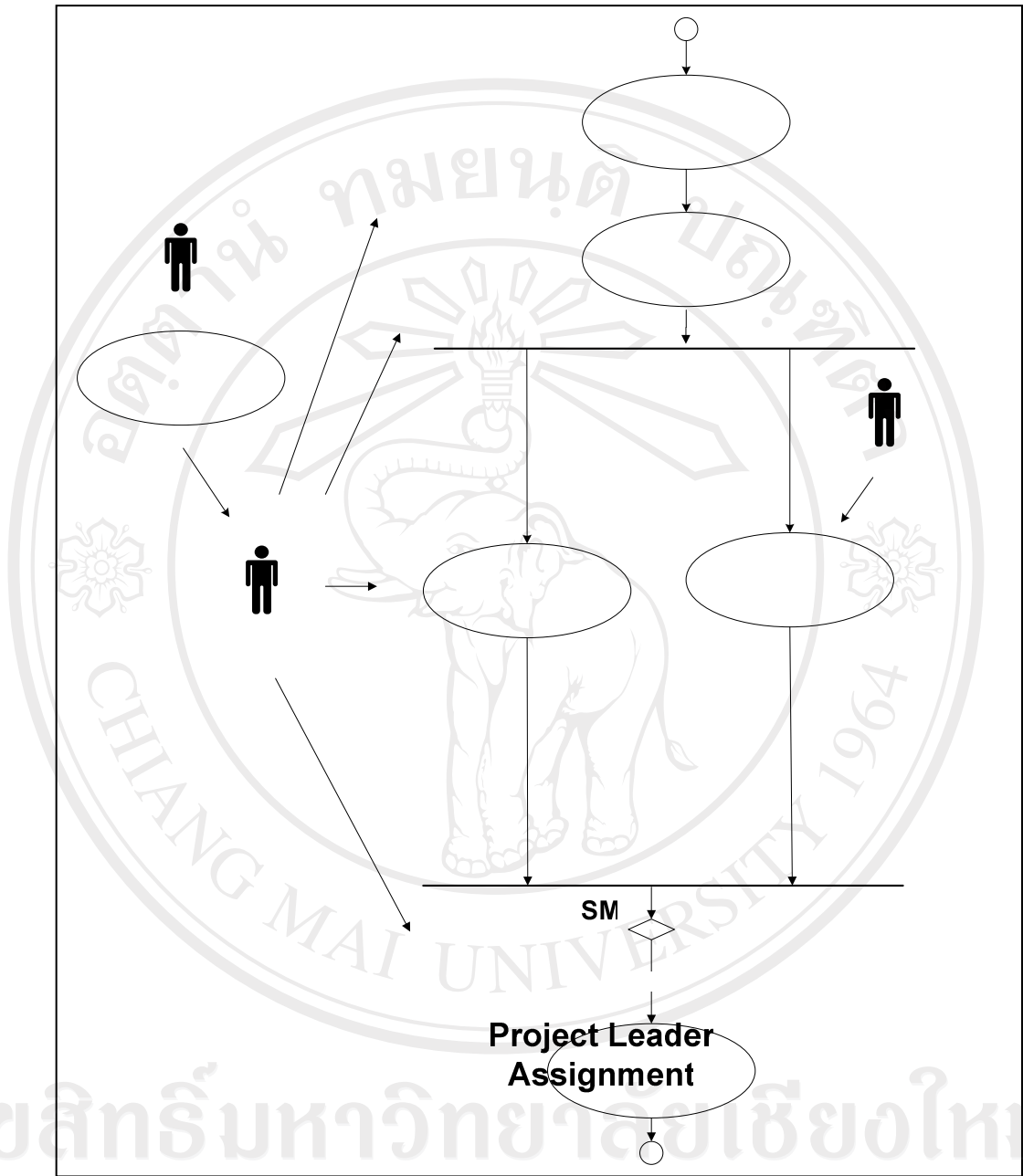


รูป 4.20 Software Project Planning Workflow (รูปแบบเดิม)

และส่วนของกระบวนการทำงานแบบใหม่ที่ได้ทำการปรับปรุงแล้ว สามารถเขียนขั้นตอนการทำงานในส่วนของ Software Project Planning ได้ดังรูป 4.21

No

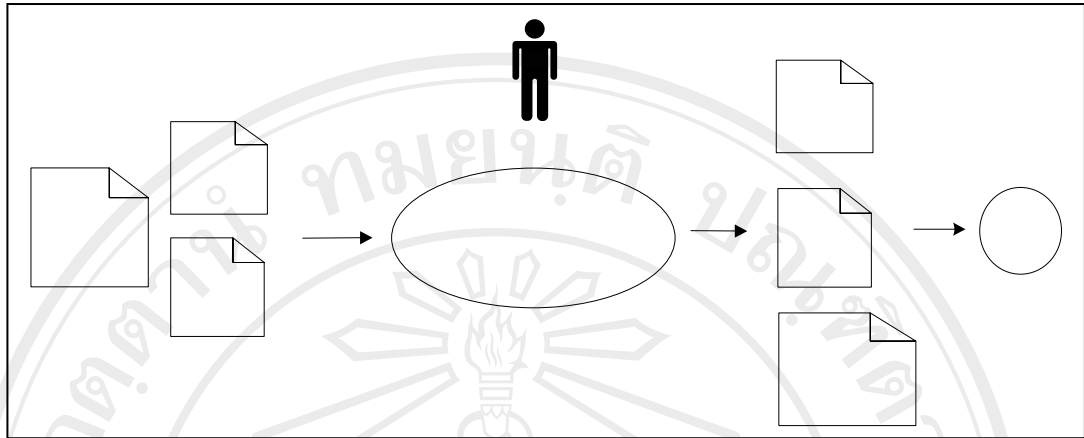
Tracking



รูป 4.21 Software Project Planning Workflow (ที่ได้รับการปรับปรุง)

จากรูป 4.21 ส่วนของกระบวนการทำงานที่ได้รับการปรับปรุงแล้ว สามารถอธิบายถึงรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.22-4.24

Create Draft Schedule

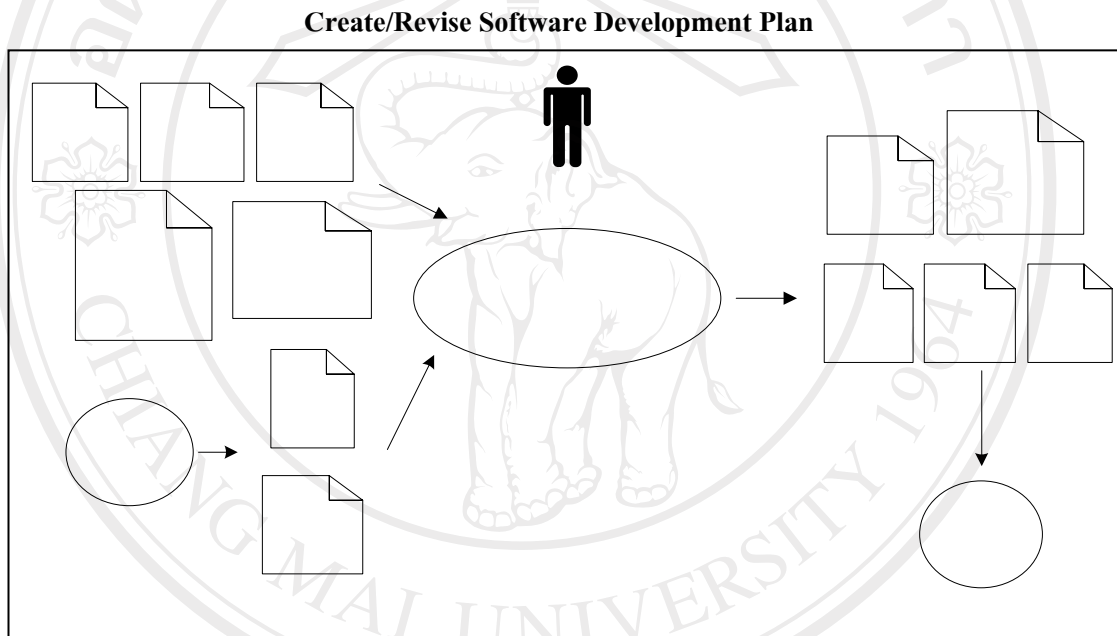


รูป 4.22 Create Draft Schedule Artifacts Flow

จากรูป 4.22 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการสร้าง schedule ของ Project เบื้องต้น สำหรับใช้ในส่วนของ Requirements Management ซึ่งมีขั้นตอนการทำงานดังนี้

1. PM ทำการ check in Project Management Assignment และ SCM จัดเก็บ
2. PM ทำการแจ้ง SQA และ SCM ให้รับทราบได้ถูก assign ใน Project
3. PM ทำการวิเคราะห์ โดยใช้ประสบการณ์ เพื่อทำประมาณเวลา(Effort) ในการทำกิจกรรมในช่วงของ Requirement Management โดยจะประมาณ Effort ที่ได้จะระบุเป็น Man Hour สำหรับช่วง Requirement Management
4. PM นำ Effort ที่ประมาณได้ มาสร้าง Draft Schedule จาก Draft Schedule Template เพื่อเริ่มต้นการทำ Project ในช่วง Requirements Management
5. PM ทำการ Assign Team Member โดยสร้าง Project Team Member Assignment จาก Project Team Member Assignment Template
6. PM ส่ง Draft Schedule และ Project Team Member Assignment ให้ SQA ทำการ Review
7. PM ส่ง Draft Schedule และ Project Team Member Assignment ให้ SM ทำการ Approve
8. PM สร้าง Project Folder และ Sub Folders ที่เกี่ยวข้อง และจัดสรรพื้นที่ใน Repository และ สิทธิการเข้าถึง Folder และ Sub Folders ใช้เอกสารให้กับ Team Member
9. PM ส่ง SCM User Request ให้กับ SCM

