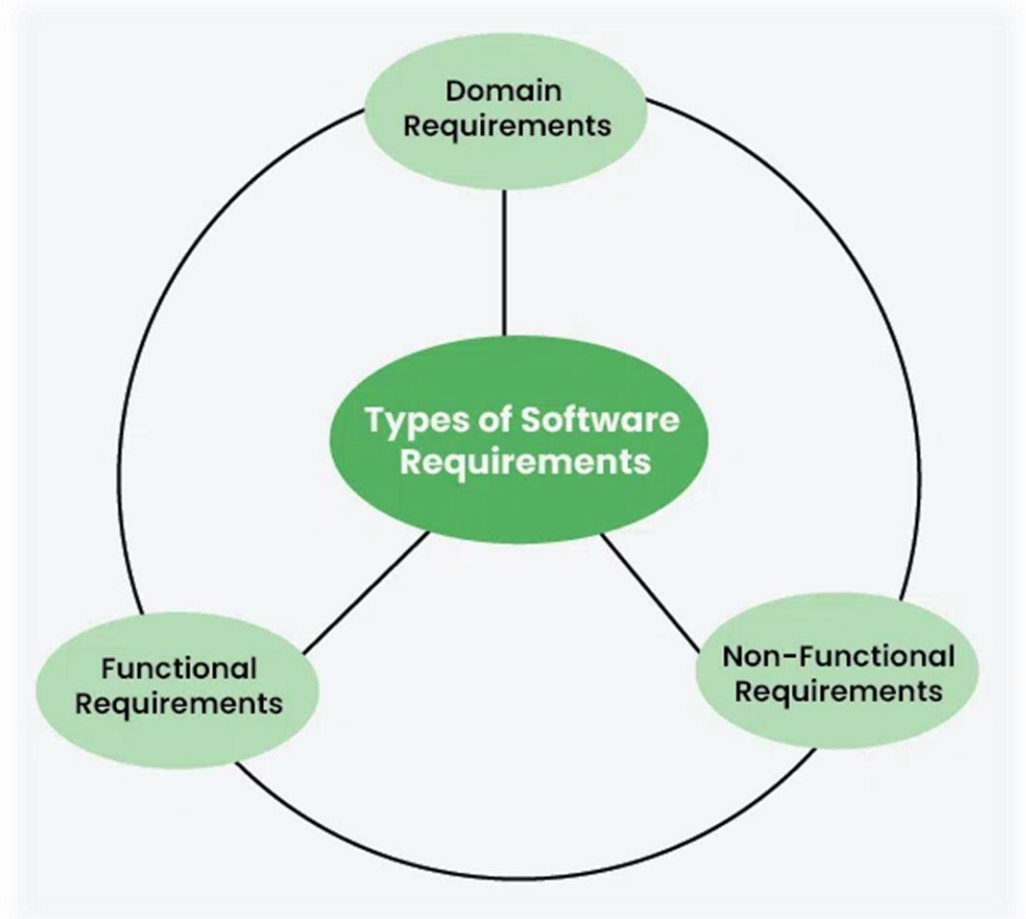


# Chapter 9 : Requirement



ASSIT.PROF. JUTHAWUT CHANTHARAMALEE

CURRICULUM OF COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY, SUAN DUSIT UNIVERSITY

# Outline of this presentation

---

1. ความหมายของวิศวกรรมความต้องการ

2. กระบวนการวิศวกรรมความต้องการ

3. การสกัดความต้องการ

4. การวิเคราะห์ความต้องการ

5. การกำหนดความต้องการ

6. การจัดการความต้องการ

# 1. ความหมายของวิศวกรรมความต้องการ

---

1. ข้อมูลความต้องการ เป็นวัตถุดิบสำคัญในการออกแบบ
2. หากสิ่งื่อนำมาใช้ในการออกแบบเป็นสิ่งที่ผิดพลาด ย่อมส่งผลให้การออกแบบผิดพลาดไปด้วย
3. ปัญหาที่เกิดขึ้นกับซอฟต์แวร์ส่วนใหญ่
  - มีผลมาจากการออกแบบที่ด้อยคุณภาพ
  - การกำหนดความต้องการไม่ถูกต้อง
4. ในอุตสาหกรรมการผลิตซอฟต์แวร์ นำหลักวิชาวิศวกรรมมาใช้ในการวิเคราะห์ความต้องการ เรียกว่า “วิศวกรรมความต้องการ (Requirement Engineering)”
5. มีวัตถุประสงค์เพื่อให้วิศวกรซอฟต์แวร์มีความเข้าใจและเข้าถึงความต้องการของผู้ใช้ได้อย่างแท้จริง

- 
1. มีการจัดทำข้อกำหนดความต้องการ ต้องมีคุณสมบัติสำคัญ คือ ความสามารถตรวจสอบ พิสูจน์ และวิเคราะห์คุณภาพได้
  2. มีการจัดทำข้อกำหนดความต้องการเป็นเครื่องบ่งชี้ว่าความต้องการที่กำหนดขึ้นจะต้องชัดเจน ไม่คลุมเครือ
  3. มีกระบวนการบางอย่าง เพื่อให้วิศวกรซอฟต์แวร์สามารถกำหนดความต้องการได้อย่างถูกต้อง และตรงกับความต้องการที่แท้จริง
  4. กระบวนการดังกล่าว คือ วิศวกรรมความต้องการ

---

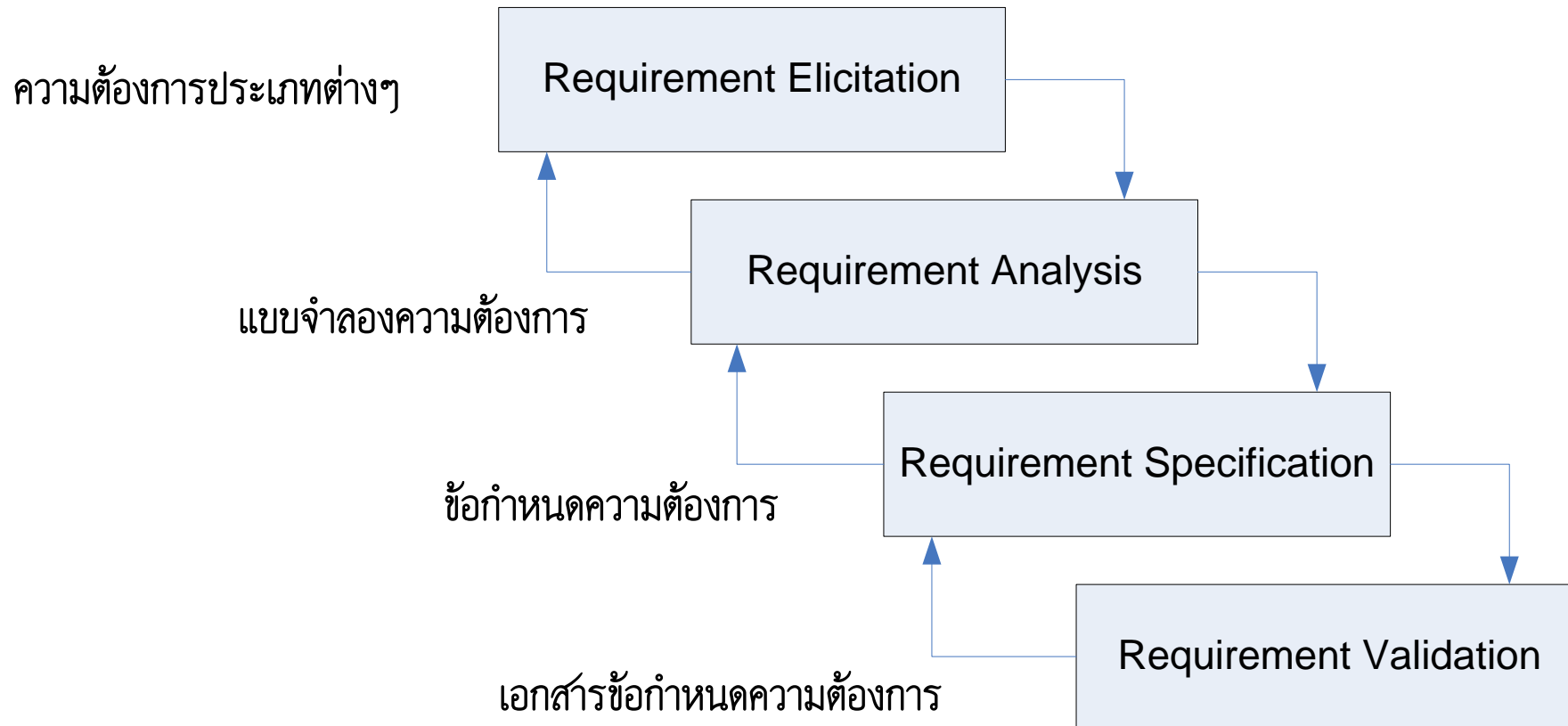
กระบวนการที่จะทำให้วิศวกรรมซอฟต์แวร์ เข้าใจและเข้าถึงความต้องการของลูกค้าได้อย่างแท้จริง ด้วยการสกัดความต้องการ ตรวจสอบ และนิยามความต้องการ เพื่อนำไปสร้างเป็นข้อกำหนดความต้องการด้านระบบหรือซอฟต์แวร์ ที่จะใช้เป็นจุดเริ่มต้นในการพัฒนาระบบในขั้นตอนต่อไป [Jawadekar, 2004]

