

# Requirement Engineering

“ วิศวกรรมความต้องการ ”

By Kanjana Eiamsaard, Version 2.0 Created 2021-12-7

# Topics Covered

- ภาพรวม
- ประเภทของความต้องการ
- กระบวนการทางวิศวกรรมความต้องการ
- แบบจำลองทางวิศวกรรมความต้องการ

# ภาพรวม

- ความต้องการ (Requirement) คือ สิ่งที่ผู้ว่าจ้างต้องการให้เกิดขึ้นภายในระบบ เพื่อบรรลุถึงวัตถุประสงค์ในการจัดทำซอฟต์แวร์ เช่น ความสามารถ เงื่อนไขการใช้งาน
- วิศวกรรมความต้องการ (Requirement Engineering) คือ กระบวนการที่ประยุกต์องค์ความรู้ ทฤษฎี เครื่องมือ มาทำให้เกิดเป็นความต้องการที่แท้จริงของผู้ว่าจ้าง/ผู้ใช้
  - เอกสารความต้องการ (Requirement Document)
  - ตัวอย่างชิ้นงาน (Prototype)
  - Etc.

## ภาพรวม (ต่อ)

- เอกสารความต้องการ (Requirement specification)
  - ประกอบสัญญาจ้าง
  - แบ่งประเภทตามวัตถุประสงค์ในการนำไปใช้
    - เอกสารความต้องการของผู้ใช้ (User requirement document) – ลูกค้า/ผู้ใช้ อ่านแล้วรู้ว่าจะได้ sw ที่มีความสามารถอะไรบ้าง
    - เอกสารข้อกำหนดความต้องการของระบบ (System requirement specification document) – ให้ทีมผู้พัฒนาเข้าใจโดยละเอียด
      - เมื่อไร ที่ไหน กระทำโดยใคร กระทำกับใคร ทำอย่างไร และ ทำไปเพื่ออะไร

## ภาพรวม (ต่อ)

ประเภทเอกสาร	รายการ	คำถามนำ
เอกสารความต้องการของผู้ใช้	1. ระบบสามารถสร้างรายงานประจำเดือนที่แสดงค่ายาที่จ่ายให้คนไข้ในแต่ละเดือนในคลินิกแต่ละสาขาได้	อะไร
เอกสารข้อกำหนดความต้องการของระบบ	1.1 วันทำงานสุดท้ายของเดือน รายงานสรุปของตัวยาและราคายาที่ถูกสั่งจ่ายในคลินิกแต่ละสาขาจะต้องถูกสร้างขึ้น	เมื่อไร, ที่ไหน
	1.2 ระบบจะสร้างรายงานสำหรับพิมพ์ลงกระดาษหลังจาก 17.30 น. ในวันทำงานสุดท้ายของเดือนเท่านั้น	เมื่อไร
	1.3 รายงานการจ่ายยาถูกสร้างตามข้อมูลของแต่ละคลินิก โดยแสดงรายการแยกตามชื่อยา ประกอบด้วย ปริมาณที่จ่าย ขนาดยา (doses) และยอดเงินรวม	อย่างไร
	1.4 ถ้ายาที่จ่ายมีหลายขนาด เช่น 10mg 20 mg เป็นต้น ให้แยกรายการออกจากกัน	อย่างไร
	1.5 การเข้าถึงรายงานของรายได้จากการจ่ายยา สงวนไว้เฉพาะผู้ใช้ในระดับผู้บริหารเท่านั้น ซึ่งจะปรากฏอยู่ในรายการควบคุมการเข้าถึง (Access Control List)	กระทำโดยใคร

