

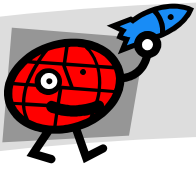
Requirement Phase

ความต้องการ (Requirement)

“วัตถุดิบที่สำคัญในการพัฒนาระบบหรือการผลิตซอฟต์แวร์ เพื่อใช้เป็นข้อกำหนดถึงหน้าที่และรายละเอียดอื่น ๆ ที่ระบบหรือซอฟต์แวร์จะต้องมี”

ในการพัฒนาระบบหรือการผลิตซอฟต์แวร์ ต้องอาศัยข้อมูลความต้องการของลูกค้าหรือผู้ใช้ เป็นตัวกำหนด ฟังก์ชัน รูปลักษณะ ความสามารถ และรายละเอียด แต่เนื่องจากการพัฒนามีหลายแนวทางเช่น พัฒนาเอง จ้าง เหมา หรือ ใช้ซอฟต์แวร์สำเร็จรูป หรือ ผลิตเพื่อจำหน่าย จึงทำให้ข้อกำหนดความต้องการมีหลายระดับ หากจำแนกระดับความต้องการไม่ถูกต้อง จะส่งผลให้ซอฟต์แวร์ไม่ตรงกับความต้องการที่แท้จริง





Requirement Phase

ในแวดวงการผลิตซอฟต์แวร์ สามารถจัดความต้องการได้ 2 ระดับ

1. ความต้องการของผู้ใช้ (User Requirement)

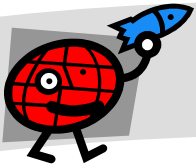
เป็นความต้องการของลูกค้าหรือผู้ที่เกี่ยวข้องกับระบบ โดยแสดงออกมาในรูปภาษาธรรมชาติ ที่แสดงถึงการคาดหวังในบริการหรือการทำงานที่ได้รับจากระบบ

2. ความต้องการด้านระบบ (System Requirement)

เป็นการกำหนดความต้องการของการทำงาน ฟังก์ชัน และบริการต่างๆ ของระบบในระดับรายละเอียดที่เรียกว่า

“Functional Specification”





Requirement Phase

ประเภทความต้องการด้านซอฟต์แวร์

1. ความต้องการที่เป็นหน้าที่หลัก (Functional Requirement)

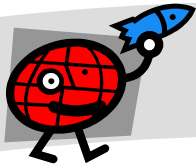
คือความต้องการให้ซอฟต์แวร์ทำหน้าที่ใด ๆ ตามที่กำหนดไว้ ซึ่งก็คือสิ่งที่ซอฟต์แวร์ควรทำเป็นหลักในการทำงานหรือเป็นบริการที่ซอฟต์แวร์ต้องมี เช่น

- นักศึกษาสามารถตรวจสอบผลการเรียนได้ และการลงทะเบียนเรียนได้
- นักศึกษาสามารถตรวจสอบตารางเรียนของตนเองได้ หรือตารางสอนอาจารย์ได้
- นักศึกษาสามารถเพิ่มรายวิชาเรียนหรือเพิกถอนรายวิชาเรียนได้

Functional Requirement -> Complete & Consistency

หมายเหตุ : ในทางปฏิบัติเป็นไปได้ยาก โดยเฉพาะระบบที่มีความซับซ้อนสูง เนื่องจากความต้องการของเจ้าของหรือผู้ใช้แต่ละคนแตกต่างกัน อีกทั้งการให้ข้อมูลอาจกำกวมไม่ชัดเจน ทำให้เกิดปัญหาความเข้าใจที่ไม่ตรงกันได้





Requirement Phase

ประเภทความต้องการด้านซอฟต์แวร์

2. ความต้องการที่ไม่ใช่หน้าที่หลัก (non-Functional Requirement)

คือความต้องการที่ไม่ได้เกี่ยวข้องกับหน้าที่หรือฟังก์ชันหลักของระบบ แต่เกี่ยวข้องกับอ้อมในลักษณะที่เป็นเงื่อนไขของฟังก์ชันหรือบริการ และอาจไม่ได้เกี่ยวข้องกับซอฟต์แวร์เพียงอย่างเดียว ได้แก่

(1) ความต้องการด้านผลิตภัณฑ์ (Product Requirement)

ได้แก่ ความต้องการด้านประสิทธิภาพ , ความน่าเชื่อถือ และใช้งานง่าย

(2) ความต้องการขององค์กร (Organizational Requirement)

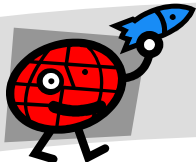
ได้แก่ ความต้องการเชิงนโยบายและระเบียบปฏิบัติของลูกค้าและผู้พัฒนา เช่น มาตรฐานการผลิต ภาษา เมธอดที่ใช้ หรือกำหนดเวลาส่งมอบ

(3) ความต้องการจากปัจจัยภายนอก (External Requirement)

ได้แก่ ความต้องการการทำงานร่วมกัน , ด้านกฎหมาย และจริยธรรม

หมายเหตุ : ความต้องการดังกล่าวต้องใช้ความรอบคอบเป็นพิเศษ เนื่องจากมีอิทธิพลต่อการตัดสินใจยอมรับซอฟต์แวร์หรือระบบของผู้ใช้อย่างมาก