นอกจากนี้ วิศวกรรมความต้องการยังรวมไปถึงกระบวนการควบคุมการเปลี่ยนแปลงของความต้องการที่จะเกิดขึ้นด้วย เรียกว่า “การจัดการความต้องการ (Requirement Management)” ดังนั้น การวิศวกรรมความต้องการจึงช่วยให้ซอฟต์แวร์ที่ผลิตออกมา สามารถแก้ไขปัญหาหรือช่วยสนับสนุนการทำงานของลูกค้านได้อย่างถูกต้องตรงตามความต้องการที่แท้จริง โดยเป้าหมายของวิศวกรรมความต้องการ ก็คือ การสร้างและบำรุงเอกสารข้อกำหนดความต้องการ ทั้งทางด้านระบบและด้านซอฟต์แวร์ ให้เป็นเอกสารที่มีคุณภาพที่สุด

- 
1. กิจกรรมของวิศวกรรมความต้องการ จะรวมอยู่ในระยะการวิเคราะห์ความต้องการของกระบวนการผลิตซอฟต์แวร์
  2. เป็นกิจกรรมที่ต้องดำเนินการอย่างเป็นลำดับขั้นตอน มีกระบวนการและทีมงานเฉพาะ
  3. มีลักษณะการทำซ้ำในแต่ละระยะของการผลิตซอฟต์แวร์ เพื่อให้เป็นเอกสารความต้องการที่มีประสิทธิภาพปัจจุบัน
  4. แบบจำลองของกระบวนการวิศวกรรมความต้องการมีหลายรูปแบบไม่ว่าจะเป็น Waterfall หรือ Spiral ตามการประยุกต์ใช้ของแต่ละองค์กร จึงทำให้ขั้นตอนของกระบวนการมีจำนวนแตกต่างกัน

## 2. กระบวนการวิศวกรรมความต้องการ

---



---

## กระบวนการที่จะเข้าถึงความต้องการที่แท้จริง

- สกัดความต้องการ
- วิเคราะห์ความต้องการ
- กำหนดหรือนิยามความต้องการ
- ตรวจสอบความต้องการ
- จัดการความต้องการ

**เป้าหมาย** คือ การสร้างและบำรุงเอกสารข้อกำหนดความต้องการทั้งด้านระบบและด้านซอฟต์แวร์ ให้เป็นเอกสารที่มีคุณภาพ




---

## สกัดความต้องการ (Requirement Elicitation)

- รวบรวมความต้องการและเข้าใจปัญหาของระบบงาน
- ความจำเป็นของการนำซอฟต์แวร์มาใช้งาน
- ความต้องการจากกลุ่มผู้ใช้

## วิเคราะห์ความต้องการ (Requirement Analysis)

- ศึกษาและประเมินความต้องการที่รวบรวมได้
  - จัดลำดับความสำคัญของความต้องการ
  - สร้างแบบจำลองระดับแนวคิด (Conceptual Model)
  - ประชุมและนำเสนอ
- 