10. เมื่อ PM ทำ Draft Schedule เสร็จเรียบร้อยแล้ว PM จะทำการจัด First Meeting เพื่อแจ้งให้ BSA, SA, Programmer, Tester, Support, SQA และ SCM ทราบว่าใครมีหน้าที่รับผิดชอบอะไรบ้าง และขอบเขตของ Project เป็นอย่างไร
11. PM ส่ง Draft Schedule ให้ลูกค้าเพื่อทำการ Approve
12. PM ส่ง Draft Schedule ให้ BSA เพื่อใช้ในการทำ Use Case Specification ต่อไป
13. PM ส่ง Draft Schedule และ Project Team Member Assignment ให้กับ SCM เพื่อทำการจัดเก็บ และควบคุม



รูป 4.23 Create/Revise Software Development Plan Artifacts Flow

จากรูป 4.23 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการสร้าง Software Development Plan ของ Project ซึ่งมีขั้นตอนการทำงานดังนี้

1. ในการพัฒนาซอฟต์แวร์ของบริษัท เป็นไปตาม Software Development Life Cycle ซึ่งเก็บไว้ใน MSSCLC
2. เมื่อได้ Use Case Specification จัด SCM Project Team Member เพื่อ review เอกสาร Use Case Specification และ ประชุมตกลงกับทีมงาน SA, SQA, SCM, Programmer, Support, Tester Leader และ Tester (ควรจะได้ MOM หรือข้อตกลง)
3. PM คำนวณ Estimation โดยใช้ Sub Process: Estimation เพื่อคำนวณค่าการประมาณ

Software Development Plan Schedule Template

Software Development Plan Template

ลิขสิทธิ์ในเอกสารนี้เป็นของสงวนโดย
Copyright © by Chiang Mai University
All rights reserved

Pr
Cr
D

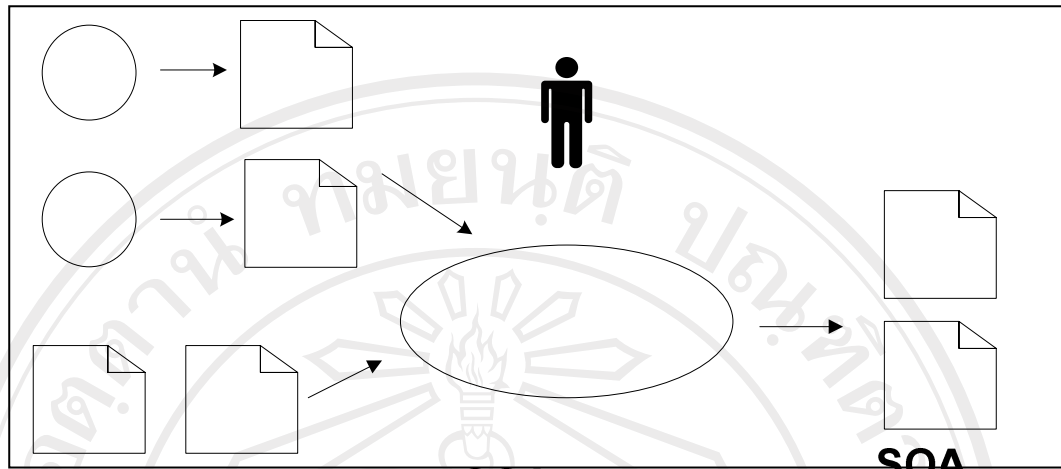
4. PM นำ Estimation มาลงใน Schedule เพื่อมาสร้าง SDP Schedule จาก SDP Schedule รวมถึง Critical computer resource ด้วย
5. PM ระบุ Tracking Schedule ใน SDP ด้วย ทั้ง Periodic และ Milestone
6. PM ประเมินการอุปกรณ์ ที่จำเป็นและ facilities ที่จะใช้ใน Project และ PM กำหนด Schedule ที่ต้องใช้อุปกรณ์ที่จำเป็นใน Project Plan
7. PM ทำการกำหนด configuration Item ส่วนที่นอกเหนือจาก Default Configuration Item โดยระบุความรับผิดชอบแต่ละ CIs แล้วส่งให้ SCM เพื่อนำไปทำการสร้าง SCM Plan ต่อไป
8. PM จัดการประชุม Project Team Member เพื่อทำการ review ว่าการ Estimate สามารถเป็นไปได้หรือไม่
9. PM ทำการกำหนด Measurement ซึ่งประกอบด้วย Effort & Cost, Size, Schedule, Use of Critical computer resource
10. PM ส่ง Schedule ให้ SQA, SCM เพื่อทำ SQA Plan, SCM Plan
11. PM ส่งข้อมูลข้อ 2 – 7 และ SRS ให้ Tester Leader ทำ Test Plan เงื่อนไขการทดสอบ และการตรวจรับ (อ้างอิง **Process:Create/Revise Test Plan**)
12. PM เอาข้อมูลจากข้อ 2 – 8 ได้แก่ Estimate, Schedule, Facilities, Software work product และเอกสาร SRS มาสร้าง Software Development Plan
13. PM นำเอกสาร Software Development Plan, SQA Plan, SCM Plan, Test Plan มารวมกัน เพื่อที่จะสร้างเป็น Software Project Plan
14. Project Team Member ต้องเข้ามา review เพื่อตกลงร่วมกันใน Plans ที่เกิดขึ้นได้แก่ Software Development Plan, SQA Plan, SCM Plan, Test Plan
15. เมื่อ Project Team Member ทำการตกลงกันเรียบร้อยแล้ว
 - 15.1 ถ้ากรณีที่เอกสารทั้งหมดมีการแก้ไข PM จะส่งเอกสารที่ต้องการให้มีการแก้ไข ให้กับบุคคลที่เกี่ยวข้อง เมื่อทำการแก้ไขเสร็จจะให้ส่งให้ PM และ SQA ทำการ Review ตามลำดับ
 - 15.2 ถ้ากรณีที่เอกสารทั้งหมดไม่มีการแก้ไข PM และ SQA ทำการ Review ตามลำดับ
16. PM ส่ง Software Development Plan, SQA Plan, SCM Plan และ Test Plan ให้ SM approve
17. เมื่อ SM ทำการ Approve เอกสารที่เกิดขึ้นทั้งหมด เอกสารจะถูกเก็บ และควบคุมโดย SCM
18. PM ขอ Setup Baseline กับ SCCB โดยจะประกอบไปด้วยเอกสาร

- Software Development Plan
- Software Development Plan Schedule
- SQA Plan
- SCM Plan
- Test Plan

หมายเหตุ : กรณีที่เกิดการเปลี่ยนแปลง ให้เข้า Change Management Process เพื่อแก้ไขเอกสารที่ถูกกระทบ ตามกระบวนการต่อไป

1. กรณีมีการ Revise Plan เนื่องจากมี Change เกิดขึ้น ให้ทำตามขั้นตอนดังนี้
 - 1.1 PM ทำ Estimation Schedule, Effort Size และ Risk Use of Critical Computer Resource ใหม่ โดยดูจากข้อมูลที่เก็บไว้ที่ SCM
 - 1.2 PM ปรับเปลี่ยน SDP Schedule และ Project Plan ใหม่ จาก Version ก่อนหน้านี้
 - 1.3 PM ส่งเอกสารให้กับผู้ที่เกี่ยวข้อง เพื่อดำเนินการแก้ไข หรือปรับเปลี่ยน โดยทำตามขั้นตอนของ Change Management Process
2. กรณีที่ไม่มีอุปกรณ์ที่จำเป็นใน Project PM ต้องระบุว่า จะจัดหาอย่างไร
 - 2.1 ไปใช้อุปกรณ์ของลูกค้า โดย PM จะต้องทำตารางเวลาในการไปใช้งาน เพื่อแจ้งให้ลูกค้ารับทราบและจัดการให้
 - 2.2 ยืมอุปกรณ์จากแผนกอื่น โดย PM จะต้องทำตารางเวลาในการไปใช้งาน เพื่อแจ้งให้แผนกนั้นรับทราบและจัดการให้
 - 2.3 จัดซื้อ PM จะต้องระบุ Spec ของอุปกรณ์และกำหนดวันที่ต้องได้รับ เพื่อให้ฝ่ายจัดซื้อดำเนินการได้ทัน

Close-Out Project



รูป 4.24 Close-Out Project Artifacts Flow

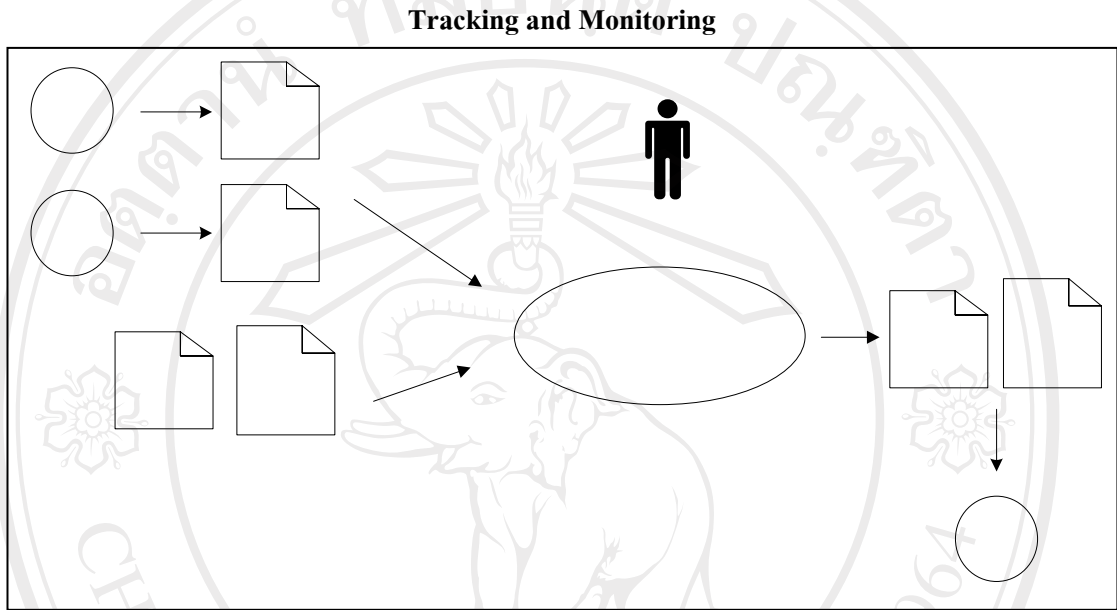
จากรูป 4.24 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการปิด Project ซึ่งมีขั้นตอนการทำงานดังนี้