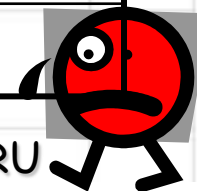


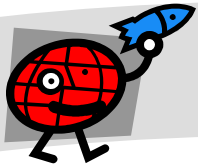
Requirement Phase

ประเภทความต้องการด้านซอฟต์แวร์

ตัวอย่างคุณลักษณะของระบบที่ใช้กำหนดความต้องการที่ไม่ใช่หน้าที่หลัก

คุณลักษณะ	หน่วยวัด
ความเร็ว	- การประมวลผลรายการข้อมูล (หน่วยเป็นวินาที) - ระยะเวลาตอบสนองต่อการใช้งาน - เวลาในการเปลี่ยนข้อมูลบนจอภาพ
ขนาด	- กิกะไบต์ (Gbytes) - ขนาดของหน่วยความจำแรม
ใช้งานง่าย	- ระยะเวลาที่ใช้ในการอบรมการใช้งาน - จำนวนของส่วนช่วยเหลือ (Help)
ความน่าเชื่อถือ	- ค่าเฉลี่ยของข้อผิดพลาดที่เกิดขึ้น - ความน่าจะเป็นของระบบที่ไม่สามารถใช้งานได้ - อัตราของการเกิดข้อผิดพลาด
ความสามารถในการทำงานข้ามระบบได้	- จำนวนของระบบอื่นที่ใช้งานได้





Requirement Phase

ความต้องการของผู้ใช้

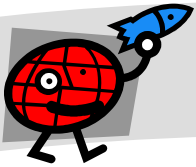
ความต้องการของผู้ใช้ (User Requirement)

: คือความต้องการที่มีต่อระบบของผู้ใช้ โดยจะอธิบายทั้งส่วนที่เป็นหน้าที่หลักและส่วนที่ไม่ใช่หน้าที่หลักของระบบ ด้วยภาษาที่ผู้ใช้เข้าใจ ไม่ควรใช้ศัพท์เทคนิคมากเกินไป เนื่องจากผู้ใช้จะเข้าใจพฤติกรรมของระบบของตนเท่านั้น สำหรับส่วนที่เกี่ยวข้องกับการออกแบบ สถาปัตยกรรม หรือคุณลักษณะของระบบ ผู้ใช้จะหลีกเลี่ยงและไม่สนใจ

ปัญหาที่มักเกิดขึ้นจากการเก็บความต้องการ

1. ขาดความชัดเจน เนื่องจากบางครั้งการเขียนคำอธิบายความต้องการให้กระชับ ไม่กำกวม โดยใช้ภาษาที่ไม่ใช่เชิงเทคนิค
2. ไม่สามารถจำแนกประเภทของความต้องการได้อย่างชัดเจน
3. บางครั้งความต้องการของผู้ใช้อาจมีจุดประสงค์เดียวกัน แต่เขียนออกมาในประโยคแตกต่างกัน ดังนั้นควรจัดกลุ่มความต้องการของผู้ใช้ที่เหมือนกันเข้าไว้ด้วยกัน





Requirement Phase

วิธีปฏิบัติด้านการสื่อสาร (Communication Practice)

กิจกรรมการติดต่อสื่อสารเป็นสิ่งที่ช่วยให้การรวบรวมความต้องการของลูกค้าเป็นไปได้ การติดต่อสื่อสารที่มีประสิทธิภาพ นับเป็นสิ่งที่ท้าทายมากที่สุด

เน้นหลักการสำคัญ (Core Principles) ในการสื่อสารกับลูกค้า

หลักที่ 1 ฟัง! (Listen!)

จงให้ความสำคัญกับทุกคำของลูกค้า ถ้ามีข้อสงสัยให้สอบถามเพื่อความกระจ่าง

หลักที่ 2 เตรียมตัวให้พร้อมก่อนเริ่มการติดต่อสื่อสาร (Prepare before you communicate)

พยายามที่จะเข้าใจปัญหาด้วยตนเองก่อน โดยอาจต้องมีการหาข้อมูลเพิ่มเติม

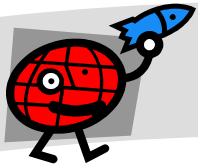
หลักที่ 3 ควรมีผู้ประสานงาน (Someone should facilitate the activity)

ในทุก ๆ การติดต่อสื่อสาร ควรจะมีผู้นำที่ทำหน้าที่ในการควบคุมการสนทนาให้มุ่งไปในทิศทางที่เกิดผล และเป็นผู้ประนีประนอมความขัดแย้งที่อาจเกิดขึ้น

หลักที่ 4 การติดต่อสื่อสารแบบพบหน้ากันดีที่สุด (Face-to-face communication is best)

โดยเฉพาะถ้ามีการเตรียมเอกสารประกอบอย่างดี เช่น โครงร่างของเรื่องที่จะพูด





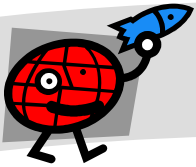
Requirement Phase

วิธีปฏิบัติด้านการสื่อสาร (Communication Practice)

เน้นหลักการสำคัญ (Core Principles) ในการสื่อสารกับลูกค้า (ต่อ)