## ภาพรวม (ต่อ)

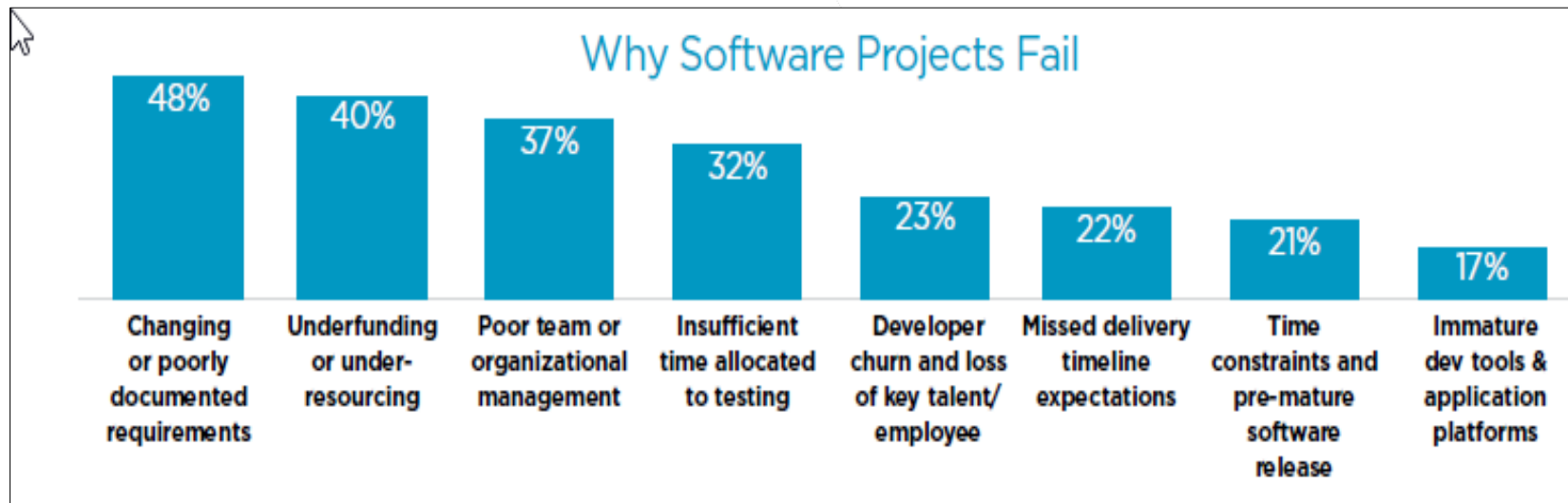
- “ความต้องการ” มีอิทธิพลต่อความสำเร็จของโครงการมากน้อยเพียงใด

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
11. Other	13.9%

เปรียบเทียบร้อยละของการมีส่วนร่วมในความสำเร็จของโครงการ (Gühl & Alam)

## ภาพรวม (ต่อ)


- จะเกิดอะไรขึ้น? - ถ้าไม่สามารถกำหนด “ความต้องการ(ที่ถูกต้อง)” ได้



สาเหตุที่ทำให้โครงการพัฒนาซอฟต์แวร์ล้มเหลว ปี ค.ศ. 2015 สํารวจโดย <https://www.developersalliance.org/>



## ● ประเภทของความต้องการ

- Functional Requirement
  - Non-Functional Requirement
  - ทำไมต้องแบ่งประเภท ?
- 



# Functional Requirement (FR)

- คือ สิ่งที่ระบุความสามารถของระบบ ข้อจำกัดในการนำระบบไปใช้ โดยสิ่งเหล่านี้จะทำให้บรรลุวัตถุประสงค์ในการพัฒนาระบบ (เพื่อนำมาทดแทนรูปแบบงานเดิม)
  - วิธีการที่ระบบตอบสนองต่อข้อมูลเข้า
  - วิธีการที่ระบบตอบสนองต่อเหตุการณ์
- ตรวจสอบ FR ในด้าน
  - ความครบถ้วน (Completeness)
  - ความสอดคล้อง (Consistency)