1. PM จัดการประชุมกับ Project Team เพื่อแจ้งสรุป Lesson-Learned ที่เกิดขึ้นใน Project เพื่อใช้ในการทำ Closure Report
2. PM นำ Closure Report Template, SQA Closure Report และ SCM Closure Report มาทำ Closure Report โดยสรุป Effort, Cost, Schedule, Critical Computer Resource, Issue, Change, Requirement, Defect ที่เกิดขึ้นและ Activity ของ Project Team Member
3. PM ส่ง Closure Report ให้ SQA เพื่อทำการ Review
4. PM ส่ง Closure Report ให้ SM เพื่อทำการ Approve
5. PM ส่ง Closure Report ให้ SCM เพื่อทำการจัดเก็บและควบคุมและประกาศ Close Project ให้กับ Team ทราบ

4.3 Software Project Tracking and Oversight

Software Project Tracking and Oversight หรือการติดตามดูแลและตรวจสอบงานที่วางไว้ มีจุดประสงค์เพื่อตรวจสอบและเฝ้าติดตามขั้นตอนการทำงาน โดยมีรูปแบบการตรวจสอบที่ชัดเจน และไม่เข้าไปขัดขวางขั้นตอนการทำงานเดิม ดังนั้นจึงได้มีการออกแบบกระบวนการทำงานในส่วนของ Software Project Tracking and Oversight ขึ้นมาใหม่เพื่อปรับปรุงกระบวนการทำงานแบบเดิม โดยอิงมาตรฐาน CMM Level 2

จากรูป 4.21 ส่วนของกระบวนการทำงานของ Software Project Tracking and Oversight ที่ได้รับการปรับปรุงแล้ว สามารถอธิบายถึงรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.25



รูป 4.25 Tracking and Monitoring Artifacts Flow

จากรูป 4.25 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการติดตามความคืบหน้าของ Project ซึ่งมีขั้นตอนการทำงานดังนี้

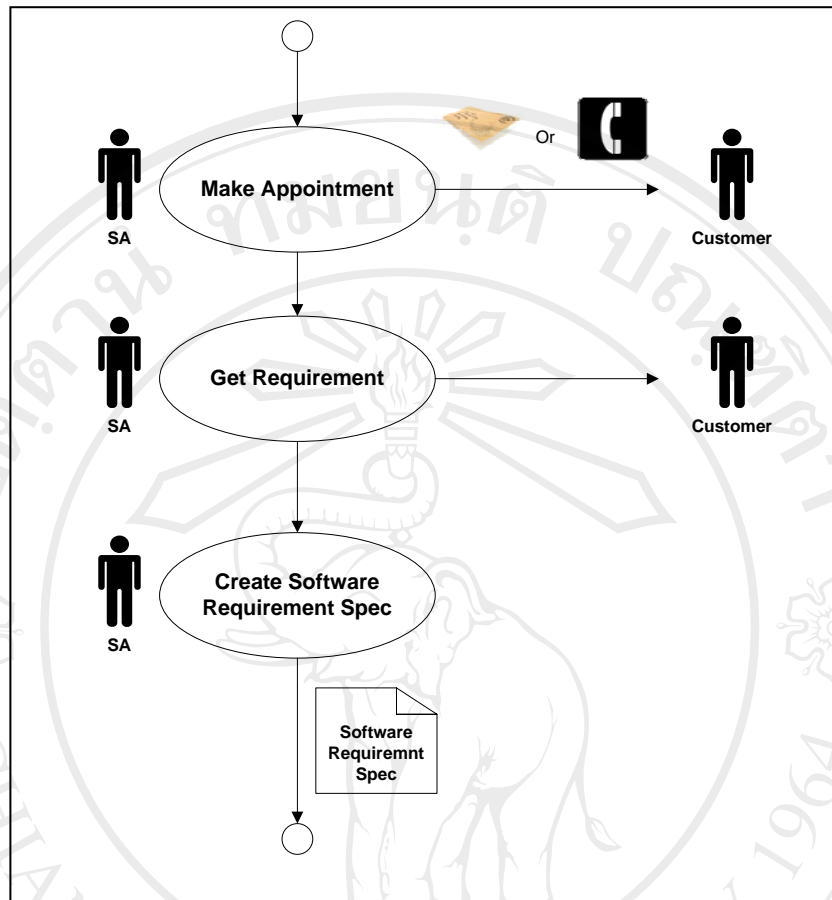
1. SM จะติดตามความก้าวหน้าโดยการจัด Monthly Meeting กับ Project Manager ของ ทุก Project
2. PM จะติดตามความก้าวหน้าของ Project โดยมีการจัด Periodic Reporting (ตามที่ตกลงไว้ในแต่ละ Project), Milestone Meeting หรือ Event-Driven ตามที่ระบุไว้ใน Project Plan
3. ในการประชุม PM จะต้องติดตามจำนวน Requirement, Software Schedule, Effort ของกิจกรรมต่าง ๆ, Critical Computer Resource (Use Materials), Issue, Change, Defect ที่เกิดขึ้นเปรียบเทียบกับที่วางแผนไว้ใน SPP และบันทึกลงใน Project Status Report
 - 3.1 หากการตรวจติดตามพบว่าคลาดเคลื่อนจาก Schedule หรือมีปัญหาเกิดขึ้น PM จะต้องหาวิธีแก้ไขหรือปรึกษากับผู้เกี่ยวข้อง
 - 3.2 ถ้าปัญหาที่เกิดขึ้นส่งผลกระทบต่อ Project ให้เข้า Change Management Process ก่อน หากปัญหาที่เกิดขึ้นรุนแรง PM ต้องรายงาน SM โดยด่วน

- 3.3 PM จะต้องติดตามการเปลี่ยนแปลงและปัญหาที่เกิดขึ้น จนกว่าจะถูกแก้ไขจนหมด
4. PM สร้าง Project Status Report จาก Project Status Report Template เพื่อรายงานความคืบหน้าของ Project ให้กับลูกค้า โดย PM จะกำหนดช่วงเวลาที่จะทำการส่ง Project Status Report ให้กับลูกค้าใน Software Development Plan
5. PM ส่ง SPTO Report ให้ SQA ทำการ Review
6. PM ส่ง SPTO Report ให้ SM ทำการ Approve
7. PM ส่ง Project Status Report และ SPTO Report ให้กับ SCM เพื่อทำการจัดเก็บ และควบคุม
8. ในขณะที่ PM ทำการ Tracking and Monitoring อยู่ นั้นอาจมีการเกิดการเปลี่ยนแปลงที่มีผลกระทบต่อ Plan เกิดขึ้น ให้ PM นำเข้าสู่ Change Management Process
9. ถ้าการทำ Process: Tracking and Monitoring นั้นยังไม่จบภายใน Iteration ก็จะต้องทำกระบวนการนี้ต่อไปจนจบ ถึงจะผ่านไปทำ **Process: Close-out Phase** ได้

4.4 Requirement Management

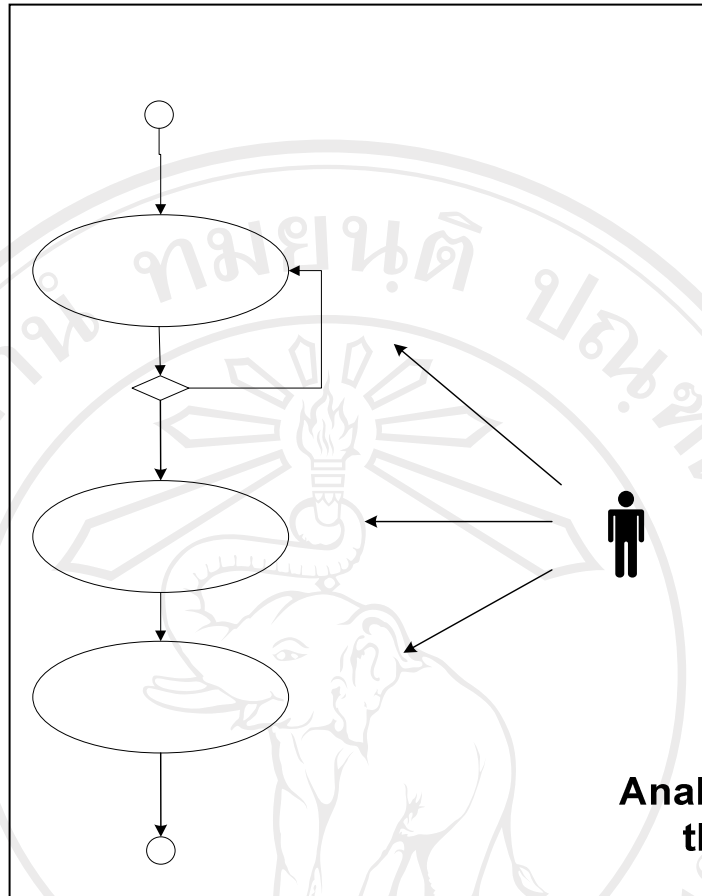
Requirement Management หรือการบริหารความต้องการของลูกค้า มีจุดประสงค์เพื่อค้นหาและจัดการความต้องการของลูกค้าและการยอมรับความต้องการของลูกค้า ดังนั้นจึงได้มีการออกแบบกระบวนการทำงานในส่วนของ Requirement Management ขึ้นมาใหม่เพื่อปรับปรุงกระบวนการทำงานแบบเดิม โดยอิงมาตรฐาน CMM Level 2