- หลักที่ 5 จดรายละเอียดการพูดคุยรวมถึงข้อตกลงในเรื่องต่าง ๆ (Take notes and document decision)
- หลักที่ 6 พยายามให้เกิดความร่วมมือ (Strive for collaboration)
ความร่วมมือกันระหว่างสมาชิกในทีม เป็นช่วยก่อให้เกิดความไว้วางใจซึ่งกันและกัน อันจะนำไปสู่เป้าหมายร่วมกันของทีมได้
- หลักที่ 7 เน้นหัวข้อ ไม่แตกประเด็น (Stay focused, modularize your discussion)
การติดต่อสื่อสารใด ๆ ก็ตาม ยังมีผู้เกี่ยวข้องมาก ก็ยังมีโอกาสขัดแย้งมากหรือวนไปวนมา ต้องพยายามอย่าให้การสนทนาออกนอกประเด็น
- หลักที่ 8 ถ้ามีอะไรไม่ชัดเจน ให้วาดภาพประกอบ (If something is unclear, draw a picture)
กรณีที่มีการสื่อสารด้วยคำพูดทำให้เกิดความไม่ชัดเจน การวาดภาพ ตาราง หรือใช้รูปประกอบ อาจช่วยได้
- หลักที่ 9 พยายามให้การติดต่อสื่อสาร “เดินหน้า” (Move on)
ไม่ว่าจะได้ข้อสรุปหรือไม่ หรือว่ามีบางเรื่องที่ยังขาดความชัดเจน ก็ให้การสื่อสารเดินหน้าไปก่อน อย่าพูดซ้ำไปซ้ำมาหาข้อยุติไม่ได้ให้เปลืองเวลา





Requirement Phase

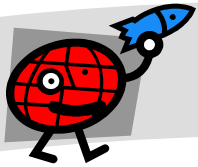
วิธีปฏิบัติด้านการสื่อสาร (Communication Practice)

เน้นหลักการสำคัญ (Core Principles) ในการสื่อสารกับลูกค้า (ต่อ)

หลักที่ 10 การต่อรองไม่ใช่การแข่งขันหรือเกม จะเป็นการดีที่สุด หากทั้งสองฝ่ายเป็นผู้ชนะ
(Negotiation is not a contest or a game, It works best when both parties win)

มีหลาย ๆ เรื่องที่นักวิศวกรซอฟต์แวร์ต้องต่อรองกับลูกค้า เช่น งานที่ซอฟต์แวร์ทำได้ ลำดับงานก่อนหลัง กำหนดวันส่งมอบ ทั้งนี้ จะต้องมีการร่วมมือร่วมใจ สร้างเป้าหมายร่วมกัน การต่อรองต้องอาศัยการประนีประนอมจากทั้งสองฝ่าย





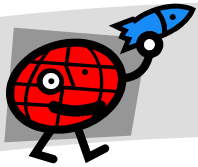
Requirement Phase

ความต้องการของผู้ใช้

การเขียนเอกสารความต้องการของผู้ใช้ควรมีหลักปฏิบัติ

1. กำหนดมาตรฐานของรูปแบบเอกสาร เช่น รูปแบบการใช้ตัวอักษร ขนาด สี และเน้นข้อความ การขีดเส้น ตัวเอง ควรใช้กับข้อความส่วนใด
(อย่าลืม!! กำหนดแหล่งที่มาของความต้องการ ผู้จัดทำ และวันที่จัดทำด้วย)
2. จำแนกความจำเป็นของความต้องการ โดยแบ่งออกเป็น “ความต้องการที่จำเป็น (Mandatory Requirement)” และ “ความต้องการที่เป็นความปรารถนา (Desirable Requirement)” โดยใช้คำนำหน้าความต้องการของทั้ง 2 ลักษณะ คือ นำหน้าว่า “**ต้อง**” สำหรับความต้องการที่จำเป็น และ “**ควร**” สำหรับความต้องการที่ปรารถนา
3. ในเอกสารควร**เน้นข้อความที่เป็นประเด็นสำคัญ**ของความต้องการให้เห็นเด่นชัด เพื่อให้ผู้ใช้มุ่งความสนใจอยู่ภายในขอบเขตของประเด็นนั้น
4. หลีกเลี่ยงการใช้คำศัพท์หรือรูปทางเทคนิคในเอกสารให้มากที่สุดเท่าที่จะทำได้ หากหลีกเลี่ยงไม่ได้ ควรหาศัพท์ที่เหมาะสมกับผู้ใช้มากที่สุด





Requirement Phase

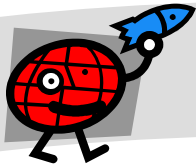
ความต้องการของระบบ

ความต้องการของระบบ (System Requirement) เป็นการกำหนดความต้องการทำงาน ฟังก์ชัน และการบริการต่าง ๆ ที่ระบบจะต้องมีเพื่อตอบสนองความต้องการของผู้ใช้ (หมายเหตุ : เกิดจากการวิเคราะห์ข้อมูลความต้องการของผู้ใช้มาแล้ว)

เหตุผลที่ต้องระบुरายละเอียดขึ้นการออกแบบไว้ในความต้องการของระบบ เพราะ

1. เพื่อช่วยให้การกำหนดโครงการของเอกสารความต้องการง่ายขึ้น เช่น ถ้าต้องการใช้กระบวนการพัฒนาแบบ Component Reuse ต้องออกแบบสถาปัตยกรรมของระบบก่อน เพื่อกำหนดคอมโพเนนต์ย่อย
2. ระบบจะต้องทำงานร่วมกับระบบอื่นที่มีอยู่แล้ว ต้องอาศัยรายละเอียดการออกแบบเข้ามาช่วยกำหนดความต้องการ
3. เพื่อให้สามารถตอบสนองความน่าเชื่อถือได้ บางครั้งจำเป็นต้องออกแบบสถาปัตยกรรมเฉพาะขึ้นมาก่อนเพื่อใช้เป็นกรอบการทำงานต่อไป