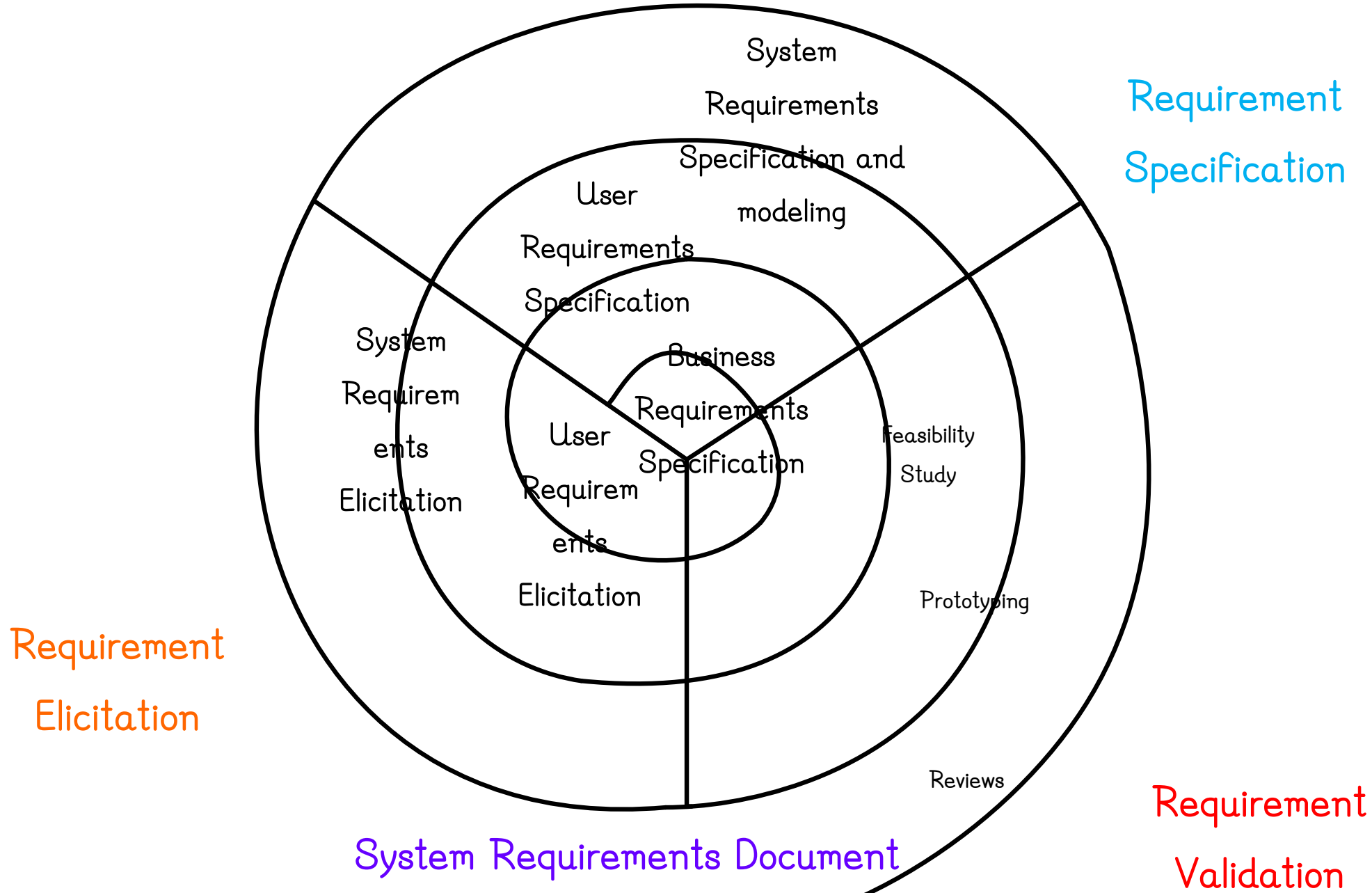
---

## กำหนดความต้องการ (Requirement Specification)

- นิยามความต้องการของระบบ
- กำหนดความต้องการด้านระบบ
- แจกแจงในรูปแบบเอกสาร

## ตรวจสอบความต้องการ (Requirement Validation)

- ทบทวนเอกสารทั้งหมด
- ตรวจสอบความสมบูรณ์
- ตรงตามเป้าหมายของการพัฒนาซอฟต์แวร์



### 3. การสกัดความต้องการ (Requirement Elicitation)

---

- เป็นขั้นตอนในการเก็บรวบรวมข้อเท็จจริง เพื่อทำความเข้าใจกับปัญหาที่เกิดขึ้น และบทบาทของซอฟต์แวร์ในการทำหน้าที่แก้ปัญหา
- วิศวกรซอฟต์แวร์จะต้องสามารถเข้าใจถึงเป้าหมายและวัตถุประสงค์ของลูกค้าได้เป็นอย่างดี
- ค้นหาความต้องการจากบุคคลที่เกี่ยวข้องแต่ละกลุ่มด้วยเทคนิคต่าง ๆ เช่น การสัมภาษณ์ การออกแบบสอบถาม
- ทักษะพื้นฐานที่วิศวกรซอฟต์แวร์ต้องมีในขั้นตอนนี้ คือ “การติดต่อสื่อสารหรือการประสานงาน (Communication)”

---

## การค้นหาความต้องการที่แท้จริงเป็นเรื่องยาก

1. ความต้องการของผู้ใช้ค่อนข้างคลุมเครือ ไม่ชัดเจน มีลักษณะเป็นนามธรรม และมีความเป็นไปได้น้อย
2. วิศวกรซอฟต์แวร์ต้องทำความเข้าใจกับคำศัพท์ของผู้ใช้ ที่ใช้บอกความต้องการ
3. ผู้ใช้แต่ละคนมีความต้องการแตกต่างกัน
4. สายบังคับบัญชาขององค์กรลูกค้า อาจส่งผลกระทบต่อความต้องการที่รวบรวมมาได้
5. สภาพแวดล้อมทางธุรกิจและสภาพเศรษฐกิจมีผลต่อการเปลี่ยนแปลงความต้องการ

---

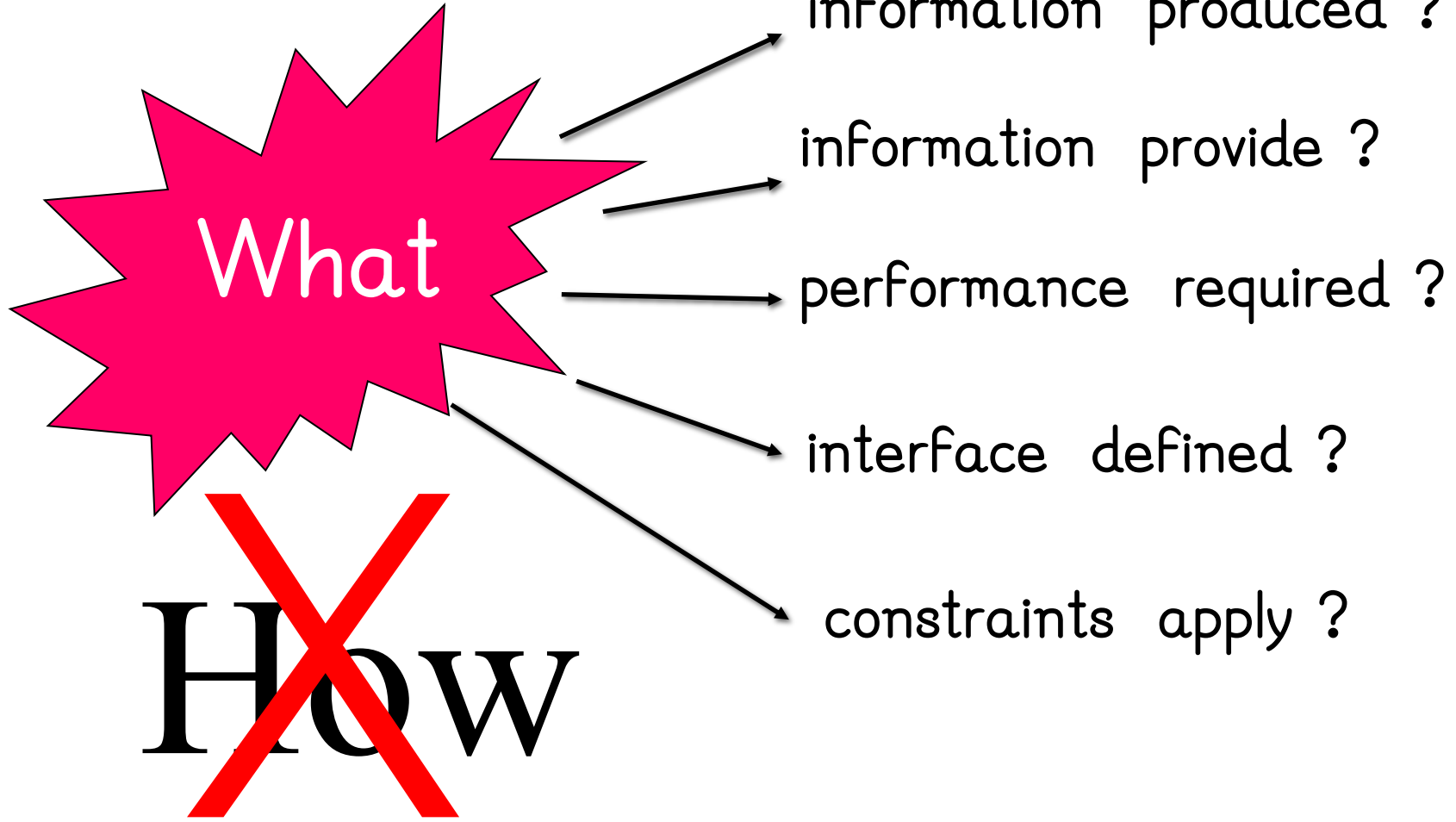
## เทคนิคการเก็บรวบรวมความต้องการ

1. การสัมภาษณ์ (Interview) นิยมใช้มากที่สุด
2. การแสดงลำดับเหตุการณ์ (Scenario) เตรียมคำถามตามลำดับงานของผู้ใช้
3. สร้างต้นแบบ (Prototype) เช่น ออกแบบจอภาพบนกระดาษ เพื่อทดสอบการยอมรับความต้องการในเบื้องต้น
4. การประชุม (Meeting) เป็นการเรียกกลุ่มบุคคลที่เกี่ยวข้องมาประชุม เพื่อขอความคิดเห็นและความต้องการ
5. การสังเกต (Observation) โดยตรวจสอบสภาพแวดล้อมการทำงานของผู้ใช้ เป็นวิธีที่ดีแต่ค่าใช้จ่ายสูง