# Functional Requirement (ต่อ)

ประเภทเอกสาร	รายการ	คำถามนำ
เอกสารความต้องการของผู้ใช้	1. ระบบสามารถสร้างรายงานประจำเดือนที่แสดงค่ายาที่จ่ายให้คนไข้ในแต่ละเดือนในคลินิกแต่ละสาขาได้	อะไร
เอกสารข้อกำหนดความต้องการของระบบ	1.1 วันทำงานสุดท้ายของเดือน รายงานสรุปของตัวยาและราคายาที่ถูกสั่งจ่ายในคลินิกแต่ละสาขาจะต้องถูกสร้างขึ้น	เมื่อไร, ที่ไหน
	1.2 ระบบจะสร้างรายงานสำหรับพิมพ์ลงกระดาษหลังจาก 17.30 น. ในวันทำงานสุดท้ายของเดือนเท่านั้น	เมื่อไร
	1.3 รายงานการจ่ายยาถูกสร้างตามข้อมูลของแต่ละคลินิก โดยแสดงรายการแยกตามชื่อยา ประกอบด้วย ปริมาณที่จ่าย ขนาดยา (doses) และยอดเงินรวม	อย่างไร
	1.4 ถ้ายาที่จ่ายมีหลายขนาด เช่น 10mg 20 mg เป็นต้น ให้แยกรายการออกจากกัน	อย่างไร
	1.5 การเข้าถึงรายงานของรายได้จากการจ่ายยา สงวนไว้เฉพาะผู้ใช้ในระดับผู้บริหารเท่านั้น ซึ่งจะปรากฏอยู่ในรายการควบคุมการเข้าถึง (Access Control List)	กระทำโดยใคร

# Non-Functional Requirement (NFR)

- สิ่งอธิบายถึงระดับคุณภาพและเงื่อนไขในการใช้งานระบบ
  - ความน่าเชื่อถือ เวลาในการตอบสนอง ความเร็วในการเขียนอ่านข้อมูล ลักษณะการแสดงผล
  - (อาจ)กำหนดรูปแบบการพัฒนาซอฟต์แวร์ เครื่องมือ ภาษาโปรแกรม ที่ทีมพัฒนาซอฟต์แวร์ต้องปฏิบัติตาม
- ความต้องการเชิงคุณภาพที่ผิดพลาดอาจส่งผลกระทบต่อการนำซอฟต์แวร์ไปใช้ได้มากกว่าความต้องการเชิงหน้าที่
  - หากระบบพัฒนาความต้องการเชิงฟังก์ชันบางรายการไม่ตรงกับความต้องการ จะพบว่าผู้ใช้มักหาทางใช้งานฟังก์ชันอื่นเพื่อให้ได้ผลลัพธ์ตามที่ลูกค้าต้องการได้

# Non-Functional Requirement (ต่อ)

- มาตรฐานด้านคุณภาพของซอฟต์แวร์ - ISO/IEC 25010



# Non-Functional Requirement (ต่อ)

- การเกิดขึ้นของ NFR หนึ่งอาจนำไปสู่ความต้องการเชิงฟังก์ชันที่ผู้ใช้ไม่ได้กล่าวถึง
- การเกิดขึ้นของ NFR หนึ่งอาจขัดแย้งกับ NFR อื่นได้
  - Usability != Security
- NFR ยังส่งผลต่อการออกแบบสถาปัตยกรรมของระบบ
  - Performance requirement --- Strong Cohesion & Loosely Coupling
- เช่น “ระบบต้องพร้อมให้บริการในช่วงเวลาทำงาน (จันทร์-ศุกร์ 08.30-17.30 น.) และระบบจะหยุดให้บริการได้ไม่เกิน 5 วินาทีต่อวัน”
  - Availability

# Non-Functional Requirement (ต่อ)


- NFR ไม่ถูกต้อง มักทำให้เกิดปัญหาในระหว่างการพัฒนาตลอดจนการส่งมอบงาน
  - เช่น “ระบบพัฒนาให้ผู้ใช้สามารถใช้งานได้ง่าย” – แบบไหนที่เรียกว่าง่าย ?
  - กำหนดตัวชี้วัด – เอาให้ชัด ๆ ไปเลย