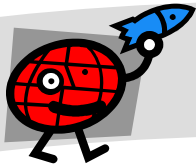
Requirement Phase

ความต้องการของระบบ

การเขียนข้อกำหนดความต้องการด้านระบบ (System Requirement Specification) ควรใช้ภาษาธรรมชาติหรือภาษาที่เข้าใจง่ายเช่นเดียวกับข้อกำหนดความต้องการของผู้ใช้ (User Requirement Specification) แต่อย่างไรก็ตามความต้องการของระบบนั้นมีรายละเอียดทางเทคนิคมากกว่า ภาษาที่ใช้อาจกำกวม ทำให้เข้าใจผิดได้ จึงมีการกำหนดมาตรฐานในการใช้ภาษาธรรมชาติมาอธิบายดังนี้

1. Form-Based Specification
2. Tubular Specification
3. Graphic Model





Requirement Phase

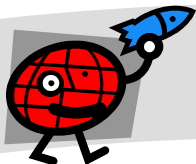
ความต้องการของระบบ

1. Form-Based Specification

แหล่งเอกสาร	
Function	
Description	
Inputs	
Outputs	
Source	
Destination	
Action	
Pre-Condition	
Post-Condition	

จัดวางรายละเอียดอย่างเป็นระเบียบ
แต่ยังไม่สามารถลดความกำกวมได้
อย่างสมบูรณ์ เนื่องจากความต้องการ
หากเป็นการคำนวณค่าต่าง ๆ จะทำ
ให้อ่านยาก





Requirement Phase

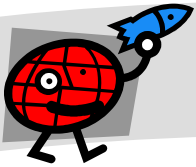
ความต้องการของระบบ

2. Tubular Specification

กรณีที่ต้องการคำนวณค่าต่าง ๆ อาจอธิบายในรูปของตารางเพิ่มเติมจาก Form-Based ได้ จะทำให้เข้าใจง่ายขึ้น นอกจากนี้ ตารางยังสามารถใช้แสดงการตัดสินใจและทางเลือกต่าง ๆ ได้อีกด้วย

เงื่อนไข	การกระทำ
Member = Yes	Discount = 5% Then Net Price = Total Price - (Total Price * Discount)
Member = No	Discount = 0% Then Net Price = Total Price





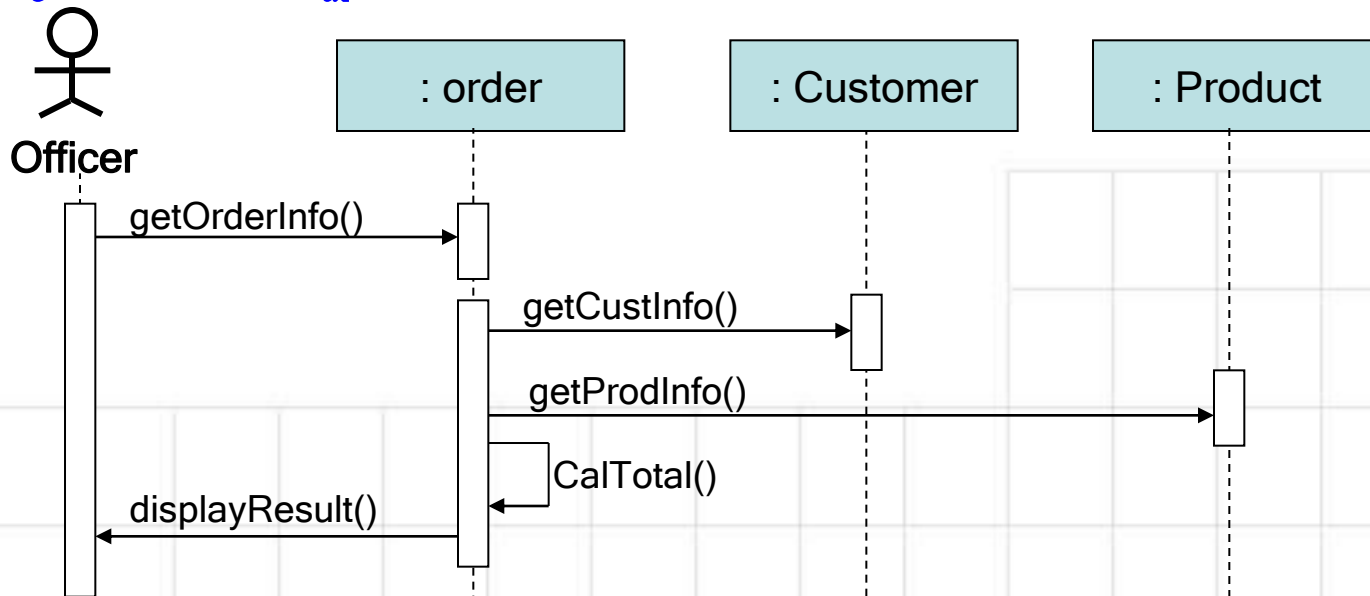
Requirement Phase

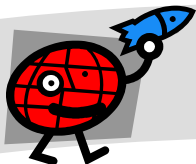
ความต้องการของระบบ

3. Graphic Model

เป็นการนำแผนภาพหรือแบบจำลองมาใช้อธิบายข้อมูลความต้องการเพิ่มเติม เนื่องจากรูปภาพลดความกำกวมหรือแทนคำอธิบายได้มากกว่า ช่วยให้เข้าใจมากยิ่งขึ้น แต่อย่างไรก็ตามการใช้แผนภาพจำเป็นต้องมีความรู้และความเข้าใจในสัญลักษณ์ที่ปรากฏในแผนภาพดังกล่าวด้วย

Ex. Sequence Diagram





Requirement Phase

ความต้องการของซอฟต์แวร์

ความต้องการของซอฟต์แวร์ (Software Requirement Specification) พัฒนาให้ทราบว่าต้องการพัฒนาอะไรบ้าง
System Requirement