## กลยุทธ์ที่นิยมใช้ในการวิเคราะห์ความต้องการ

1. Asking
2. Derivation from Existing System
3. Process Analysis หรือ Decision Analysis
4. Prototyping

# 1. Asking



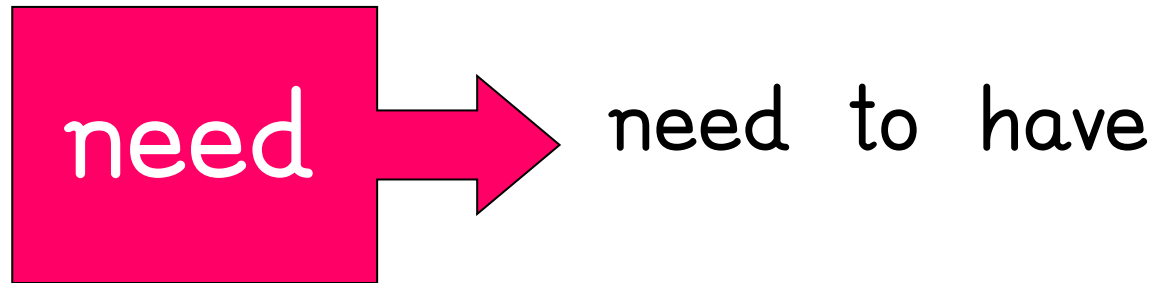


## Interview

---

1. กำหนดบุคคลที่จะไปสัมภาษณ์ (Determine who to Interview)
2. กำหนดวัตถุประสงค์ (Establish Objective for the Interview) มีหน้าที่อะไรบ้าง หน่วยงานนี้มีความต้องการอะไรบ้าง อยากให้ช่วยเหลือหรือปรับปรุงอะไร มีขั้นตอนการทำงานอะไรบ้าง
3. เตรียมการสัมภาษณ์ (Prepare for the Interview) ปลายเปิด/ปลายปิด
4. ทำการสัมภาษณ์ (Conduct the Interview) ตรวจสอบ การบันทึกวิดีโอ/เทป
5. ทำเอกสารหลังการสัมภาษณ์ (Document the Interview) สรุปผลการสัมภาษณ์

## User Requirements



“I know you believe you understand what you think I said , but I am not sure you realize that what you heard is not what I meant..”

## 4. การวิเคราะห์ความต้องการ (Requirement Analysis)

---

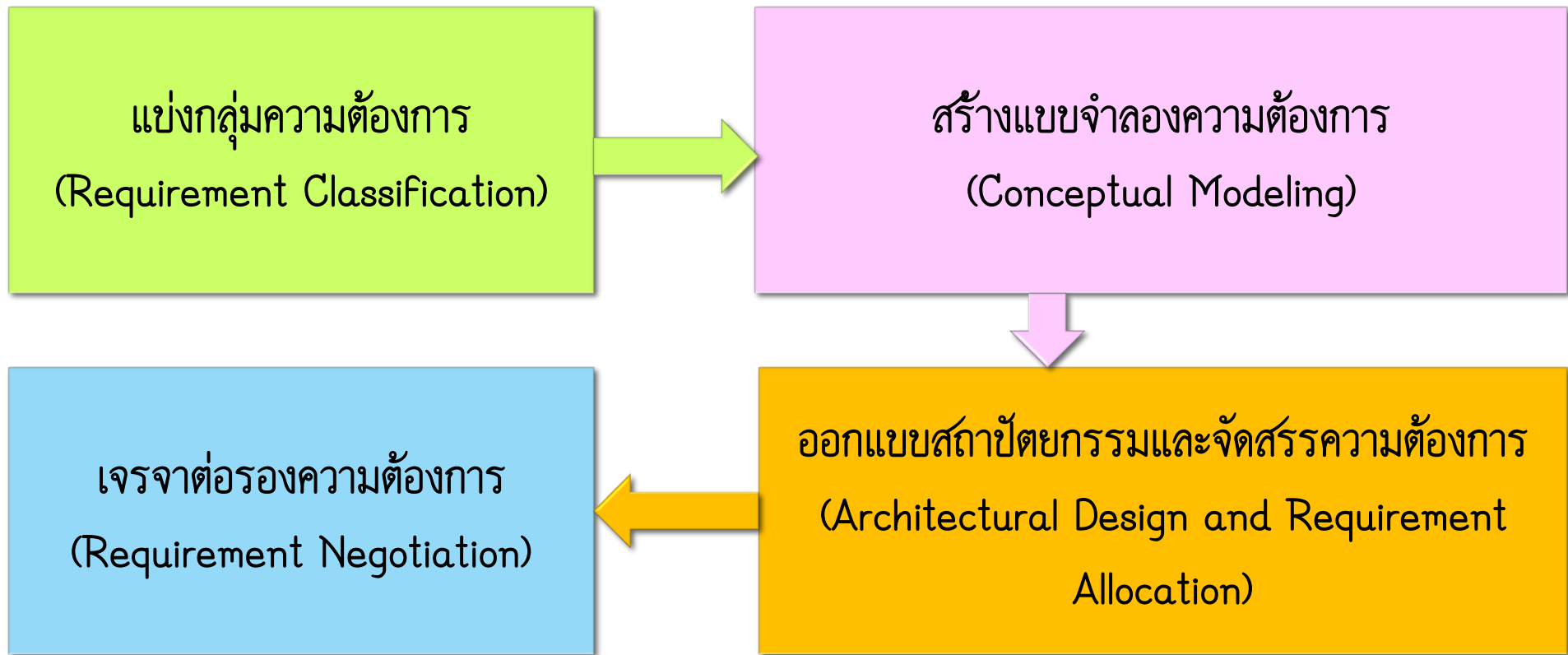
1. การนำข้อมูลความต้องการที่รวบรวมได้มาวิเคราะห์หรือประเมิน
2. เพื่อจำแนกกลุ่มของความต้องการ
3. จัดลำดับความสำคัญ ดูความสอดคล้อง ซ้ำซ้อนความขัดแย้ง
4. สร้างแบบจำลอง ออกแบบสถาปัตยกรรมของซอฟต์แวร์
5. นำไปทดสอบการยอมรับจากลูกค้า
6. เมื่อลูกค้ายอมรับในข้อกำหนดความต้องการ คือ เอกสารความต้องการทั้งหมด (ฉบับร่าง)