โดยกระบวนการทำงานแบบเดิม สามารถนำมาเขียนขั้นตอนการทำงานในส่วนของ Requirement Management ได้ดังรูป 4.26



รูป 4.26 Requirement Management Workflow (รูปแบบเดิม)

และส่วนของกระบวนการทำงานแบบใหม่ที่ได้ทำการปรับปรุงแล้ว สามารถเขียนขั้นตอนการทำงานในส่วนของ Requirement Management ได้ดังรูป 4.27



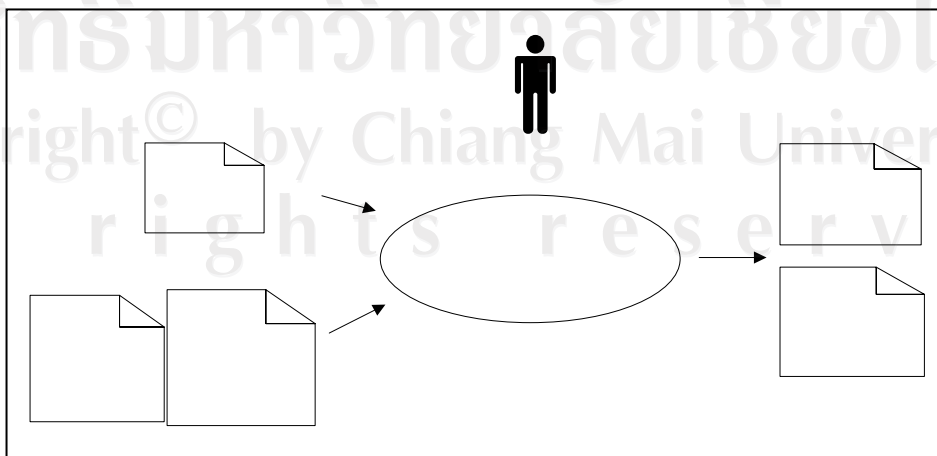
รูป 4.27 Requirement Management Workflow (ที่ได้รับการปรับปรุง)

จากรูป 4.27 ส่วนของกระบวนการทำงานของ Requirement Management ที่ได้รับการปรับปรุงแล้ว สามารถอธิบายถึงรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.28-4.29

Not Approve

Approve

Analyze/Reanalyze the Problem

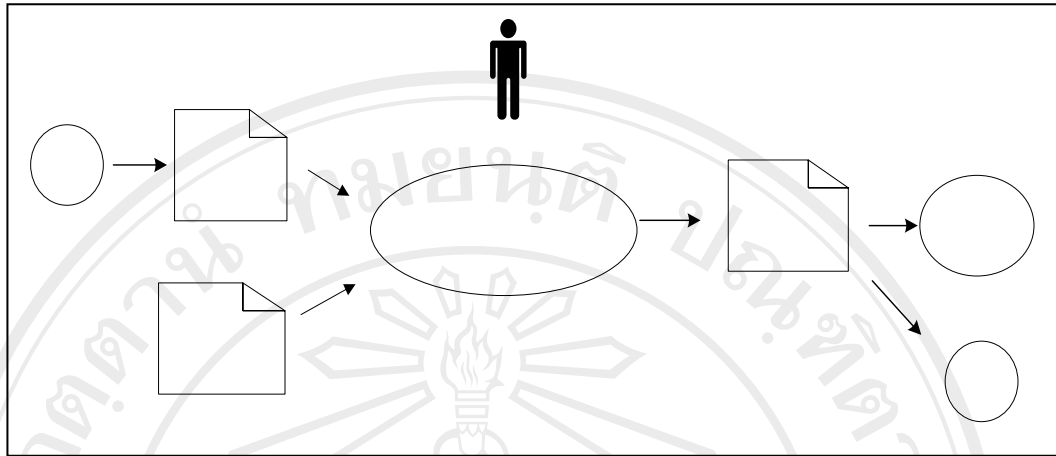


รูป 4.28 Analyze/Reanalyze the Problem Artifacts Flow

จากรูป 4.28 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการวิเคราะห์และเก็บรวบรวมความต้องการของลูกค้าและทำการกำหนดขอบเขตของระบบซึ่งมีขั้นตอนการทำงานดังนี้

1. BSA นำเอา Draft Schedule จาก SCM มาดูระยะเวลาที่ PM กำหนดให้ทำเอกสารของ Requirement Management
2. BSA นำเอา Use-Case Specification Template จาก Library มาทำการเก็บรายละเอียดและวิเคราะห์ความต้องการของลูกค้าโดยการทำการระบุ Actor ที่เข้ามาเกี่ยวข้องกับระบบลงใน Use-Case Specification
3. BSA ส่ง Use-Case Specification ให้กับ PM เพื่อทำการ Review เอกสารว่าเอกสารเสร็จตาม Draft Schedule ที่กำหนดไว้หรือไม่ ถ้าไม่เป็นไปตาม Draft Schedule ที่กำหนดไว้ จะเข้า Change Management Process
4. BSA ส่ง Use-Case Specification ให้ PM ทำการ Review
5. BSA ส่ง Use-Case Specification ให้ SQA ทำการ Review
6. BSA ส่ง Use-Case Specification ให้ SM ทำการ Approve
7. BSA ไปพบ Customer เพื่อ approve ว่า ครอบคลุมความต้องการ หรือไม่ โดยพิจารณาจากเอกสาร Use-Case Specification
8. ถ้าไม่ Approve BSA ก็จะย้อนกลับไปทำการแก้ไข Use-Case Specification
9. เมื่อ Customer อนุมัติเอกสารเรียบร้อยแล้ว BSA ก็จะ ส่ง Use-Case Specification ให้ SCM ทำการจัดเก็บและควบคุม
10. BSA นำ Software Architecture Document Template จาก Librarian มาใช้สร้าง Software Architecture Document เพื่อทำการอธิบายภาพรวมสถาปัตยกรรมของ Use-Case ในเอกสาร Software Architecture Document ในส่วนของ หัวข้อ Use Case View เท่านั้น
11. BSA ส่ง Software Architecture Document ให้ PM ทำการ Review
12. BSA ส่ง Software Architecture Document ให้กับ SCM ทำการจัดเก็บและควบคุม

Refine the System Definition



รูป 4.29 Refine the System Definition Artifacts Flow

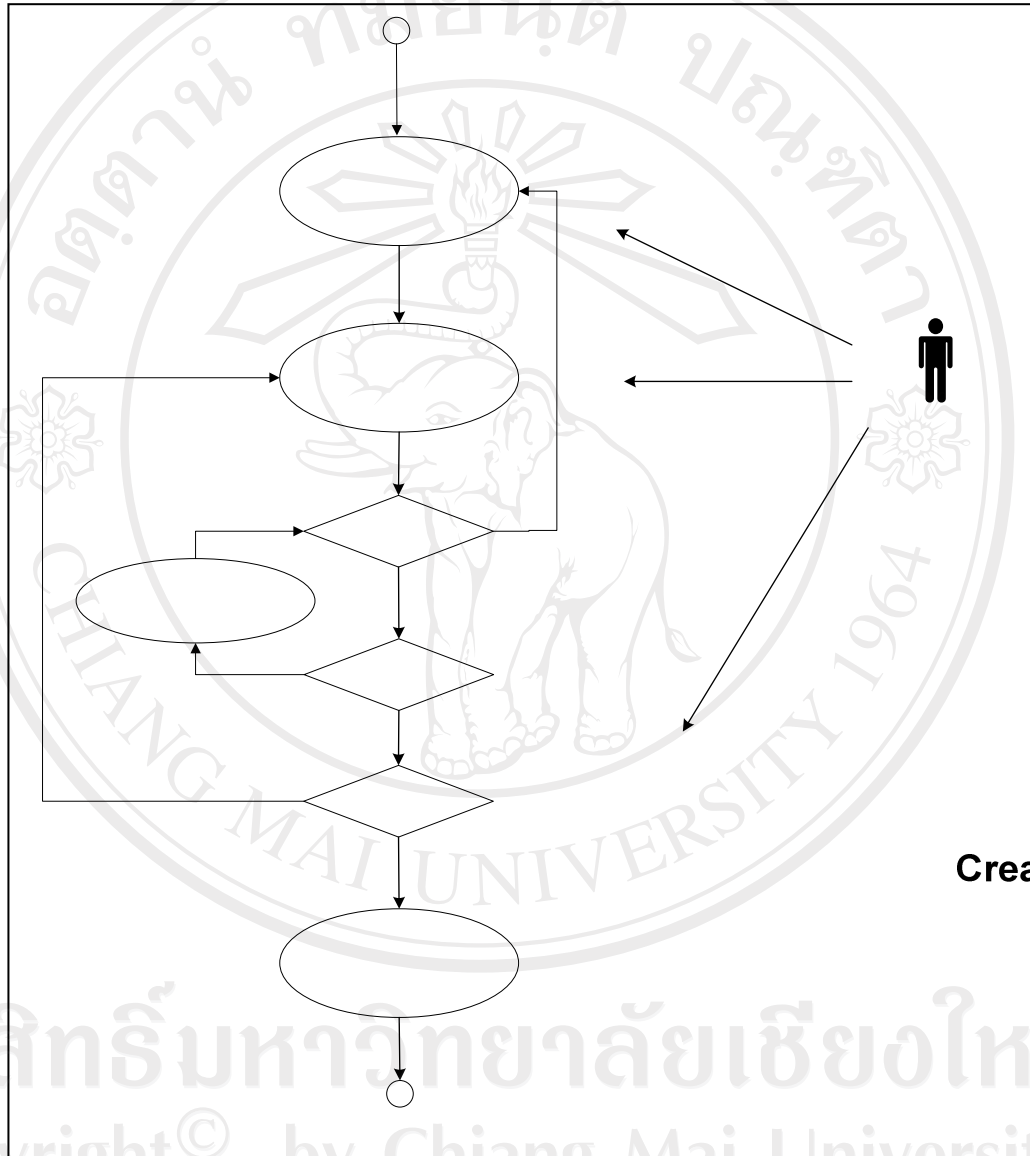
จากรูป 4.29 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการทำการปรับปรุงเอกสารด้าน Requirement ซึ่งมีขั้นตอนการทำงานดังนี้