# Non-Functional Requirement (ต่อ)

ชื่อคุณสมบัติ	ตัวอย่างตัวชี้วัด
ความเร็ว (Speed)	<ul style="list-style-type: none"><li>- จำนวน transactions ที่ประมวลผลได้ในหนึ่งวินาที</li><li>- เวลาที่ใช้เมื่อ refresh หน้าจอ</li></ul>
ขนาด (Size)	หน่วยวัดขนาดพื้นที่ในระบบคอมพิวเตอร์ เช่น Mbytes
ง่ายต่อการใช้งาน (Ease of use)	<ul style="list-style-type: none"><li>- เวลาที่ใช้ศึกษา/อบรม</li><li>- จำนวนตัวช่วยที่ปรากฏ</li></ul>
ความน่าเชื่อถือ (Reliability)	<ul style="list-style-type: none"><li>- เวลาที่ระบบล้มเหลวโดยเฉลี่ย (Mean time to failure)</li><li>- ความน่าจะเป็นที่ระบบไม่สามารถเข้าถึงได้</li><li>- อัตราการเกิดข้อผิดพลาด</li><li>- ความพร้อมใช้ของระบบ</li></ul>
ความทนทาน (Robustness)	<ul style="list-style-type: none"><li>- เวลาที่ใช้เริ่มต้นระบบอีกครั้งหลังเกิดข้อผิดพลาด</li><li>- ร้อยละของเหตุการณ์ที่ทำให้เกิดข้อผิดพลาด</li><li>- ความน่าจะเป็นที่ข้อมูลผิดพลาดเมื่อระบบเกิดความเสียหาย</li></ul>
ความสามารถในการเคลื่อนย้าย (Portability)	<ul style="list-style-type: none"><li>- ร้อยละของสิ่งที่ไม่เป็นอิสระจากกัน</li></ul>



## กระบวนการทางวิศวกรรมความต้องการ

- Requirement Elicitation
  - Requirement Specification
  - Requirement Validation
- 



# กระบวนการทางวิศวกรรมความต้องการ (Requirement Engineering Processes)

- กลุ่มของกิจกรรมที่ผู้ว่าจ้างและผู้พัฒนาดำเนินการร่วมกัน ประกอบด้วย
  - Requirement Elicitation
  - Requirement Specification
  - Requirement Validation
- เพื่อให้ทีมผู้พัฒนาสามารถกำหนดขอบเขตในการแก้ปัญหาด้วยซอฟต์แวร์ได้
- ผู้ว่าจ้างมีหน้าที่ให้ข้อมูลที่จำต่อผู้พัฒนา
  - ผู้พัฒนามีหน้าที่กำหนดกิจกรรมที่ก่อให้เกิดข้อมูลที่ ครอบคลุม ถูกต้อง สอดคล้อง

# Requirement Elicitation

- กิจกรรมที่กำหนดขึ้นเพื่อแสวงหาความต้องการ แบ่งเป็น 3 กิจกรรมย่อย
  - Requirements discovery ผู้พัฒนาต้องค้นหาผู้มีส่วนได้ส่วนเสีย (Stakeholders) ของระบบ
    - เครื่องมือ: Mind Mapping หรือ ต้นแบบเอกสาร (ถ้ามี)
  - Requirements classification and organization
    - เครื่องมือ: โปรแกรมจัดทำเอกสารทั่วไป, Post-it หรือ เครื่องมือที่เคลื่อนย้ายข้อความได้สะดวก เป็นต้น
  - Requirements prioritization and negotiation
    - เครื่องมือ: เกณฑ์สำหรับระบุระดับความสำคัญ เช่น H/M/L , Complexity Bucket, Planning เป็นต้น

# Requirement Elicitation --- Requirements discovery

Project name	ระบุชื่อโครงการ	Project No.	ระบุหมายเลขโครงการ
Responsible	ระบุผู้รับผิดชอบโครงการ อาจใส่เฉพาะหัวหน้าโครงการได้		
Version/Date	รุ่นและวันที่ในการทำเอกสารฉบับนี้	Status	สถานะของโครงการ