---

## การวิเคราะห์ความต้องการ มีวัตถุประสงค์ ดังนี้

1. เพื่อตรวจหาและแก้ไขความขัดแย้งระหว่างความต้องการในแต่ละรายการ
2. เพื่อค้นหาขอบเขตของซอฟต์แวร์และการทำงานกับสภาพแวดล้อมนอกระบบ
3. เพื่อศึกษาความต้องการด้านระบบอย่างละเอียด เพื่อใช้ในการกำหนดความต้องการด้านซอฟต์แวร์

## การวิเคราะห์ความต้องการ มีกิจกรรมย่อย ดังรูป



---

## การแบ่งกลุ่มความต้องการ (Requirement Classification)

1. แบ่งเป็นความต้องการที่เป็นหน้าที่หลัก (Functional Requirement) และไม่ใช่หน้าที่หลัก (Non-Functional Requirement)
2. แบ่งความต้องการที่เกี่ยวข้องกับผลิตภัณฑ์ (Product) และกระบวนการ (Process)
3. แบ่งกลุ่มตามขอบเขตของความต้องการ
  - จำเป็น (Mandatory)
  - ปรารถนาสูง (Highly Desirable)
  - ปานกลาง (Desirable)
  - ละเว้นได้ (Optional)

---

4. แบ่งกลุ่มตามขอบเขตของความต้องการ โดยต้องให้ความสำคัญต่อความต้องการที่มีขอบเขตกว้าง

5. แบ่งกลุ่มตามการเปลี่ยนแปลงของความต้องการ

- ความต้องการที่เปลี่ยนแปลงได้ (Volatility)
- ความต้องการที่ไม่สามารถเปลี่ยนแปลงได้ (Stability)

---

## จุดประสงค์การสร้างแบบจำลองความต้องการ (Requirement Model)

1. เพื่อจำลองความต้องการที่รวบรวมมา ทำให้ผู้ที่เกี่ยวข้องเห็นภาพของความต้องการ
2. เข้าใจความต้องการได้ตรงกัน
3. ชี้ถึงจุดผิดพลาดของความต้องการได้ง่าย
4. แก้ไขได้ทันทีก่อนนำไปออกแบบจำลองความต้องการ
5. ถือเป็นกุญแจสำคัญสำหรับการวิเคราะห์ความต้องการและออกแบบระบบ



---

## ชนิดและวิธีการสร้างแบบจำลองจะแตกต่างกัน ตามแนวทางของการวิเคราะห์ระบบ

1. เซิงโครงสร้าง (SSAD) จะใช้ DFD และ ERD เป็นแบบจำลองกระบวนการ การไหลของข้อมูล และโครงสร้างข้อมูล
2. เซิงวัตถุ (OOSAD)
3. Use case เพื่อให้เห็นหน้าที่การทำงานของซอฟต์แวร์
4. Class/Object Diagram แสดงให้เห็นข้อมูลและพฤติกรรมของระบบ

---

## ปัจจัยที่ส่งผลกระทบต่อการใช้แบบจำลอง ดังนี้

1. ธรรมชาติของปัญหา เช่น ความต้องการการทำงานแบบเวลาจริง (Real-time) จะต้องใช้แบบจำลองในลักษณะ Control Flow และ State Model
2. ความชำนาญของวิศวกรซอฟต์แวร์
3. ความต้องการด้านกระบวนการของลูกค้า
4. ระเบียบวิธีปฏิบัติและเครื่องมือที่เลือกใช้ อาจไม่ได้รับการยอมรับจากลูกค้า

---

## ข้อดีของการสร้างแบบจำลองของระบบ

- ทีมงานทราบถึงภาพรวมของระบบ
- ทราบถึงการโต้ตอบกับสภาพแวดล้อมอื่นนอกกระบบ
- ประโยชน์ในการออกแบบการทำงานของซอฟต์แวร์

## สิ่งสำคัญในการสร้างแบบจำลอง คือ

- ระเบียบวิธีปฏิบัติที่ใช้สร้างแบบจำลอง
- UML (Unified Modeling Language)
- Formal Modeling

## การออกแบบสถาปัตยกรรมและการจัดสรรความต้องการ (Architectural Design and Requirement Allocation)

---

1. เพื่อลดความซับซ้อนของกระบวนการวิศวกรรมซอฟต์แวร์ ทำให้ต้องมีการออกแบบสถาปัตยกรรมของซอฟต์แวร์
2. เพื่อแสดงคอมโพเนนต์หรือส่วนประกอบของซอฟต์แวร์ ที่เข้ามาสนับสนุนและรองรับความต้องการส่วนใดของผู้ใช้
3. เพื่อการจัดสรรความต้องการ (Requirement Allocation) เข้ากับองค์ประกอบแต่ละส่วนของซอฟต์แวร์
4. เพื่อการจัดสรรความต้องการมีความสำคัญ
  - ช่วยให้ทีมงานนำแต่ละส่วนที่จัดสรรแล้วไปวิเคราะห์ในระดับรายละเอียดเพิ่มเติมต่อไป
  - ในโครงการขนาดใหญ่ การจัดสรรความต้องการทำให้เกิดการวิเคราะห์ระบทย่อยรอบใหม่

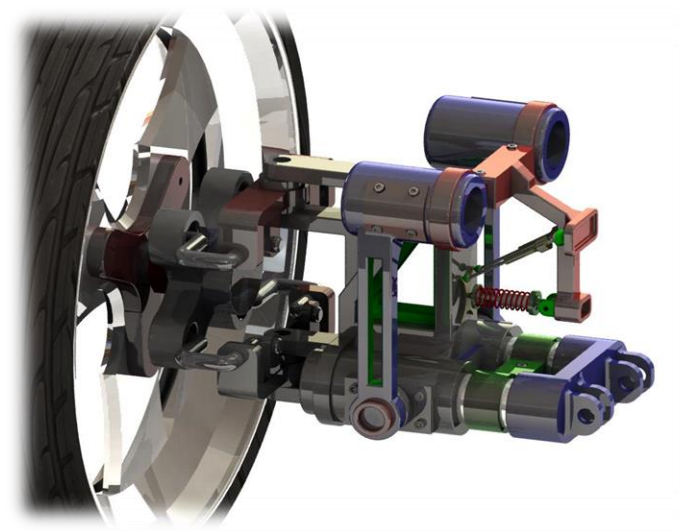
## ตัวอย่าง

1. ความต้องการให้พัฒนาประสิทธิภาพของระบบห้ามล้อ (Brake System) ของรถยนต์

ความต้องการคือ

- ระยะหยุด
- แรงดันของเบรก
- การขับขี่ในสภาพที่ไม่ปลอดภัย
- ความนุ่มนวลในการทำงาน

2. ความต้องการจะถูกจัดสรรให้กับฮาร์ดแวร์ของระบบเบรก และระบบป้องกันการล้อตาย (Anti-Lock Breaking System) และทำให้มีการกำหนดความต้องการในด้านอื่น ๆ เช่น ความต้องการด้านประสิทธิภาพของ ABS



---

การเจรจาต่อรองความต้องการ หรือเรียกว่า การแก้ไขข้อขัดแย้งระหว่างความต้องการ (Conflict Resolution) จะเกิดขึ้นเมื่อ

1. มีการนำเสนอแบบจำลองความต้องการต่อลูกค้า เพื่อรับทราบและยอมรับ
2. ลูกค้าพบข้อผิดพลาด หรือไม่พอใจในข้อกำหนดความต้องการ หรืออาจมีการเปลี่ยนแปลง

