

# บทที่ 1

## บทนำ

### 1.1 หลักการและเหตุผล

ในปัจจุบันนี้ซอฟต์แวร์มีขนาดใหญ่ขึ้นและซับซ้อนมากขึ้นจนยากที่จะพัฒนาระบบเหล่านี้ได้ด้วยตัวเอง การพัฒนาซอฟต์แวร์ร่วมกันเป็นทีมจึงจำเป็นจะต้องมีกฎเกณฑ์และแนวทางในการทำงานอย่างรัดกุมการพัฒนาซอฟต์แวร์หรือโปรแกรมในอดีตนั้นถือกันว่าเป็นศิลปะแบบหนึ่งเมื่อเป็นศิลปะจึงยากที่จะกำหนดกฎเกณฑ์ว่าใครจะทำอะไรก่อนอะไรหลัง จำทำอย่างไร หรือจะควบคุมตรวจสอบอย่างไร ในเมื่อซอฟต์แวร์หรือโปรแกรมเป็นงานสร้างสรรค์ทางปัญญาที่ซับซ้อนการที่จะสื่อความกันระหว่างการพัฒนาโปรแกรมจึงเป็นเรื่องยาก งานซอฟต์แวร์ขนาดใหญ่จึงมักจะประสบกับความล้มเหลวมากกว่าความสำเร็จ

กระบวนการซอฟต์แวร์ของบริษัทและหน่วยงานต่าง ๆ นั้นมีความแตกต่างกันมาก ขึ้นกับความสามารถในการบริหารงานซอฟต์แวร์ และบุคลากรผู้พัฒนาซอฟต์แวร์ บริษัทและหน่วยงานบางแห่งสามารถพัฒนาซอฟต์แวร์ได้ดี แต่บริษัทและหน่วยงานอื่นที่มีสมบัติคล้ายกันอาจไม่สามารถพัฒนาซอฟต์แวร์ได้ดีเท่า

การพัฒนาองค์กรสู่ความเป็นเลิศในด้านการพัฒนาซอฟต์แวร์นั้นจะพบว่าประเด็นสำคัญของการพัฒนาซอฟต์แวร์นั้นไม่ได้อยู่ที่ด้านเทคนิค แต่อยู่ที่กระบวนการจัดการ หากหน่วยงานใดมีผู้บริหารที่เอาใจใส่ในเรื่องคอมพิวเตอร์ และสนใจต้องการพัฒนาซอฟต์แวร์ที่มีคุณภาพ ตลอดจนได้กำหนดวิธีการและขั้นตอนในการบริหารงานในส่วนนี้อย่างเป็นระบบแล้ว หน่วยงานนั้นก็จะมีโอกาสเลื่อนระดับขึ้นไปสู่ความเป็นเลิศได้ แต่หน่วยงานใดไม่มีผู้บริหารที่สนใจด้านนี้ คงมีแต่เพียงนักคอมพิวเตอร์เท่านั้น หน่วยงานเช่นนี้ก็คงจะไต่ระดับได้เพียงแต่ระดับหนึ่งหรือสองอย่างแน่นอน

การนำ CMM (Capability Maturity Model) มาช่วยเป็นแนวทางหนึ่งที่จะทำให้หน่วยงานที่ทำงานด้านซอฟต์แวร์มีแนวทางสำหรับควบคุมกระบวนการในการพัฒนาและดูแลรักษาซอฟต์แวร์ และแนวทางสำหรับทำให้หน่วยงานก้าวหน้าขึ้นไปถึงระดับขั้นที่มีความเป็นเลิศทางด้านวิศวกรรมซอฟต์แวร์ และการจัดการซอฟต์แวร์แบบจำลองนี้ช่วยให้หน่วยงานซอฟต์แวร์สามารถเลือกกลยุทธ์ในการปรับปรุงกระบวนการซอฟต์แวร์โดยการกำหนดว่าปัจจุบันหน่วยงานมีระดับความเจริญก้าวหน้าอยู่ขั้นใด แล้วจึงกำหนดประเด็นต่างๆที่มีความสำคัญต่อการปรับปรุงคุณภาพและกระบวนการซอฟต์แวร์ โดยวิธีให้ความสนใจเฉพาะในประเด็นที่สำคัญบางเรื่องจะ

ช่วยให้หน่วยงานปรับปรุงกระบวนการซอฟต์แวร์ได้อย่างต่อเนื่องโดยตลอด และสามารถยกระดับความเจริญก้าวหน้าได้อย่างยั่งยืน

## 1.2 วัตถุประสงค์ของการศึกษา

เพื่อออกแบบกระบวนการพัฒนาซอฟต์แวร์เพื่อเตรียมความพร้อมขององค์กรให้เข้าสู่มาตรฐาน CMM Level 2

## 1.3 ประโยชน์ที่ได้รับจากการศึกษา

- 1) ได้กระบวนการพัฒนาซอฟต์แวร์เพื่อเตรียมความพร้อมขององค์กรให้เข้าสู่มาตรฐาน CMM Level 2
- 2) องค์กรมีกระบวนการพัฒนาซอฟต์แวร์ มีระบบขั้นตอนที่ชัดเจน สามารถควบคุมและตรวจสอบได้
- 3) องค์กรมีการทำงานเป็นระบบขั้นตอนที่เหมือนกัน มีเอกสารเป็นหลักฐานในการทำงาน เพื่อให้ตรวจสอบได้ง่ายขึ้นและสมบูรณ์มากขึ้น