ผู้มีส่วนได้ส่วนเสีย (Stakeholder)	ตำแหน่ง/บทบาทใน โครงการ	ทัศนคติที่มีต่อโครงการ	อิทธิพลที่มีต่อโครงการ	ตัวชี้วัด / กลยุทธ์ในการทำงาน (Measures/strategies)
ชื่อ นามสกุล ของผู้มีส่วนได้ส่วนเสีย	ระบุบทบาทในโครงการ	ระบุระดับความเห็นที่มีต่อโครงการ เช่น - เห็นด้วยอย่างยิ่ง - เห็นด้วย - ปานกลาง - ไม่เห็นด้วย - ไม่เห็นด้วยอย่างยิ่ง	ระบุระดับอิทธิพลที่ผู้มีส่วนได้ส่วนเสีย ส่งผลให้โครงการสำเร็จ เช่น สูง กลาง ต่ำ	ระบุวิธีที่นักพัฒนาเลือกใช้ในการทำงานร่วมกับผู้มีส่วนได้ส่วนเสีย และอาจรวมถึงตัวชี้วัดที่แสดงให้เห็นถึงความก้าวหน้าในการทำงาน เช่น วิธีการทำงานร่วมกับผู้ไม่เห็นด้วยกับการทำโครงการแต่มีอิทธิพลต่อความสำเร็จของโครงการสูง เป็นต้น
คุณสมหญิง	เจ้าหน้าที่การเงิน	ไม่เห็นด้วย	สูง	<ol style="list-style-type: none"> <li>หากมีแหล่งข้อมูลเดิมให้ทีมพัฒนารวบรวมข้อมูลที่เกี่ยวข้องให้มากที่สุด</li> <li>จัดทำตัวอย่างชิ้นงานประกอบการรวบรวมความต้องการ</li> <li>อื่น ๆ</li> </ol>

# รู้จัก Stakeholder แล้วทำอะไรต่อ ?

- กำหนดกิจกรรมที่เหมาะสมกับ stakeholder คนนั้น ๆ ตามกลยุทธ์ฯ ที่วิเคราะห์ได้
  - การสัมภาษณ์
  - การทำต้นแบบซอฟต์แวร์ (Throwaway prototype & Software prototype)
  - การประชุมเชิงปฏิบัติการ (Workshop)
  - การระดมความคิด (Brainstorm)
  - ยกตัวอย่างด้วยชิ้นงานที่คล้ายกัน
- ณ ขณะนี้ สิ่งที่ทีมพัฒนาควรได้รับคือ ความจริง ความคิดเห็น ข้อมูลประกอบ ตัวอย่างชิ้นงาน กระบวนการทำงานแบบเดิม (ก่อนมีระบบ) และอื่น ๆ ที่จะเป็นประโยชน์ต่อทีม --- แบบการจัดกระจาย ยังไม่เป็นรูปเป็นร่าง

## Requirement Elicitation --- Requirements classification and organization

- นำความต้องการที่ได้มาจัดกลุ่มตามความเกี่ยวข้องในมุมมองของผู้พัฒนา
- เพื่อให้ทำงานต่อได้ง่าย

## Requirement Elicitation --- Requirements prioritization and negotiation

- นำความต้องการที่ผ่านกิจกรรมที่ 1 และ 2 มาจัดลำดับความสำคัญและประเมินความซับซ้อน
- ต่อรองเพื่อขจัดความต้องการที่ขัดแย้งหรือไม่เป็นเหตุเป็นผลออกจากรายการความต้องการ
- จำเป็นต้องได้รับความเห็นชอบจากผู้ว่าจ้างด้วย

# Requirement Elicitation --- Requirements prioritization and negotiation

Project name	ระบุชื่อโครงการ	Project No.	ระบุหมายเลขโครงการ
Responsible	ระบุผู้รับผิดชอบโครงการ อาจใส่เฉพาะหัวหน้าโครงการได้		
Version/Date	รุ่นและวันที่ในการทำเอกสารฉบับนี้		

ลำดับที่	รายการความต้องการ	ผู้มีส่วนได้ส่วนเสีย (Stakeholder)	ความซับซ้อน	ลำดับความสำคัญ	สถานะการดำเนินงาน	หมายเหตุ
1	ระบุรายการความต้องการ (สามารถนำความต้องการในระดับ User requirement หรือ System requirement มาใช้)	ชื่อ นามสกุล ของผู้มีส่วนได้ส่วนเสีย	ระบุระดับความซับซ้อน	ระบุระดับความสำคัญ เช่น สูง กลาง หรือ ต่ำ	ระบุสถานะการดำเนินการ เช่น - กำลังวางแผน - อยู่ในระหว่าง ดำเนินการ - ดำเนินการเรียบร้อยแล้ว	
2.	ระบบสามารถสร้างรายงานประจำเดือนที่แสดงค่ายาที่จ่ายให้คนไข้ ในแต่ละเดือนของคลินิกแต่ละสาขาได้	คุณสมหมาย (หัวหน้าแผนกบัญชี)	สูง	สูง	-อยู่ในระหว่าง ดำเนินการ	