---

## ลักษณะความขัดแย้งหรือความไม่สอดคล้อง

- ความขัดแย้งระหว่างผู้มีส่วนได้ส่วนเสียที่มีความเห็นไม่ตรงกัน
- ความขัดแย้งระหว่างความต้องการกับทรัพยากรที่มีอยู่
- ความขัดแย้งระหว่างข้อกำหนดความต้องการที่เป็นหน้าที่หลักกับที่ไม่ใช่หน้าที่หลัก

## ความสำคัญในการแก้ไขความขัดแย้ง

- ไม่รีบแก้ไข หรือละเลยจะส่งผลย้อนหลัง
- กลายเป็นวัตถุดิบที่ไม่มีคุณภาพที่จะนำไปสู่การออกแบ

---

เมื่อผ่านขั้นตอนการวิเคราะห์ความต้องการ รายการความต้องการต้องมีลักษณะ ดังนี้

1. มีความถูกต้อง (Validity)
2. สอดคล้อง (Consistency)
3. เป็นไปได้ (Feasibility)
4. พร้อมนำไปจัดทำข้อกำหนดความต้องการในเอกสาร



## 5. การกำหนดความต้องการ

---

การกำหนดความต้องการ หรือ ข้อกำหนดความต้องการ (Requirement Specification)

**ข้อกำหนดความต้องการสำหรับวิศวกรรม** คือ การกำหนดค่าทางตัวเลขหรือกำหนดข้อจำกัดต่อเป้าหมายในการออกแบบผลิตภัณฑ์ แต่ข้อกำหนดความต้องการสำหรับซอฟต์แวร์ ค่าที่เป็นตัวเลขที่จะนำมาใช้เป็นข้อความในการออกแบบซอฟต์แวร์มีน้อยมาก ความต้องการส่วนใหญ่ของซอฟต์แวร์จะบ่งบอกถึงคุณลักษณะของซอฟต์แวร์มากกว่า

---

ข้อกำหนดความต้องการด้านซอฟต์แวร์ (Software Requirement Specification) หมายถึง การสร้างเอกสารความต้องการ แสดงรายละเอียดทางด้านซอฟต์แวร์ ที่สามารถตรวจสอบ ประเมินค่า และยอมรับได้

ระบบที่มีความซับซ้อนสูง เอกสารข้อกำหนดความต้องการของระบบ จะต้องประกอบด้วย (นอกเหนือจากคอมโพเนนต์ของซอฟต์แวร์)

- การนิยามระบบ (System Definition)
- ความต้องการด้านระบบ (System Requirement)
- ความต้องการด้านซอฟต์แวร์ (Software Requirement)

---

## การจัดทำข้อกำหนด (Specification) เอกสารที่เกี่ยวข้อง

1. เอกสารนิยามระบบ เป็นเอกสารที่ถูกจัดทำขึ้นจากมุมมองของผู้ใช้ โดยแสดงถึงรายการความต้องการด้านระบบ
2. เอกสารข้อกำหนดความต้องการด้านระบบ ดำเนินงานโดยวิศวกรระบบ มักถูกจัดทำก่อนข้อ 3
3. เอกสารข้อกำหนดความต้องการด้านซอฟต์แวร์ ระบุถึงหน้าที่ของซอฟต์แวร์ ซึ่งเป็นข้อตกลงขั้นพื้นฐานของทีมงานกับผู้ใช้

---

เอกสารนิยามระบบ (System Definition Document) หรือ เอกสารความต้องการของผู้ใช้ (User Requirement Document) คือ

1. เอกสารบันทึกความต้องการด้านระบบของผู้ใช้
2. เป็นการกำหนดความต้องการในระดับสูงจากมุมมองของผู้ใช้
3. ผู้ที่อ่านเอกสาร คือ กลุ่มของผู้ใช้หรือลูกค้า (และฝ่ายการตลาด)
4. เอกสารต้องเขียนด้วยภาษาที่เข้าใจง่าย หรือคำศัพท์ที่ลูกค้าใช้ในงานธุรกิจ

---

## รายละเอียดของเอกสารนิยามระบบ

- รายการความต้องการด้านระบบตามหลักการและเหตุผลหรือที่มาของระบบ และต้องสอดคล้องกับวัตถุประสงค์ของระบบ
- รายละเอียดของสภาพแวดล้อมภายนอกระบบ
- ข้อจำกัด ข้อสมมติฐาน ความต้องการที่ไม่ใช่หน้าที่หลักของระบบ
- แบบจำลองความต้องการในระดับสูง (Context System)
- แผนภาพลำดับเหตุการณ์ (Scenario) หรือเอนิตีตี (Entity)
- ข้อมูลและลำดับขั้นตอนการทำงาน

---

## เอกสารข้อกำหนดด้านซอฟต์แวร์

เอกสารข้อกำหนดความต้องการด้านระบบ (System Requirement Specification)

- นักพัฒนาระบบ จะแยกรายละเอียดความต้องการด้านระบบออกจากรายละเอียดความต้องการด้านซอฟต์แวร์
- เอกสารข้อกำหนดความต้องการด้านระบบ จะถูกกำหนดขึ้นมาก่อน เพื่อนำไปกำหนดความต้องการด้านซอฟต์แวร์
- การจัดทำเอกสารข้อกำหนดความต้องการด้านระบบ เป็นการดำเนินงานของวิศวกรรมระบบ

- 
- เอกสารที่ระบุถึงหน้าที่ของซอฟต์แวร์
  - ช่วยให้ทีมพัฒนาทราบว่าต้องพัฒนาอะไร
  - เอกสารข้อกำหนดความต้องการด้านซอฟต์แวร์ หรือ SRS เปรียบเสมือนข้อตกลงพื้นฐานระหว่างลูกค้ากับบริษัทผู้ผลิต
  - เอกสารนิยามซอฟต์แวร์จะเป็นบทนำก่อนเข้าสู่ความต้องการด้านซอฟต์แวร์

- 
- เอกสารข้อกำหนดความต้องการต้องผ่านการประเมินความต้องการ ก่อนนำไปใช้เป็นวัตถุดิบในการออกแบบ
  - เพื่อให้ความต้องการที่ระบุในเอกสารมีลักษณะที่จะนำไปประเมินราคา ความเสี่ยงและจัดตารางงาน
  - เมื่อข้อมูลความต้องการผ่านการประเมินแล้ว สามารถนำไปใช้ในการวางแผนการผลิตผลในการผลิตซอฟต์แวร์



## ตัวอย่างเอกสาร SRS (IEEE830,1993 [Jawadkar, 2005])

Chapter/Section	Topic
1	Introduction
1.1	บทคัดย่อ
1.2	วัตถุประสงค์ของเอกสาร
1.3	โครงสร้างของเอกสาร
1.4	คำศัพท์ที่ใช้
1.5	คำศัพท์ที่ใช้เฉพาะในเอกสารนี้
1.6	อื่น ๆ ที่ต้องการอ้างอิงถึง

Chapter/Section	Topic
2	System Description
2.1	ภาพรวมของระบบปัจจุบัน
2.2	ปัญหาที่พบในระบบปัจจุบัน
2.3	เป้าหมายของระบบที่พึงประสงค์
2.4	ภาพรวมของระบบที่พึงประสงค์
2.5	คุณลักษณะของผู้ใช้ระบบ
2.6	ข้อสมมติฐาน

Chapter/Section	Topic
3	Functional Requirements
3.1	ฟังก์ชันหลักของระบบทั้งหมด
3.2	ฟังก์ชัน/โมดูล-1
3.2.1	รายละเอียดของฟังก์ชัน (สร้างแบบจำลองประกอบ ถ้าจำเป็น)
3.2.2	การบรรจุเข้าหมายเหตุที่ต้องการ
3.2.3	ตัวอย่างความต้องการ

Chapter/Section	Topic
3.3	ฟังก์ชัน/โมดูล-2
3.3.1	รายละเอียดของฟังก์ชัน (สร้างแบบจำลองประกอบ ถ้าจำเป็น)
3.3.2	การบรรจุเป้าหมายที่ต้องการ
3.3.3	ตัวอย่างความต้องการ
3.4	ฟังก์ชัน/โมดูล-3
3.4.1	...
3.4.2	...
...	...