## 1.4 ขอบเขต และวิธีการศึกษา

- 1) ขอบเขตการศึกษา
  - 1.1) ออกแบบ Software Development Process สำหรับบริษัท เมนชา ซอฟต์แวร์ แอนด์ คอนซัลตติ้ง จำกัด ตามมาตรฐาน CMM Level 2 ซึ่งจะทำการพัฒนา 5 กระบวนการ โดยจะไม่ดำเนินการในส่วนของ Software Subcontract Management (SSM) เนื่องจากบริษัทไม่ได้มีการจ้าง Outsource หรือจัดจ้าง
  - 1.2) Software Project Plan จะต้องสามารถบอกถึงระยะเวลาของ Project เพื่อใช้ในการจัดการกระบวนการพัฒนาระบบได้ รวมถึงการ Estimate Size, Effort และ Schedule
  - 1.3) Software Project Tracking and Oversight บอกถึงขั้นตอนการติดตามการทำงาน การพัฒนาระบบ
  - 1.4) Software Quality Assurance ตรวจสอบความถูกต้องของกระบวนการทำงานของทีมและเอกสารที่สร้างและใช้ในการพัฒนาระบบ
  - 1.5) Software Configuration Management สามารถควบคุม ดูแลและรักษาเอกสาร ข้อมูล รวมถึงกระบวนการขององค์กรได้

1.6) Requirement Management บอกวิธีการจัดการความต้องการของลูกค้าและขอบเขตของระบบ

2) วิธีการศึกษา

2.1) ศึกษาข้อกำหนดและรายละเอียดของ CMM Level 2

2.2) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Software Project Planning

- ทำการสร้างกระบวนการทำงานในขั้นตอน Software Project Planning
- สร้างเอกสารที่เกี่ยวข้องกับ Software Project Planning เช่น ทำการสร้าง Draft Schedule สำหรับช่วง Requirement, ทำการสร้างและปรับปรุง Software Development Plan โดยการใช้ Estimate Size, Effort และ Schedule
- มีการระบุถึงการวิเคราะห์ Project Scope และ Project Risk เพื่อค้นหาและระบุความเสี่ยงที่อาจเกิดขึ้นใน Project และทำ Business Case

2.3) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Software Project Tracking and Oversight

- ทำการออกแบบขั้นตอนการติดตามความคืบหน้าของ Project ในแต่ละ Phase
- สร้างเอกสารที่เกี่ยวข้องกับ Software Project Tracking and Control

2.4) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Software Quality Assurance

- ทำการออกแบบขั้นตอนการตรวจสอบและติดตามของ SQA
- ทำการสร้าง SQA Plan เพื่อใช้ในการทำงานของ SQA
- ทำรายงานการตรวจสอบการทำงานของ Project Team Member ที่เกี่ยวข้องและเอกสารที่เกิดขึ้นในแต่ละ Process ให้แก่ Project Manager, Senior Manager

2.5) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Software Configuration Management

- ทำการสร้าง SCM Plan เพื่อใช้ในการทำงานของ SCM
- กำหนดกระบวนการกำหนด Baseline ให้กับ Configuration Items
- กำหนดขั้นตอนการควบคุมและจัดเก็บ Configuration Items

2.6) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Requirement Management

- ออกแบบกระบวนการจัดการความต้องการของลูกค้า
- วิเคราะห์ความต้องการของลูกค้าและทำการกำหนดขอบเขตของระบบ

2.7) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Analysis and Design

- 2.8) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Implementation
- 2.9) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Test
- 2.10) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Change Management
- 2.11) วิเคราะห์ออกแบบกระบวนการทำงานในขั้นตอน Deployment
- 2.12) ตรวจสอบและปรับปรุงกระบวนการทำงานให้เหมาะสมกับการทำงาน โดยทำการทดลองใช้กับ Project จริง
- 2.13) จัดทำเอกสารอธิบายกระบวนการทำงาน

### 1.5 สถานที่ใช้ในการดำเนินการศึกษาและรวบรวมข้อมูล

- 1) บริษัท เมนชา ซอฟต์แวร์ แอนด์ คอนซัลติง จำกัด
- 2) สำนักหอสมุด มหาวิทยาลัยเชียงใหม่
- 3) สาขาวิชาเทคโนโลยีสารสนเทศและการจัดการ บัณฑิตวิทยาลัยมหาวิทยาลัยเชียงใหม่

### 1.6 เครื่องมือที่ใช้ในการศึกษา

- 1) ฮาร์ดแวร์
  - เครื่องคอมพิวเตอร์ส่วนบุคคลเพื่อใช้ในการศึกษาและพัฒนามาตรฐาน มีคุณสมบัติ ดังนี้
    - หน่วยประมวลผลกลางแบบเพนเทียม 3 ความเร็ว 500 เมกะเฮิร์ตซ์
    - หน่วยความจำขนาด 128 เมกะไบต์
    - ฮาร์ดดิสก์ขนาด 20 กิกะไบต์
- 2) ซอฟต์แวร์
  - ระบบปฏิบัติการ
    - ใช้ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 98 เป็นต้นไป
  - โปรแกรมสำเร็จรูป
    - ไมโครซอฟท์ออฟฟิศ : เป็นซอฟต์แวร์ที่ช่วยในการออกแบบเอกสาร
    - ไมโครซอฟท์โปรเจ็ค : เป็นซอฟต์แวร์ที่ช่วยในการจัดทำตารางและแผนการดำเนินงาน
    - ไมโครซอฟท์วิซิโอ : เป็นซอฟต์แวร์ที่ช่วยในการออกแบบขั้นตอนการทำงาน ออกแบบระบบ