# Requirement Specification

- กระบวนการจัดทำเอกสารเพื่อบันทึกผลลัพธ์ที่ได้จากขั้นตอนการแสวงหาความต้องการ (Requirement elicitation) ทั้งความต้องการเชิงฟังก์ชันและความต้องการเชิงคุณภาพ โดยรูปแบบของเอกสารขึ้นอยู่กับความเหมาะสม
  - การเขียนบรรยายความต้องการด้วยภาษาธรรมชาติ
  - การเขียนความต้องการภายใต้แบบฟอร์ม เช่น IEEE 830 เป็นต้น
  - การเขียนความต้องการด้วยภาษาธรรมชาติร่วมกับแผนภาพ เช่น แผนภาพยูสเคส (Use case diagram) แผนภาพลำดับ (Sequence diagram) เป็นต้น
  - การเขียนความต้องการแบบเนื้อเรื่องของผู้ใช้ (User stories)



# การเขียนความต้องการแบบเนื้อเรื่องของผู้ใช้ (User stories)

- เพื่อกระตุ้นให้นักพัฒนาระบุความต้องการที่ชัดเจนได้ นิยมใช้รูปแบบประโยค
  - “As a <user> (who) I want to...<what> so that...<why>”
  - ในฐานะที่ฉันเป็นนิสิต ฉันต้องการเห็นผลการลงทะเบียนเทียบหลักสูตร เพื่อสามารถวางแผนการเรียนด้วยตัวเองได้
  - ในฐานะที่ฉันเป็นผู้ดูแลผู้สูงอายุ ฉันอยากได้รับการเตือนเมื่อผู้สูงอายุภายใต้การดูแลล้มลงอย่างทันที เพื่อที่จะวางแผนการช่วยเหลือได้
  - ในฐานะที่ฉันเป็นผู้ดูแลผู้สูงอายุ ฉันอยากทราบตำแหน่งที่ผู้สูงอายุล้ม เพื่อที่จะติดต่อคนในบริเวณใกล้เคียงได้
  - ในฐานะที่ฉันเป็นผู้สูงอายุ ฉันต้องการขอความช่วยเหลือจากผู้ดูแลผ่านการใช้เสียงได้ เพื่อเป็นทางเลือกในกรณีไม่สามารถขอความช่วยเหลือทางอื่นได้

# Requirement Validation

- การตรวจสอบความต้องการที่ระบุไว้ในเอกสาร ว่าเป็นความสามารถหรือคุณภาพตามลูกค้าต้องการหรือไม่
- กิจกรรมนี้ ทำแล้วลดความเสียหายที่เกิดต่อโครงการได้เป็นอย่างมาก
- ตรวจสอบอะไร ?
  - ความสมเหตุสมผล (Validity) ระบบมีฟังก์ชันที่สนับสนุนความต้องการของผู้ใช้ได้ดีที่สุดหรือไม่
  - ความสอดคล้อง (Consistency) มีความต้องการใดขัดแย้งกันบ้างหรือไม่
  - ความสมบูรณ์ (Completeness) มีฟังก์ชันที่ลูกค้าต้องการอยู่ครบถ้วนหรือไม่
  - ความสมจริง (Realism) ความต้องการที่ระบุมาทำได้จริงหรือไม่ ทั้งในแง่ของงบประมาณและความสามารถทางเทคนิค
  - การพิสูจน์ความจริงได้ (Verifiability) ความต้องการที่ระบุมา สามารถยืนยันความถูกต้องได้หรือไม่ (รู้ได้มาจากใคร)

# Requirement Validation (ต่อ)

- ตรวจสอบอย่างไร ?
  - การอ่านทบทวนเอกสารโดยผู้พัฒนาและผู้ว่าจ้าง (Requirement reviews)
  - การทำต้นแบบซอฟต์แวร์ (Software prototype)
  - สร้างกรณีทดสอบ (Test case generation)
  - Etc.