Chapter/Section	Topic
4	Non-Functional Requirements
4.1	ประสิทธิภาพ
4.2	ความได้ผลของเครื่องมืออรรถประโยชน์
4.3	ระบบรักษาความปลอดภัยของระบบ
4.4	ความปลอดภัยในการใช้งาน
4.5	สมรรถภาพ
4.6	ส่วนประสานกับผู้ใช้
4.7	ใช้งานได้ดี

Chapter/Section	Topic
4.8	ความน่าเชื่อถือ
4.9	ความถูกต้องแม่นยำ
4.10	การนำไปใช้ได้อีก
4.11	ง่ายต่อการใช้งาน
4.12	ความสามารถทำงานร่วมกันได้
4.13	ความสามารถเข้ากันได้กับระบบอื่น
4.14	มีการกำหนดระดับความเป็นส่วนตัว
4.15	สามารถดูแลระบบได้ง่าย

Chapter/Section	Topic
4.16	การขยายระบบในอนาคต
4.17	ความสามารถในการดูแลรักษา
4.18	ความสามารถในการทดสอบ
5	Design Constraint
6	Project Requirement (Optional)
6.1	การประมาณการขนาดของซอฟต์แวร์
6.2	การประมาณแรงงานการผลิตซอฟต์แวร์
6.3	การประมาณการต้นทุน

Chapter/Section	Topic
6.4	ตารางการทำงาน
6.5	Platform ที่ใช้ในการพัฒนา
6.6	เงื่อนไขที่ใช้ในการยอมรับ
	Appendices
A	ประเด็นที่อยู่ในระหว่างศึกษา (Pending Issues)
B	อภิธานศัพท์ (Glossary)
C	ดัชนี (Index)



---

## การตรวจสอบความต้องการ

คือ ขั้นตอนสุดท้ายก่อนนำไปสู่การออกแบบ เป็นการวิเคราะห์และตรวจหาข้อผิดพลาดหรือปัญหาที่อาจเกิดขึ้นจากการทับซ้อนความต้องการมีความสำคัญ เพราะเมื่อเกิดปัญหาจะทำให้เกิดข้อผิดพลาดของซอฟต์แวร์ ส่งผลต่องบประมาณ

---

การตรวจสอบเอกสารข้อกำหนดความต้องการ ควรตรวจสอบตามลักษณะดังนี้

1. มีความเที่ยงตรง (Validity)
2. มีความสอดคล้อง (Consistency)
3. มีความครบถ้วนสมบูรณ์ (Completeness)
4. มีความเป็นไปได้ (Feasibility)
5. สามารถพิสูจน์ได้ (Verifiability)

---

## 1. มีความเที่ยงตรง (Validity)

- การกำหนดความต้องการจะตอบสนองความต้องการของผู้ใช้ ทุก ๆ กลุ่มอย่างเท่าเทียม
- ความต้องการควรเกิดจากความต้องการของผู้ใช้อย่างทั่วถึงและยุติธรรม

## 2. มีความสอดคล้อง (Consistency)

- ความต้องการในเอกสารจะต้องไม่ขัดแย้งกัน

---

### 3. มีความครบถ้วนสมบูรณ์ (Completeness)

- เอกสารต้องระบุรายละเอียดฟังก์ชันและบริการครบถ้วน

### 4. มีความเป็นไปได้ (Feasibility)

- เพื่อให้มั่นใจว่าความต้องการที่ระบุในเอกสารสามารถพัฒนาระบบได้จริง
- เทคโนโลยี งบประมาณ และระยะเวลาในการพัฒนา

### 5. สามารถพิสูจน์ได้ (Verifiability)

- ความต้องการจะต้องพิสูจน์เพื่อหาความจริงได้
  - สามารถทดสอบ ทดลอง ให้เห็นถึงการทำงานจริงของระบบ

---

## เทคนิคในการตรวจสอบความต้องการ

### 1. การทบทวนความต้องการ (requirement review)

- โดยมีการตรวจสอบเอกสารอย่างละเอียด เพื่อหาข้อผิดพลาดตามลักษณะต่างๆ
- การทบทวนความต้องการอาจเป็นแบบทางการหรือไม่เป็นทางการก็ได้ขึ้นอยู่กับองค์กร
- การทบทวนแบบไม่เป็นทางการ (Informal Review) ทีมทบทวนจะนำเอกสารความต้องการมาพิจารณาหาข้อผิดพลาดร่วมกับผู้มีส่วนได้ส่วนเสียและผู้รับเหมาช่วง
- การทบทวนแบบเป็นทางการ (Formal Review) ทีมทบทวนจะต้องพิจารณาความต้องการร่วมกับผู้ใช้ที่ละรายการ

- 
- สามารถพิสูจน์ได้ (Verification) ความต้องการนั้นต้องสามารถพิสูจน์การทำงานหรือทดลองได้
  - สามารถเข้าใจได้ (Comprehensibility) ผู้ใช้สามารถเข้าใจในความต้องการนั้นหรือไม่
  - สามารถย้อนกลับไปตรวจสอบได้ (Traceability) สามารถย้อนกลับไปตรวจสอบแหล่งที่มาของความต้องการ เมื่อต้องมีการเปลี่ยนแปลง
  - สามารถดัดแปลงได้ (Adaptability) ความต้องการจะต้องสามารถดัดแปลงได้โดยไม่ส่งผลกระทบต่อระบบ

---

## 2. การจัดทำต้นแบบ (prototyping)

- เป็นการสร้างต้นแบบของระบบ (Executable Model)
- เพื่อสาธิตให้ลูกค้าหรือผู้ใช้ระบบดู หรือทดลองใช้ด้วยตนเอง
- ตรวจสอบว่าตรงตามความต้องการของลูกค้าหรือไม่
- เป็นวิธีที่ช่วยรวบรวมความต้องการที่เกิดขึ้นใหม่
- เป็นเทคนิคที่ดีในการตรวจสอบความต้องการ มีความชัดเจน
- การสร้างต้นแบบต้องใช้เงินทุนสูง แต่ได้ผลลัพธ์ที่ดี
- เป็นเทคนิคที่ดีที่สุดอย่างหนึ่ง เหมาะแก่การลงทุน

---

### 3. การสร้างแบบทดสอบ (Test-Case Generation)

- โดยนำแบบทดสอบนั้นไปออกแบบหรือพัฒนาระบบขึ้นใช้
- ถ้าทำได้ยาก ควรพิจารณาความต้องการนั้นใหม่
- มักใช้กับการพัฒนาซอฟต์แวร์แบบ Extreme Programming



---

#### 4. ข้อเสนอแนะสำหรับการตรวจสอบความต้องการ

- ทีมงานตรวจสอบความต้องการ ควรเป็นทีมงานอื่น ที่ไม่ใช่ทีมพัฒนา
- หากพบข้อผิดพลาดควรแก้ไขให้ถูกต้องและจัดทำเอกสารพร้อมกับเสนอวิธีแก้ไขปัญหา
- ทุกครั้งที่มีการแก้ไขข้อมูลความต้องการ ต้องมั่นใจว่ามีความสอดคล้องและถูกต้องเสมอ
- ควรระบุเทคนิคหรือวิธีการที่ใช้ในการตรวจสอบความต้องการไว้ในสัญญา