1. BSA จะนำ Use-Case Specification มาสร้าง SRS เพื่อให้สอดคล้องกับความต้องการของ Software Requirement ที่สมบูรณ์
2. BSA ส่ง Software Requirement Specification ให้ PM ทำการ Review
3. BSA ส่ง Software Requirement Specifications ให้ SQA ทำการ Review
4. BSA ส่ง Software Requirement Specification ให้ SM ทำการ Approve
5. BSA จะไปพบกับ Customer โดยนำเอา Software Requirement Specification, ให้ Customer ทำการ Approve
6. BSA ส่ง Software Requirement Specification ให้กับ SCM เพื่อทำการจัดเก็บและควบคุม
7. PM ขอ Setup Baseline กับ SCCB โดยจะประกอบไปด้วยเอกสาร
 - Software Requirement Specification
 - Use-Case Specification

4.5 Software Quality Assurance

Software Quality Assurance หรือการตรวจสอบคุณภาพของซอฟต์แวร์ มีจุดประสงค์เพื่อรับประกันในเรื่องคุณภาพของซอฟต์แวร์ว่าถูกต้องตรงกับความต้องการของผู้ใช้ ดังนั้นจึงได้มีการออกแบบกระบวนการทำงานในส่วนของ Software Quality Assurance ขึ้นมาใหม่ โดยอิงมาตรฐาน CMM Level 2

โดยในกระบวนการทำงานแบบเดิมนั้นจะไม่พบขั้นตอนการทำงานรวมถึงผู้ที่มีหน้าที่รับผิดชอบในส่วนนี้ ดังนั้นส่วนของกระบวนการทำงาน Software Quality Assurance ที่ได้ออกแบบขึ้นมาใหม่ สามารถเขียนขั้นตอนการทำงานได้ดังรูป 4.30



รูป 4.30 Software Quality Assurance Workflow

จากรูป 4.30 ส่วนของกระบวนการทำงานของ Software Quality Assurance สามารถอธิบายถึงรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.31-4.33

Create/Revise SQ Plan

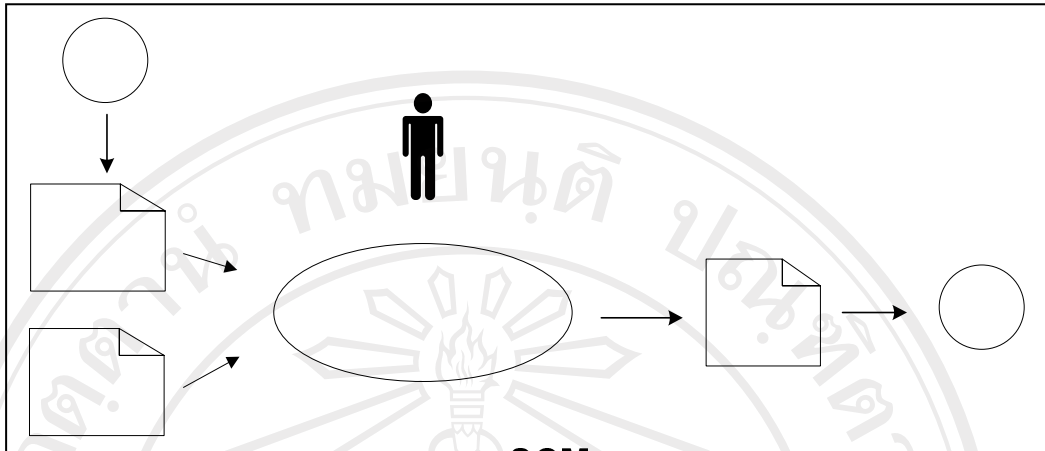
Review/Audit Product Process

Change to SQA Plan

Go to Change Management Process

No

Create/Revise SQA Plan



SCM

รูป 4.31 Create/Revise SQA Plan Artifacts Flow

จากรูป 4.31 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อสร้าง SQA Plan เพื่อใช้ในการทำงานของ SQA ซึ่งมีขั้นตอนการทำงานดังนี้

1. SQA สร้าง SQA Plan จาก SQA Plan Template โดยทำการวางแผนและประมาณ จำนวน ชั่วโมงการทำงานของ SQA สำหรับกิจกรรมของ SQA ให้สอดคล้องกับ SDP Schedule และกิจกรรมของ Project โดยภา **SDP Schedule** ร่วมกันกับ PM และ ได้ SPP แล้ว SQA จะใช้ SQA Plan ในการทำงานต่อไป

2. SQA ส่ง SQA Plan ที่ทำเสร็จแล้วให้ PM เพื่อเอาไปรวมเป็น SPP จากนั้นเข้าสู่ **Create/Revis Process:**

SQA

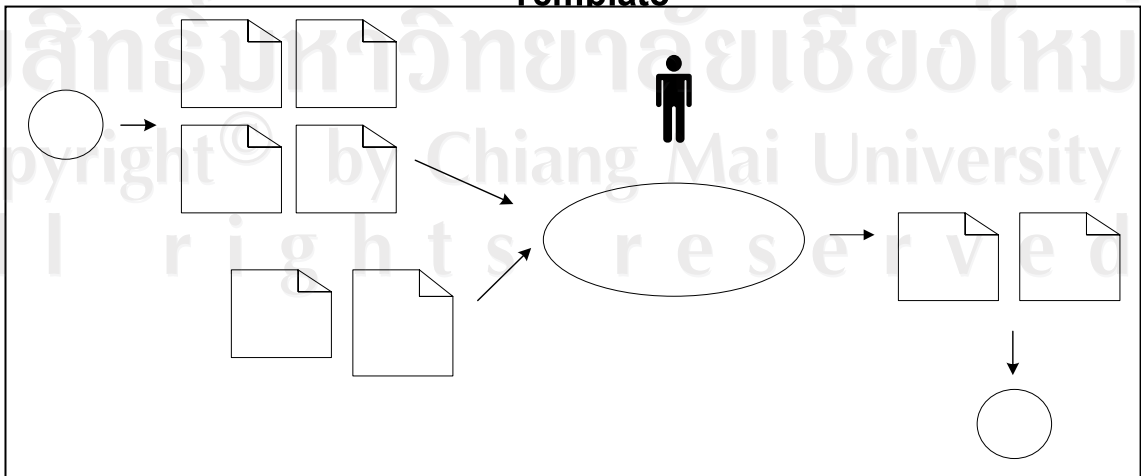
Create/Revis Plan

Create/Revise Software Project Plan

SQA Plan

Review/Audit Product & Process

Template

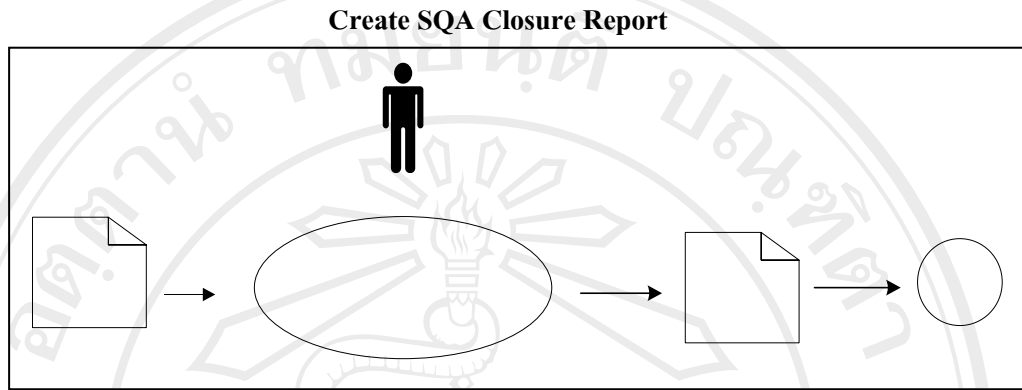


รูป 4.32 Review/Audit Product & Process Artifacts Flow

จากรูป 4.32 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อตรวจสอบการทำงานของ Project Team Member ที่เกี่ยวข้อง และเอกสารที่เกิดขึ้นในแต่ละกระบวนการซึ่งมีขั้นตอนการทำงานดังนี้