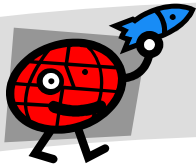
ข้อกำหนดความต้องการของซอฟต์แวร์ (Software Requirement Specification)

1. บทนำ (Introduction)
เกริ่นที่มาของการพัฒนาระบบเบื้องต้น
 - 1.1 ปัญหาที่พบ (Problem Statement)
 - 1.2 บุคลากรที่เกี่ยวข้องกับระบบ (System Users)
 - 1.3 การปฏิบัติ (Operational Setting)
(อธิบายลำดับขั้นตอนการปฏิบัติ)
 - 1.4 การวิเคราะห์ผลกระทบ (Impact Analysis)
 - 1.5 ระบบที่เกี่ยวข้อง (Related Systems)

ระบบอื่นที่เกี่ยวข้องกับระบบคอมพิวเตอร์ที่ใช้งานอยู่ โดยความเกี่ยวพันนี้อาจเป็นฟังก์ชันหนึ่งที่กำหนดให้มีขึ้น ซึ่งโดยส่วนใหญ่ระบบในเชิงพาณิชย์ หรือระบบสำคัญควรมี และสามารถชี้แจงให้เห็นถึงคำตอบหัวข้อต่อไปนี้:

- อะไรคือข้อดีในการเกี่ยวข้องกับระบบอื่น i.e., คุณลักษณะที่ควรนำเสนอในระบบ
- อะไรคือข้อเสียในการเกี่ยวข้องกับระบบอื่น i.e., คุณลักษณะที่ไม่ควรนำมาเสนอในระบบ แต่หาสิ่งที่แตกต่างกันหรือหาหนทางอื่นที่ดีกว่าแทน
- อะไรคือข้อผิดพลาดหากเกี่ยวข้องกับระบบอื่น i.e., คุณลักษณะใหม่ที่ควรนำเสนอในระบบเมื่อไม่พบในระบบอื่นที่เกี่ยวข้อง
- คุณลักษณะสำคัญอื่น ๆ





Requirement

ความต้องการของซอฟต์แวร์

ข้อกำหนดความต้องการของซอฟต์แวร์ (Software Requirements)

1. บทนำ (Introduction)
2. ข้อกำหนดความต้องการ (Specification Requirements)
 - 2.1 Functional Requirement
 - 2.1.1 User Interface Overview
 - 2.1.2 System Specific Overview
 - 2.2 Non-Functional Requirement
 - 2.2.1 System-Related Non-Functional Requirement
 - 2.2.2 Process-Related Non-Functional Requirement
 - 2.2.3 Personnel-Related Non-Functional Requirement
3. การวิเคราะห์และการออกแบบระบบ (System Analysis and Design)
4. ภาคผนวก (Appendices)
5. ดัชนี (Index)

Non-functional system require-

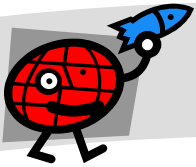
Non-functional process requirements ประกอบด้วย:

- a) เวลาในการพัฒนา
- b) ค่าใช้จ่ายในการพัฒนา
- c) ข้อจำกัดของ SDLC
- d) การส่งมอบระบบ
 - extent of deliverables
 - deliverable formats
- e) การติดตั้งระบบ
 - developer access to

Non-functional personnel requirements ประกอบด้วย :

- a) สำหรับผู้พัฒนา:
 - หนังสือรับรอง/การการันตีบุคคล
 - ประกาศนียบัตร/ลิขสิทธิ์การพัฒนา
 - แอปพลิเคชัน
- b) สำหรับผู้ใช้:
 - ระดับทักษะการใช้งาน
 - การเข้าถึงระดับพิเศษ
 - การฝึกอบรม

- simplicity versus power .



Requirement Engineer

การจัดทำข้อกำหนดความต้องการ ต้องมีคุณสมบัติสำคัญ คือ

“สามารถตรวจสอบ พิสูจน์ และวิเคราะห์คุณภาพได้”

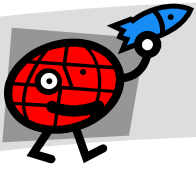
ดังนั้นจึงต้องมีกระบวนการบางอย่างที่ทำให้กำหนดความต้องการถูกต้องและตรงกับที่ต้องการอย่างแท้จริง เรียกว่า **วิศวกรรมความต้องการ**

**“กระบวนการที่ทำให้วิศวกรซอฟต์แวร์
เข้าใจและเข้าถึงความต้องการของลูกค้าได้อย่างแท้จริง”**

เป้าหมายของวิศวกรรมซอฟต์แวร์

คือ การสร้างและบำรุงรักษาเอกสารข้อกำหนดความต้องการ ทั้งในด้านระบบ และซอฟต์แวร์ ให้เป็นเอกสารที่มีคุณภาพมากที่สุด