## 6. การจัดการความต้องการ (Requirement Management)

---

คือ กระบวนการทำความเข้าใจและควบคุมความเปลี่ยนแปลงของความต้องการของระบบ [Somerville, 2007] สามารถเริ่มดำเนินการได้ทันทีที่จัดทำเอกสารข้อกำหนดความต้องการฉบับร่างเสร็จ การวางแผนการจัดการความต้องการ ควรเริ่มตั้งแต่ขั้นตอนการสกัดความต้องการ

โดยที่ความต้องการในระบบมักเปลี่ยนแปลงไปตลอดช่วงชีวิตของระบบ การจัดการความต้องการเป็นชุดของกิจกรรมที่ช่วยให้ทีมงานกำหนดกลไกในการควบคุมและติดตามความสำเร็จและการเปลี่ยนแปลงความต้องการ ณ เวลาใดเวลาหนึ่งขณะที่โครงการดำเนินไป

---

## สาเหตุของการเปลี่ยนแปลงความต้องการ

1. มีผู้ใช้หลายกลุ่มซึ่งมีความต้องการต่างกัน จึงมีความขัดแย้งกัน จำเป็นต้องมีการปรับสมดุลความต้องการใหม่
2. เกิดความขัดแย้งระหว่างผู้ใช้ที่จ่ายเงินลงทุน กับผู้ใช้ที่เป็นผู้ใช้ระบบโดยตรง
3. มีการเปลี่ยนสภาพแวดล้อมทางธุรกิจและเทคโนโลยีภายหลังมีการติดตั้งใช้ระบบงาน

---

## กระบวนการ

1. จำแนกความต้องการที่เปลี่ยนแปลงและความต้องการที่ไม่เปลี่ยนแปลง
2. วางแผนการจัดการความต้องการ โดยระบุความเป็นเอกลักษณ์ให้กับทุกความต้องการ กำหนดกิจกรรมการประเมินผล กำหนดความสัมพันธ์ของความต้องการแต่ละรายการและใช้ Case Tool เข้าช่วยจัดการความต้องการ
3. จัดการกับการเปลี่ยนแปลงความต้องการ เพื่อให้การเปลี่ยนแปลงทั้งหมดที่เกิดขึ้นมีความสอดคล้องกัน โดยอาศัยหลักการจัดการโครงข่ายของระบบ

---

## ความต้องการที่เปลี่ยนแปลงและไม่เปลี่ยนแปลง

วิวัฒนาการของความต้องการ แบ่งความต้องการออกเป็น 2 ประเภท

### 1. ความต้องการที่ไม่เปลี่ยนแปลง (Enduring Requirement)

- เป็นความต้องการแบบคงที่ ไม่เปลี่ยนแปลงได้ง่าย
- เป็นความต้องการที่เกิดจากการทำงานหลักของธุรกิจในแต่ละวัน

### 2. ความต้องการที่เปลี่ยนแปลง (Volatile Requirement)

- เป็นความต้องการที่มีการเปลี่ยนแปลงอยู่เสมอในระหว่างการพัฒนาระบบหรือหลังจากการติดตั้งระบบเพื่อใช้งานไปแล้ว

การจัดการความต้องการเป็นกระบวนการที่ต้องใช้งบประมาณค่อนข้างสูง

---

การวางแผนก่อนเริ่มดำเนินงาน ตามกิจกรรมต่อไปนี้

1. **จำแนกความต้องการ (Requirement Identification)** ทีมงานต้องทำการระบุความเป็นเอกลักษณ์ให้กับทุกความต้องการ เพื่อไม่ให้ความต้องการซ้ำซ้อนกัน และเพื่อการอ้างอิง
2. **กระบวนการจัดการการเปลี่ยนแปลง (Change Management Process)** ทีมงานต้องกำหนดกิจกรรมในการประเมินผลกระทบและต้นทุนที่เกิดจากการเปลี่ยนแปลง
3. **นโยบายการสืบหา (Traceability Policy)** เป็นการกำหนดความสัมพันธ์ระหว่างความต้องการแต่ละรายการ และเก็บบันทึก เพื่อประโยชน์ในการบำรุงรักษา
4. **CASE Tool** ทีมงานต้องสรรหาเครื่องมือเข้ามาสนับสนุนกระบวนการจัดการความต้องการ เครื่องมือจะช่วยให้จัดการข้อมูลได้ง่ายขึ้น

---

การจัดการความต้องการ จะพิจารณาในเรื่องของความสัมพันธ์ระหว่างความต้องการกับการ  
ออกแบบระบบเมื่อเกิดการเปลี่ยนแปลง จะต้องทำการออกแบบหรือแก้ไขในส่วนนั้นใหม่การ  
เปลี่ยนแปลงความต้องการย่อมส่งผลกระทบต่อส่วนอื่นที่สัมพันธ์กัน การสืบหาส่วนที่ได้รับ  
ผลกระทบหรือแหล่งที่มาของความต้องการจึงมีความจำเป็น

---

## การสืบหา แบ่งออกเป็น 3 ชนิด ดังนี้

1. การสืบหาแหล่งที่มา (Source Traceability) เป็นการสืบหาแหล่งที่มาของการเปลี่ยนแปลง เพื่อสอบถามถึงเหตุและช่วงเวลาการนำเสนอให้เปลี่ยน
2. การสืบหาความต้องการ (Requirement Traceability) เป็นการสืบหาจำนวนความต้องการที่ได้รับผลกระทบจากการเปลี่ยนแปลง
3. การสืบหาในส่วนการออกแบบ (Design Traceability) เป็นการสืบหาส่วนการออกแบบจากความต้องการที่เปลี่ยนแปลง เพื่อทำการแก้ไขให้ถูกต้อง




---

การจัดการความต้องการต้องให้ความสำคัญกับรายละเอียดต่าง ๆ ของความต้องการ ระบบขนาดใหญ่ย่อมต้องมีความต้องการจำนวนมาก ส่งผลให้การจัดการมีความยุ่งยากและต้องใช้เวลาอันจำเป็นต้องนำเครื่องมือเข้ามาช่วยสนับสนุนกระบวนการจัดการความต้องการ

วิศวกรรมซอฟต์แวร์มี CASE Tool ที่สนับสนุนงานสำคัญของกระบวนการ

1. แหล่งจัดเก็บความต้องการ (Requirement Storage)
2. การจัดการกับการเปลี่ยนแปลง (Change Management)
3. การจัดการความสามารถสืบหาได้ (Traceability Management)

เมื่อมีการเปลี่ยนแปลงความต้องการ ทีมงานหรือองค์กรต้องมีกระบวนการจัดการกับการเปลี่ยนแปลง (Requirement Change Management) เพื่อให้การเปลี่ยนแปลงที่เกิดขึ้นมีความสอดคล้องกับส่วนอื่นที่สัมพันธ์กัน และอยู่ภายใต้การควบคุมอย่างเป็นทางการ กระบวนการจัดการการเปลี่ยนแปลงจะเกี่ยวข้องกับกระบวนการวิเคราะห์ถึงความคุ้มค่า เมื่อต้องเปลี่ยนแปลง



**Any**  
**Questions**

---