1. กรณีที่ SQA เข้าไป Review/Audit การทำงานของ Project Team Member ที่เกี่ยวข้อง และเอกสารที่เกิดขึ้นในแต่ละกระบวนการ
2. SQA ใช้ SQA Checklists Template สร้าง SQA Checklist ที่เกี่ยวข้องในแต่ละกระบวนการที่จะเข้าไปทำการ Review/Audit ตามรายการที่แสดงใน SQA Checklists Table
3. SQA ทำการ Review/Audit การทำงานและเอกสารในกระบวนการต่างๆ เปรียบเทียบกับ MSSCLC Process และ MSSCLC Templates ตามที่ได้กำหนดไว้ใน SQA Plan พร้อมทั้งให้ผู้ที่ถูกตรวจสอบเซ็นรับทราบผลการ Review/Audit ด้วย
4. เมื่อพบว่า มี non-compliance เกิดขึ้น SQA จะต้องแจ้ง non-compliance ให้ผู้ที่ถูกตรวจสอบทราบ พร้อมทั้งในวันและทำการ Re-Review/Audit อีกครั้ง หาก non-compliance นั้น ไม่ได้รับการแก้ไข หลังจากที่ SQA เข้าไป Re-Review/Audit แล้ว SQA จะรายงานต่อ SM เพื่อพิจารณาดำเนินการ SQA จะติดตามสถานะของ non-compliance จนกว่าจะถูกแก้ไขเรียบร้อย โดยสถานะของการ Review/Audit ประกอบด้วย
 - Yes = Process/เอกสาร ได้ทำตาม MSSCLC Process และ MSSCLC Templates
 - No = Process/เอกสาร ไม่ได้ทำตาม MSSCLC Process และ MSSCLC Templates
 - Not Applicable = ไม่มีการทำ Process/เอกสาร ใน project นี้
 - Defer = Process/เอกสาร จาก MSSCLC Process และ MSSCLC Templates ที่ยอมรับได้ โดยไม่ส่งผลกระทบต่อการทำงานของ project เช่น การพิมพ์ผิด
5. SQA ทำ SQA Status Report และรายงาน non-compliance , SQA Activities ให้ PM Review ตามระยะเวลาที่ได้ตกลงในแผนงานของแต่ละ Project โดย SQA จะรายงาน SQA Activities ที่ได้กระทำไป ในการประชุมกับ Project Team Member ตามระยะเวลาที่ได้ตกลงในแผนงานของแต่ละ Project
6. SQA จะทำการบันทึกเวลาที่ใช้ในการทำงานของ SQA และ Review/Audit แต่ละครั้งลงใน SQA Status Report
7. SQA รายงานการทำงานโดยตรงต่อ SM ในการประชุม หรือในกรณีที่มีเหตุการณ์เร่งด่วนเกิดขึ้น

8. SQA Status Report และ SQA Checklists ที่ทำการกรอกข้อมูลแล้วจะถูกเก็บและ ควบคุม โดย SCM



รูป 4.33 Create SQA Closure Report Artifacts Flow

จากรูป 4.33 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อสร้าง SQA Closure Report เพื่อรายงานต่อ PM ซึ่งมีขั้นตอนการทำงานดังนี้

1. SQA นำ SQA Closure Report Template ออกมาใช้
2. SQA ประเมินการทำงานทั้งหมดตลอด Project
3. SQA ทำ SQA Closure Report
4. SQA ส่ง SQA Closure Report ให้ PM เพื่อสรุปเป็น Closure Report ของ Project ต่อไป
5. SQA Closure Report ถูกเก็บและควบคุมโดย SCM

SQA

SQA

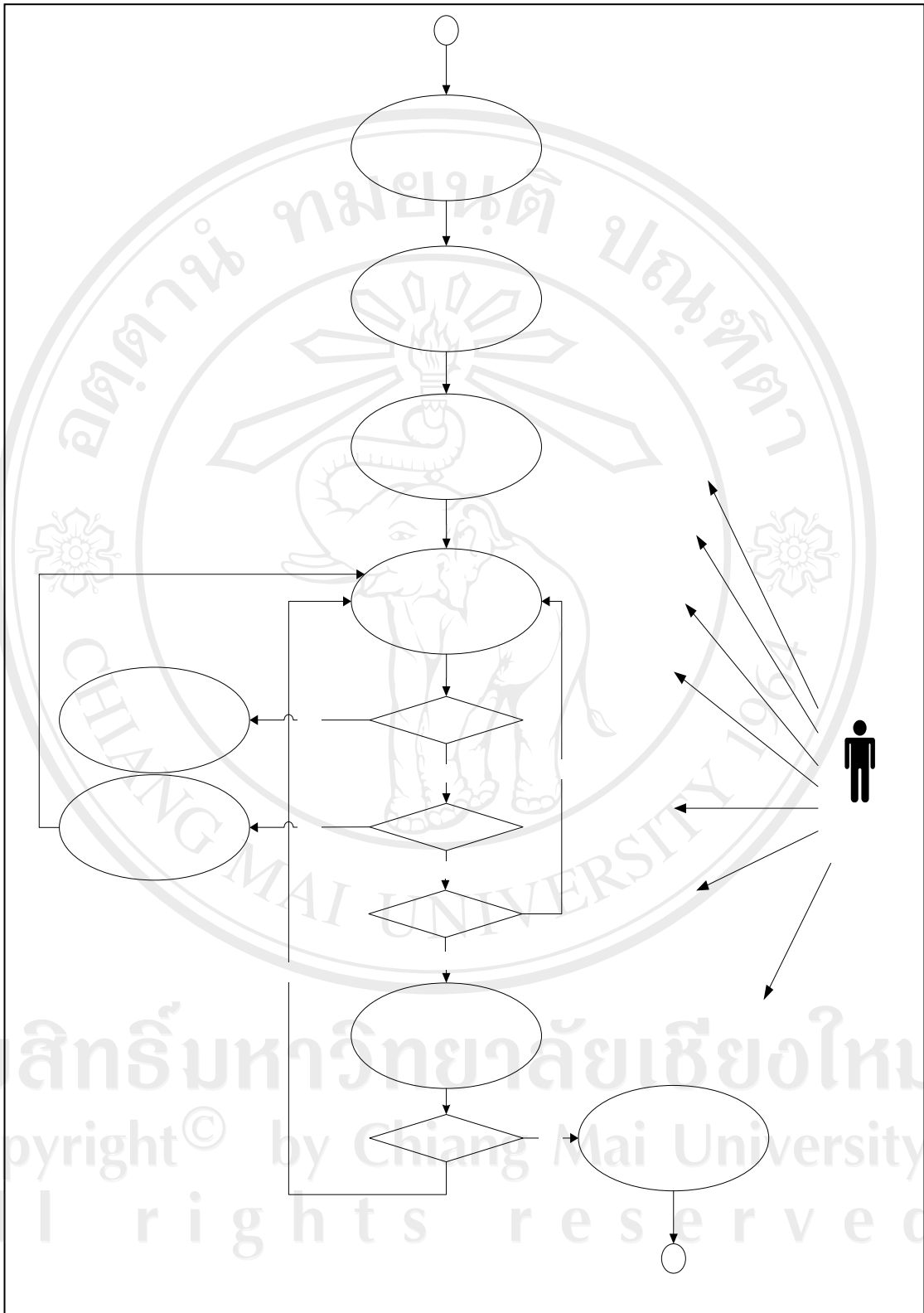
4.6 Software Configuration Management

Software Configuration Management หรือการพิจารณาทุกส่วนของการทำงานซอฟต์แวร์ มีจุดประสงค์เพื่อตรวจสอบเรื่องเทคนิคในส่วนของการระบุค่าข้อมูลต่างๆ ทั้งเอกสารและ Source Code ต่างๆ ดังนั้นจึงได้มีการออกแบบกระบวนการทำงานในส่วน of Software Configuration Management ขึ้นมาใหม่ โดยอิงมาตรฐาน CMM Level 2

โดยในกระบวนการทำงานแบบเดิมนั้นจะไม่พบขั้นตอนการทำงานที่แน่นอนชัดเจน รวมถึงผู้ที่ทำหน้าที่รับผิดชอบในส่วนนี้ ดังนั้นส่วนของกระบวนการทำงาน Software Configuration Management ที่ได้ออกแบบขึ้นมาใหม่ สามารถเขียนขั้นตอนการทำงานได้ดังรูป 4.34

SQA
Closure
Report
Template

Create SQA Cl
Report



Allocat
fo

Config

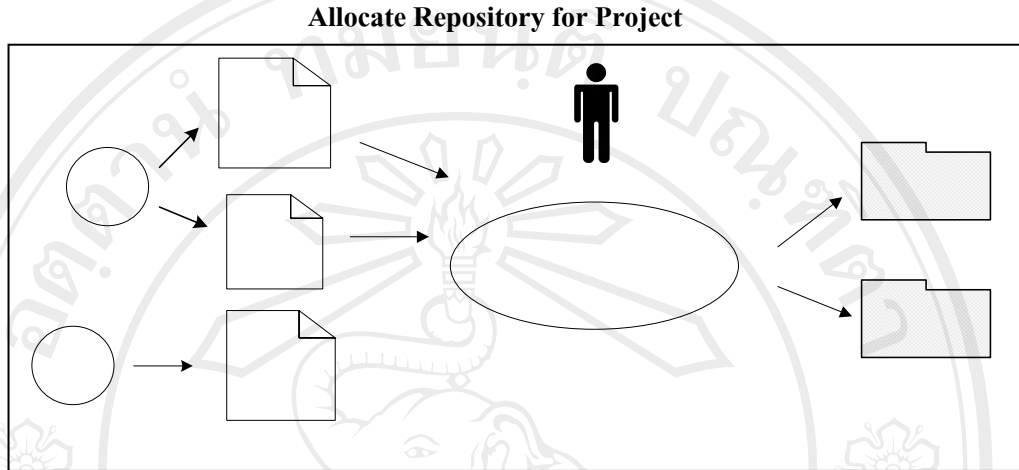
Create

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

รูป 4.34 Software Configuration Management Workflow

Perform
and
Pr

จากรูป 4.34 ส่วนของกระบวนการทำงานของ Software Configuration Management สามารถอธิบายถึงรายละเอียดของแต่ละกระบวนการได้ดังรูป 4.35-4.41