## แบบจำลอง

### กระบวนการวิศวกรรมความต้องการ

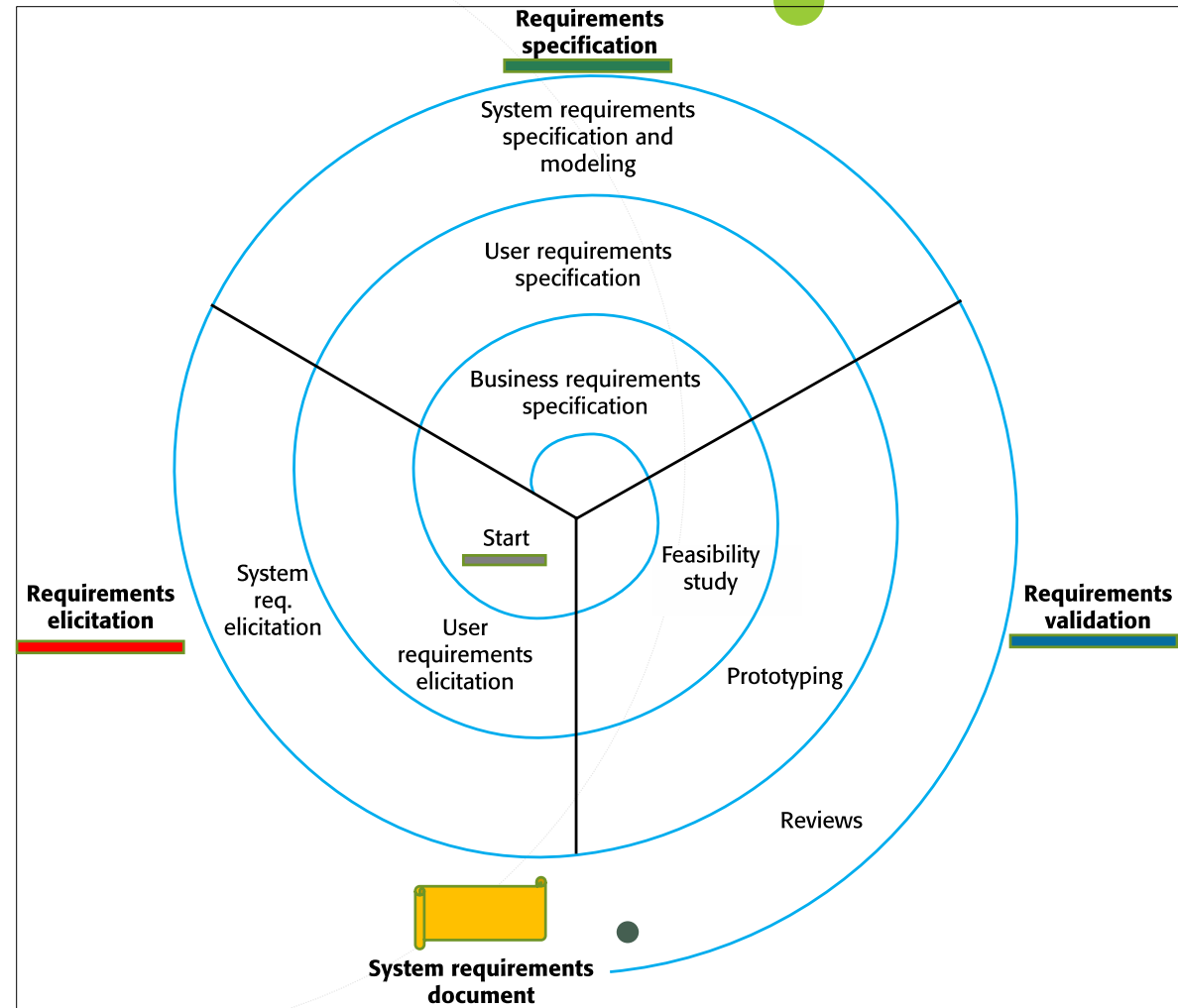