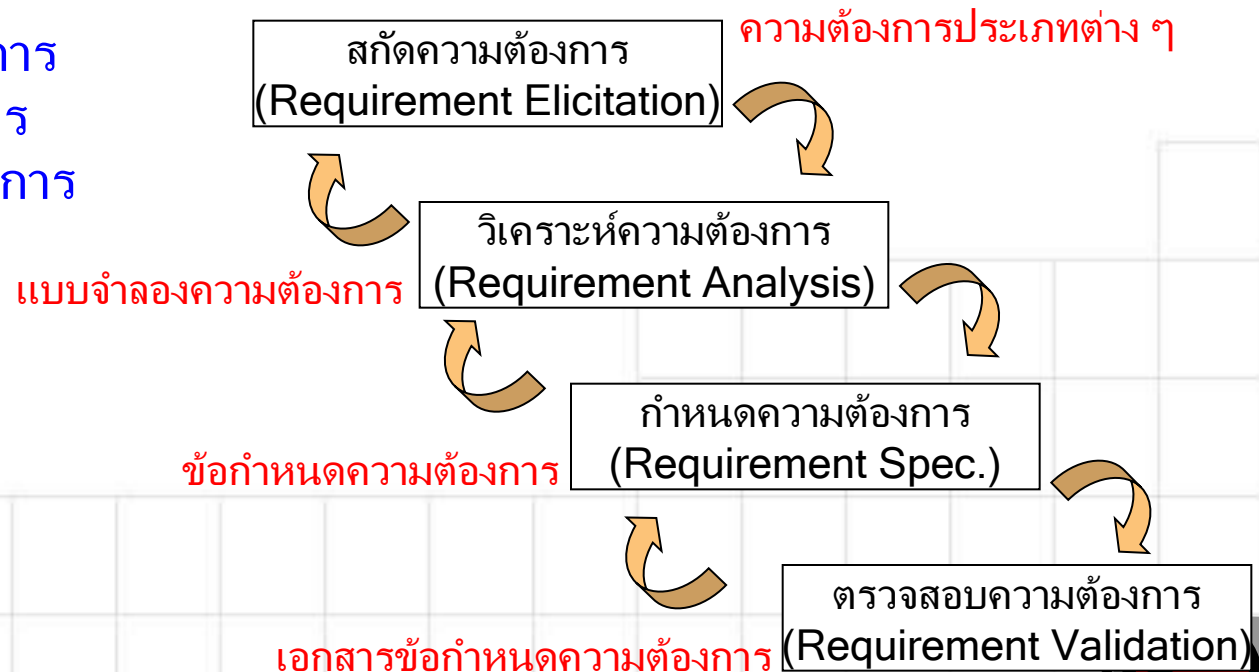
Requirement Engineer

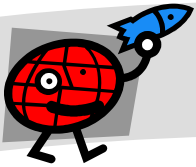
กระบวนการวิศวกรรมความต้องการ

กิจกรรมของวิศวกรรมความต้องการ

อยู่ระยะการวิเคราะห์ความต้องการของกระบวนการผลิตซอฟต์แวร์ ต้องดำเนินงานอย่างเป็นขั้นตอน มีกระบวนการและทีมงานเฉพาะ ประกอบไปด้วยกิจกรรมย่อยดังนี้

1. สกัดความต้องการ
2. วิเคราะห์ความต้องการ
3. กำหนดความต้องการ
4. ตรวจสอบความต้องการ





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

1. สกัต์ความต้องการ (Requirement Elicitation)

สกัต์ความต้องการ คือ การรวบรวมหรือค้นหาความต้องการ เป็นขั้นตอนการทำความเข้าใจปัญหาที่เกิดขึ้นที่ต้องการแก้ไขด้วยซอฟต์แวร์ (ความจำเป็นของการนำซอฟต์แวร์มาใช้) โดยเริ่มต้นจากกำหนดกลุ่มบุคคลที่เกี่ยวข้องซึ่งเป็นแหล่งที่มาของความต้องการ จากนั้นรวบรวมความต้องการด้วยเทคนิคต่าง ๆ

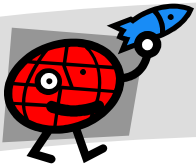
- สัมภาษณ์ (Interview)

เป็นวิธีดั้งเดิมและนิยมใช้มากที่สุด วิศวกรซอฟต์แวร์ควรทราบถึงข้อดี-ข้อเสียของวิธีการสัมภาษณ์ รวมทั้งวิธีการโน้มน้าวผู้ถูกสัมภาษณ์ด้วย

- การแสดงลำดับเหตุการณ์ (Scenario)

เป็นการเตรียมคำถามตามลำดับงานของผู้ใช้ ในแต่ละงานมีการตั้งคำถามว่า “จะเกิดอะไรขึ้น ถ้า...” และ “จะต้องทำอะไร” การรวบรวมความต้องการด้วยเทคนิคนี้จะเชื่อมโยงไปยังการสร้างแบบจำลอง Use Case ได้ง่าย และ Use Case เป็นลักษณะหนึ่งของ Scenario ผู้ใช้จึงเข้าใจคำถามได้ง่าย ทำให้การตอบคำถามได้ชัดเจนขึ้น





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

1. สกัความต้องการ (Requirement Elicitation) ต่อ

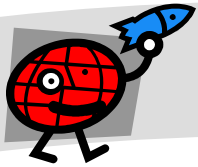
- ต้นแบบ (Prototype)

เป็นเทคนิคที่ทำให้ผู้เข้าใจในสถานการณ์และคำถามได้ง่ายเช่นกัน การทำต้นแบบมีหลายชนิด เช่น การออกแบบจอภาพบนกระดาษ เพื่อทดสอบการยอมรับความต้องการในเบื้องต้น ซึ่งอาจเป็นเทคนิคซ้ำกับการตรวจสอบความต้องการ

- การประชุม (Facilitated Meeting)

เป็นการเรียกกลุ่มบุคคลที่เกี่ยวข้องเข้าร่วมประชุมเพื่อขอความคิดเห็นและความต้องการ เพื่อให้เกิดความเข้าใจในความต้องการอย่างถ่องแท้มากกว่าการทำงานเพียงลำพัง เป็นการระดมความคิดเพื่อแก้ไขปัญหาเมื่อพบข้อขัดแย้งในความต้องการ อย่างไรก็ตาม ต้องระมัดระวังเรื่องความขัดแย้งระหว่างกลุ่มบุคคลที่อาจเกิดขึ้นได้ในทุกองค์กร