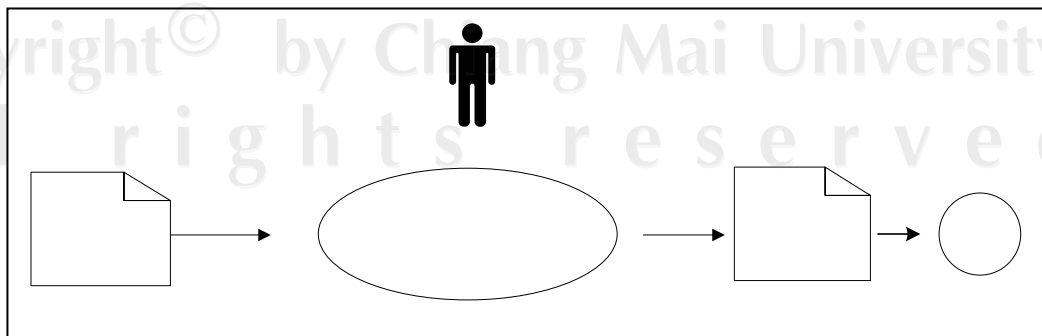


รูป 4.35 Allocate Repository for Project Artifacts Flow

จากรูป 4.35 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อสร้าง จัดสรรพื้นที่ในการจัดการควบคุมและจัดเก็บ Configuration Items ต่างๆ ใน Project ซึ่งมีขั้นตอนการทำงานดังนี้

1. หลังจากที่ SCM ได้รับการแต่งตั้งใน Project Team Assignment แล้ว **PM** สร้าง Project Folder และ Subfolder สำหรับ Project โดย PM สามารถขอแก้ไขเปลี่ยนแปลงได้
2. SCM ทำการจัดสิทธิการเข้าถึง Project Folder และ Subfolder ตามหน้าที่ของแต่ละบุคคล
3. SCM ทำการแจ้งให้ทุกคนภายในทีมรับทราบถึง Project Folder และ Subfolder ที่ได้จัดสร้างเรียบร้อยแล้ว **SCM User Request**

Identify Configuration Items

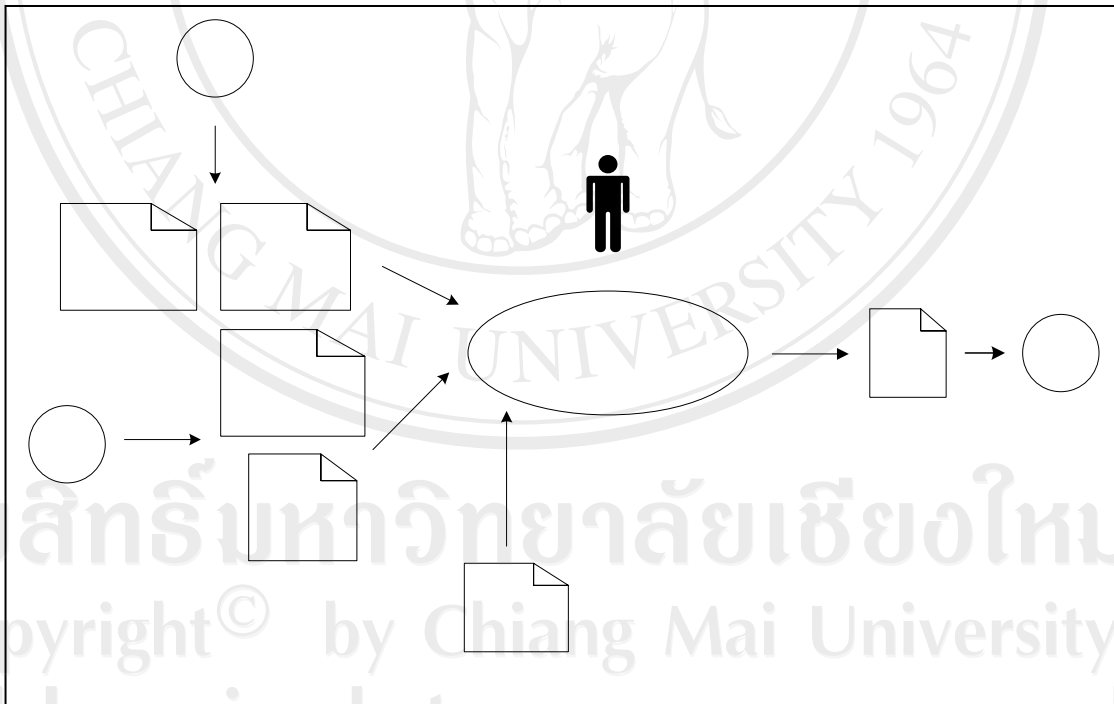


รูป 4.36 Identify Configuration Items Artifacts Flow

จากรูป 4.36 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อกำหนด Configuration Items ที่เกิดขึ้นใน Project ซึ่งมีขั้นตอนการทำงานดังนี้

1. SCM กำหนด Default Configuration Items ที่จำเป็นต้องใช้ใน Project ตาม Default Configuration Items ที่อยู่ในคู่มือประกอบการทำงานหรือ SCM Guidelines ในส่วนรายการเอกสารโดยใช้ Configuration Items List Template ในการบันทึก
2. PM กำหนด Configuration Items ที่นอกเหนือจาก Default Configuration Item ยกตัวอย่างเช่น Development Tools, Database Version and Etc โดยอ้างอิงจากคู่มือประกอบการทำงานหรือ SCM Guidelines ในส่วนรายการเอกสาร แล้วส่งให้ SCM บันทึกเพิ่มเติมลงใน Configuration Items List.
3. PM พิจารณาและอนุมัติ Configuration Items List
4. Configuration Items List ถูกจัดเก็บและควบคุมโดย SCM

Create/Revise SCM Plan



รูป 4.37 Create/Revise SCM Plan Artifacts Flow

จากรูป 4.37 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อสร้าง Plan ในการทำงานของ SCM ซึ่งมีขั้นตอนการทำงานดังนี้

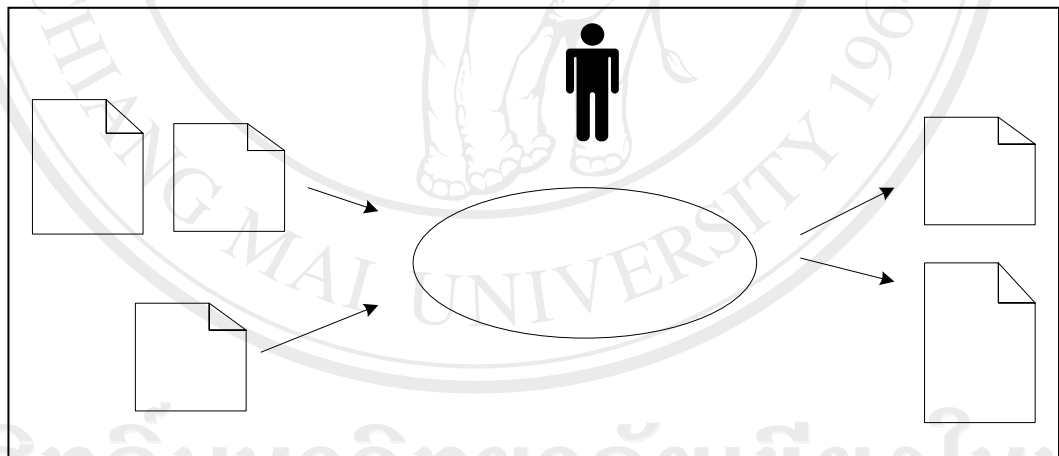
กรณี Create SCM Plan

1. SCM ใช้ SCM Plan Template มาทำการสร้าง SCM Plan โดยให้สอดคล้องกับ Software Development Plan Schedule และ Software Development Plan ซึ่ง การ Estimate effort จะอ้างอิงจากคู่มือประกอบการทำงาน (SCM Guidelines)
2. SCM ส่ง SCM Plan ให้ PM และ SQA ทำการ Review
3. SCM ส่ง SCM Plan ให้ SM Review และ Approve
4. SCM ส่ง SCM Plan ให้ PM เพื่อนำไปประกอบในการทำ Project Plan ต่อไป
5. SCM Plan ถูกจัดเก็บและควบคุมโดย SCM

กรณี Revise SCM Plan

1. SCM นำ SCM Plan มาทำการ Review โดยให้สอดคล้องกับ Revised Software Development Plan และ Revised Software Development Plan Schedule แล้วดำเนินการเหมือนกรณี Create SCM Plan ตั้งแต่ข้อ 2-5

Setup/Re-Setup Baseline



รูป 4.38 Setup/Re-Setup Baseline Artifacts Flow

จากรูป 4.38 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อทำการ Setup/Re-Setup Configuration Items ที่กำหนดเข้าสู่ Baseline ซึ่งมีขั้นตอนการทำงานดังนี้

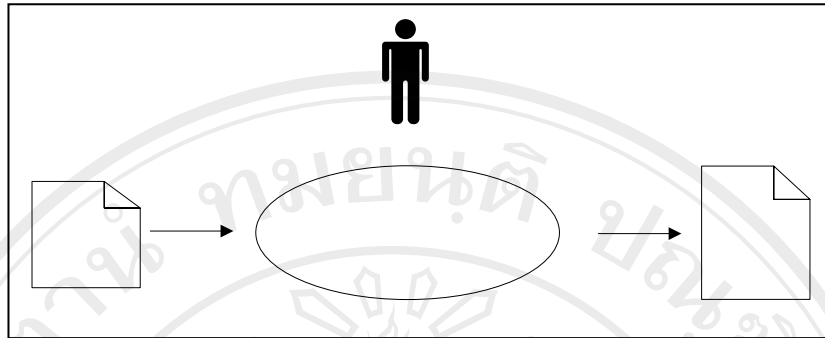
กรณี Setup Baseline

1. เมื่อ PM Request ให้มีการ Setup Baseline SCM จะทำการจัดเตรียมเอกสาร SCM Request Setup Baseline แล้วนัดประชุม SCCB เพื่อพิจารณา Configuration Items ที่จะทำการ Setup Baseline
2. ถ้า SCCB Approve การ Setup Baseline SCM จะทำการย้าย Configuration Items ที่มีการขอ Setup Baseline จาก Folder Docs (อ้างอิงคู่มือประกอบการทำงานหรือ SCM Guidelines) เข้าสู่ Folder Baseline แล้วทำการบันทึกลงใน List of Baselines พร้อมทั้ง Set Version ของ Baseline ด้วย ถ้า SCCB ไม่ Approve PM จะเป็นผู้รับผิดชอบในการแก้ไขให้ถูกต้อง แล้วกลับไปทำตั้งแต่ข้อ 1 ใหม่
3. เมื่อมีการ Setup Baseline เสร็จแล้ว SCM จะส่ง List of Baselines ให้ PM เพื่อให้ PM แจ้งต่อ Project Team Members เพื่อให้ใช้ Configuration Items ที่อยู่ใน Baseline Folder ในการทำงานต่อไป