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

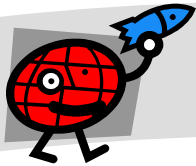
2. วิเคราะห์ความต้องการ (Requirement Analysis)

ประเมินความต้องการที่รวบรวมมาได้ เพื่อจัดกลุ่มความต้องการ จัดลำดับความสำคัญ แก้ไขความขัดแย้งระหว่างความต้องการเพื่อให้เกิดความสอดคล้องกัน จากนั้นสร้างแบบจำลองที่เป็น Conceptual Model และนำเสนอผู้เกี่ยวข้องทั้งหมดให้ยอมรับในความต้องการที่ได้ หากไม่ยอมรับต้องแก้ไข เปรียบเทียบและนำเสนอจนกว่าจะได้รับการยอมรับในที่สุด

3. กำหนดความต้องการ (Requirement Specification)

หลังจากแบบจำลองความต้องการได้รับการยอมรับแล้ว จะนำมาจัดทำเอกสารความต้องการ โดยเริ่มจากการนิยามความต้องการ และจัดทำเป็นข้อกำหนดของระบบ เพื่อแจกแจงเป็นข้อกำหนดความต้องการของซอฟต์แวร์ โดยเอกสารทั้งหมด ต้องตรวจสอบและวัดคุณภาพได้





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

4. ตรวจสอบความต้องการ (Requirement Validation)

เป็นการทบทวนและตรวจสอบข้อกำหนดความต้องการในเอกสารทั้งหมด เพื่อให้เกิดความเที่ยงตรง สอดคล้อง ครบถ้วนสมบูรณ์ มีความเป็นไปได้ และสามารถพิสูจน์ได้ตามเป้าหมายของกระบวนการวิศวกรรมซอฟต์แวร์ จากนั้นจะนำไปทดสอบเพื่อให้เกิดการยอมรับจากบุคคลทุกฝ่ายที่เกี่ยวข้อง

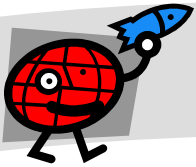
- ความเที่ยงตรง (Validity)

นอกจากฟังก์ชันพื้นฐานที่ผู้ใช้ส่วนใหญ่ต้องการแล้ว อาจมีฟังก์ชันบางอย่างที่ผู้ใช้กลุ่มอื่นต้องการ ดังนั้นทีมงานควรให้ความสำคัญ เนื่องจากผู้ใช้แต่ละกลุ่มมีความต้องการแตกต่างกัน ดังนั้นการกำหนดความต้องการจะตอบสนองกับผู้ใช้ทุกกลุ่มอย่างเท่าเทียมกัน

- ความสอดคล้อง (Consistency)

ความต้องการในเอกสารจะต้องไม่มีการทับซ้อนกัน ไม่มีความขัดแย้งกัน หากพบความต้องการไม่เป็นแนวทางเดียวกัน ต้องตรวจสอบและแก้ไขให้สอดคล้องกันทันที





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

4. ตรวจสอบความต้องการ (Requirement Validation)

- **ความครบถ้วนสมบูรณ์ (Completeness)**

เอกสารต้องระบุนายละเอียดฟังก์ชันและการบริการอย่างครบถ้วน และครอบคลุมความต้องการของผู้ใช้ จะต้องไม่มีฟังก์ชันใดที่ผู้ใช้ต้องการขาดหายไป

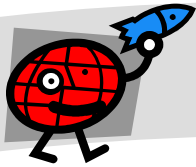
- **ความเป็นไปได้ (Feasibility)**

เป็นการตรวจสอบความต้องการด้วยองค์ความรู้เกี่ยวกับเทคโนโลยีที่องค์กรมีอยู่ เพื่อให้มั่นใจว่าสามารถนำความต้องการที่ระบุในเอกสารไปพัฒนาระบบได้จริง ซึ่งนอกจากความเป็นไปได้ทางเทคโนโลยีแล้ว ยังต้องพิจารณาถึงความเป็นไปได้ด้านงบประมาณและระยะเวลาในการพัฒนาด้วย หากใช้งบประมาณหรือเวลาในการพัฒนามากกว่าที่มีอยู่ ถือว่าความเป็นไปได้น้อยมาก

- **สามารถพิสูจน์ได้ (Verifiability)**

ความต้องการนั้นต้องพิสูจน์หาความจริงได้ คือ ต้องทดสอบและทดลองลูกค้าเห็นถึงการทำงานจริงของระบบตามที่ระบุไว้ในเอกสารได้





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

4. ตรวจสอบความต้องการ (Requirement Validation)

★ เทคนิคของการตรวจสอบความต้องการ

- การทบทวนความต้องการ (Requirement Review)

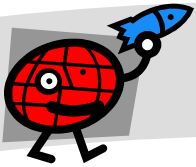
เป็นการตรวจสอบเอกสารความต้องการอย่างละเอียด เพื่อตรวจหาความต้องการหรือข้อสมมติฐานที่ผิดพลาด ถูกละเอียด ไม่ชัดเจน และไม่ตรงกับมาตรฐานที่กำหนด

(1) ทบทวนแบบไม่เป็นทางการ (Informal Review) ทีมทบทวนจะนำเอกสารความต้องการมาพิจารณาหาข้อผิดพลาดร่วมกับผู้มีส่วนได้ส่วนเสียและผู้รับเหมาช่วง