กรณี Re-Setup Baseline

1. PM ส่ง Approved Change Request ที่มีผลกระทบต่อ Baseline และ Configuration Items ที่ได้รับการแก้ไขแล้ว ให้ SCM เพื่อจัดเตรียมเอกสาร SCM Request Setup Baseline
2. SCM นัดประชุม SCCB เพื่อพิจารณาการ Re-Setup Baseline
3. ถ้า SCCB Approve การ Re-Setup Baseline SCM จะทำการ Check out Configuration Items Version เดิม จาก Baseline Folder ไปไว้ใน Backup Folder และ Check in Configuration Items Version ใหม่ เข้าไปใน Baseline Folder แล้วทำการบันทึกลงใน List of Baselines พร้อมทั้ง Set Version ของ Baseline ด้วย ถ้า SCCB ไม่ Approve PM จะเป็นผู้รับผิดชอบในการแก้ไขให้ถูกต้อง แล้วกลับไปทำตั้งแต่ข้อ 1 ใหม่
4. เมื่อมีการ Setup Baseline เสร็จแล้ว SCM จะส่ง List of Baselines ให้ PM เพื่อให้ PM แจ้งต่อ Project Team Members เพื่อให้ใช้ Configuration Items ที่อยู่ใน Baseline Folder ในการทำงานต่อไป

Prepare Deliverable Configuration Items



รูป 4.39 Prepare Deliverable Configuration Items Artifacts Flow

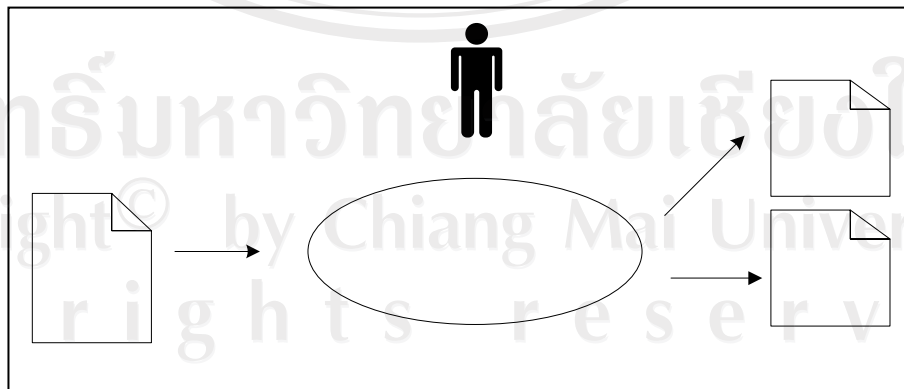
จากรูป 4.39 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อเตรียม Configuration Items ที่จะ Release ส่งให้กับ Customer ซึ่งมีขั้นตอนการทำงานดังนี้

1. Support ทำ Deliverable items List ส่งให้ PM ทำการ review และ Approve แล้วส่งให้ SCM เพื่อให้ SCM ทำการจัดเตรียม Configuration Items จาก Baseline ล่าสุด ตาม Deliverable items List ที่ได้รับมา
2. SCM นัดประชุม SCCB เพื่อให้พิจารณา Approve Deliverable Item List
3. SCM จะบันทึก Configuration Items ลงบนแผ่น CD-ROM และพิมพ์เอกสารประกอบ เพื่อสร้าง package ให้ SQA ทำการ Audit
4. หลังจาก SQA ทำการ Audit ความถูกต้องของ package แล้วให้ support นำส่งลูกค้าต่อไป

**Deliverable
Items List**

**Prepare
Configu**

Perform CM Activity



จากรูป 4.40 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อให้ SCM สามารถดำเนินงานได้อย่างมีประสิทธิภาพตลอด Project ซึ่งมีขั้นตอนการทำงานดังนี้

Sub Procedure: Check In/Out**Input:** SCM Check In Template or SCM Check Out Template**Output:** SCM Check in or SCM Check out**Work Description:**

อ้างอิงถึงคู่มือประกอบการทำงานหรือ SCM Guidelines ในหัวข้อการ Check In / Check Out Configuration Items

Sub Procedure: CIs and Baseline Backup and Recovery**Input:** CIs ที่อยู่ใน Configurations Repository**Output:** CIs ที่อยู่ใน Backup Folder**Work Description:**

1. SCM จะทำการ Copy CIs ทั้งหมดจาก Folder Docs และ Folder Baseline ของแต่ละ Project ใน Configurations Repository ไปเก็บไว้ใน Folder SCMBackup ของแต่ละ Project สัปดาห์ละหนึ่งครั้ง (อ้างอิงคู่มือประกอบการทำงานหรือ SCM Guidelines ในหัวข้อการสำรองและการกู้คืนข้อมูลของ Project)
2. ในกรณีที่ข้อมูลของ Project มีการสูญหาย SCM จะทำการแจ้งให้กับ PM เพื่อให้แจ้งต่อให้กับ Project Team Members รับทราบจากนั้น SCM จะทำการกู้ข้อมูลของ Project โดยการ Copy ข้อมูลจาก Folder SCM Backup ของ Project ดังกล่าว ครั้งล่าสุดมาไว้ใน Folder Docs และ Folder Baseline (อ้างอิงคู่มือประกอบการทำงานหรือ SCM Guidelines ในหัวข้อการสำรองและการกู้คืนข้อมูลของ Project) เมื่อเสร็จแล้ว SCM จะแจ้งให้กับ PM รับทราบเพื่อที่จะแจ้งต่อให้กับ Project Team Members ทราบต่อไป

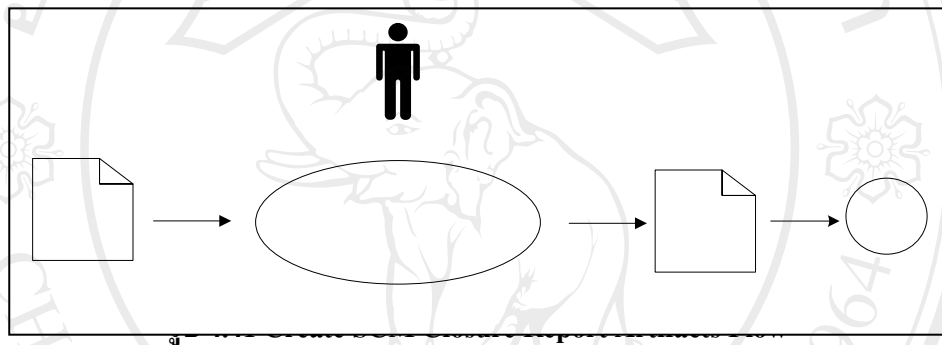
Sub Procedure: CIs and Baseline Audit**Input:** CIs ที่อยู่ใน Configurations Repository, Configuration Items List, List of Baselines, SCM Request Setup Baseline, SCM Status Report Template, SDPS**Output:** SCM Status Report**Work Description:**

1. SCM จะทำการตรวจสอบ CIs และ Baselines ใน Configuration Repositories ว่ามีการ Check In หรือ Check Out CIs ครบถ้วนและถูกต้องตามที่ได้กำหนดไว้ใน DS (Draft Schedule) หรือ SDPS หรือไม่ โดยอ้างอิงความถูกต้องของ CIs และ Baselines ใน Configuration Repositories จากเอกสาร Configuration Items List, SCM Request Setup

Baseline, SCM Check In, SCM Check Out และ List of Baselines เปรียบเทียบกับ CIs และ Baselines ที่ถูกจัดเก็บใน CM Repository

2. ในแต่ละสัปดาห์ SCM จะบันทึกรายงานการ Audit และ Status ของ CIs และ Baselines ลงใน SCM Status Report และส่งมอบให้กับ PM เพื่อทำการ Approve และส่งต่อให้กับ SQA และ SM เพื่อทำการ Review
3. หากเกิดปัญหาเกี่ยวกับ CIs หรือ Baseline ขึ้น SCM จะระบุไว้ใน SCM Status Report และจะทำการ Re-Audit จนกว่าปัญหาจะได้รับการแก้ไขจนเสร็จสิ้น

Create SCM Closure Report



จากรูป 4.41 แสดงถึงลักษณะการไหลของเอกสาร โดยขั้นตอนนี้มีจุดประสงค์เพื่อสร้าง SCM Closure Report เพื่อรายงานต่อ PM ซึ่งมีขั้นตอนการทำงานดังนี้

1. SCM นำ SCM Closure Report Template ออกมาใช้
2. SCM ประเมินการทำงานทั้งหมดตลอด Project
3. SCM ทำ SCM Closure Report
4. SCM ส่ง SCM Closure Report ให้ PM เพื่อสรุปเป็น Closure Report ของ Project ต่อไป
5. SCM Closure Report ถูกจัดเก็บและควบคุมโดย SCM

ลิขสิทธิ์ © โดย Chiang Mai University
All rights reserved

SCM

SCM
Closure
Report
Template

Create SCM
Closure