(2) ทบทวนแบบเป็นทางการ (Formal Review) ทีมทบทวนจะต้องพิจารณาความต้องการร่วมกับผู้ใช้ที่ละรายการ เพื่อตรวจสอบความสอดคล้องและความครบถ้วนสมบูรณ์ตามลักษณะดังต่อไปนี้

- สามารถพิสูจน์ได้
- สามารถเข้าใจได้
- สามารถย้อนกลับไปตรวจสอบได้
- สามารถดัดแปลงได้ (โดยไม่ส่งผลกระทบต่อระบบ)





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

4. ตรวจสอบความต้องการ (Requirement Validation)

เทคนิคของการตรวจสอบความต้องการ

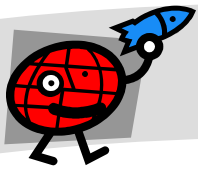
- การจัดทำต้นแบบ (Prototyping)

เป็นการสร้างต้นแบบของระบบ (Executable Model) เพื่อสาธิตให้ลูกค้าหรือผู้ใช้ระบบดู หรือทดลองใช้ด้วยตนเอง นอกจากนี้ยังเป็นวิธีที่รวบรวมความต้องการที่เกิดขึ้นใหม่ได้ด้วย อย่างไรก็ตาม การสร้างต้นแบบต้องใช้เงินทุนสูง แต่เมื่อเปรียบเทียบกับผลที่ได้รับนับว่าคุ้มค่ามาก จัดเป็นวิธีที่ดีที่สุดอย่างหนึ่ง

- การสร้างแบบทดสอบ (Test-Case Generation)

ความต้องการที่ดีต้องสามารถทดสอบได้ และถ้าทดสอบนั้นทำได้ยากหรือออกแบบยาก แสดงว่าการนำความต้องการดังกล่าวไปพัฒนาจากตามไปด้วยจึงความนำความต้องการนั้นกลับไปพิจารณาใหม่





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

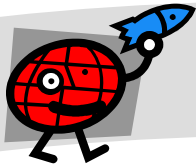
การจัดการความต้องการ (Requirement Management) จัดเป็นส่วนหนึ่งของกระบวนการวิศวกรรมความต้องการ

“ กระบวนการทำความเข้าใจและควบคุมการเปลี่ยนแปลงความต้องการของระบบ ”

โดยสามารถเริ่มดำเนินการได้ทันทีที่จัดทำเอกสารข้อกำหนดความต้องการฉบับร่างเสร็จเรียบร้อยแล้ว (ขั้นที่ 4) แต่การวางแผนการจัดการความต้องการนั้น ควรเริ่มตั้งแต่ ขั้นตอนการสกัดความต้องการเริ่มต้น (ขั้นที่ 1) สาเหตุของการเปลี่ยนแปลง :

1. ผู้ใช้มีหลายกลุ่ม ความต้องการแตกต่างกันออกแบบ อาจเกิดการขัดแย้ง จึงหลีกเลี่ยงไม่ได้ที่จะต้องปรับความสมดุลของความต้องการใหม่
2. โดยทั่วไป ผู้ใช้ซึ่งเป็นผู้จ่ายเงินลงทุน กับผู้ใช้ระบบโดยตรงไม่ใช่ผู้ใช้กลุ่มเดียวกัน
3. ภายหลังการติดตั้งระบบเพื่อใช้งาน สภาพแวดล้อมทางธุรกิจ และเทคโนโลยีที่เปลี่ยนแปลงไป มีผลทำให้ระบบต้องเปลี่ยนแปลงตามไปด้วย





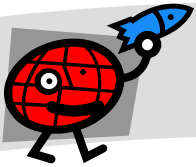
Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

มุมมองของวิวัฒนาการของความต้องการ แบ่งความต้องการออกเป็น 2 ประเภท ได้แก่

1. **ความต้องการที่ไม่เปลี่ยนแปลง (Enduring Requirement)**
เป็นความต้องการแบบคงที่ ไม่เปลี่ยนแปลงได้ง่าย เป็นความต้องการที่เกิดจากการทำงานหลักของธุรกิจในแต่ละวัน เช่น ระบบลงทะเบียน วิชาเรียน ค่าลงทะเบียน เป็นต้น
2. **ความต้องการที่เปลี่ยนแปลง (Volatile Requirement)**
เป็นความต้องการที่เปลี่ยนแปลงอยู่เสมอในระหว่างการพัฒนา
ระบบหรือหลังจากการติดตั้งระบบ เช่น นโยบายการลงทะเบียน เป็นต้น





Requirement Engineer

กระบวนการวิศวกรรมความต้องการ

การจัดการกับการเปลี่ยนแปลงความต้องการ

เมื่อมีการยื่นข้อเสนอให้มีการเปลี่ยนแปลงความต้องการใด ๆ เกิดขึ้น ทีมงานหรือองค์กรต้องมีกระบวนการจัดการกับการเปลี่ยนแปลง (Requirement Change Management) ดังกล่าว เพื่อให้การเปลี่ยนแปลงที่เกิดขึ้นมีความสอดคล้องกับส่วนอื่นที่สัมพันธ์กัน และอยู่ภายใต้การควบคุมอย่างเป็นทางการ

กระบวนการจัดการการแปลงเปลี่ยนแปลงจะเกี่ยวข้องกับการวิเคราะห์ถึงความคุ้มค่าเมื่อต้องการเปลี่ยนแปลงตามข้อเสนอ หากมีความคุ้มค่าจะอนุมัติให้ดำเนินการเปลี่ยนแปลงได้ แต่หากพบว่าไม่ได้ผลตอบแทนที่คุ้มค่าก็จะยกเลิกข้อนั้นไป