- Spiral process model



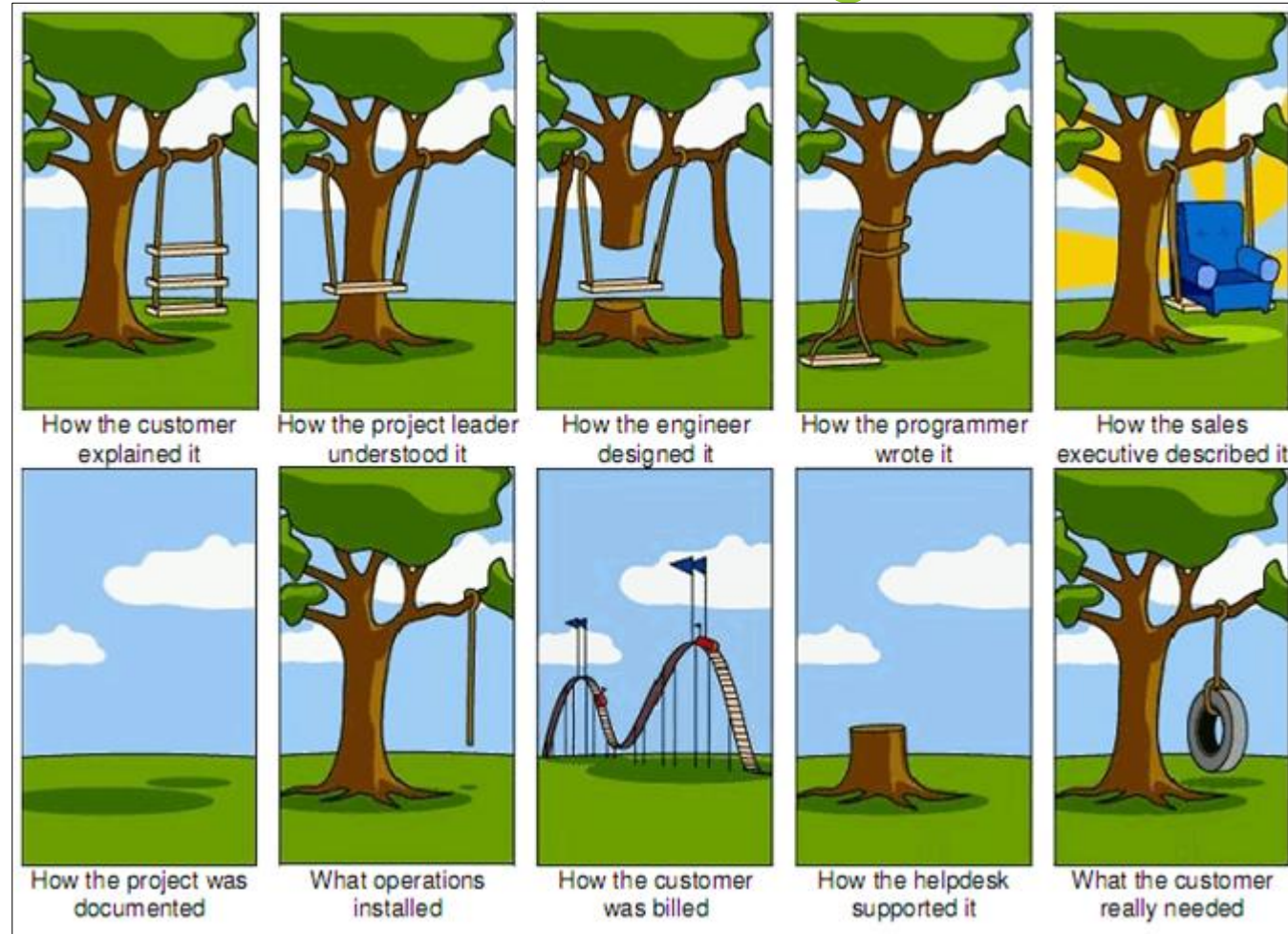
# แบบจำลองกระบวนการวิศวกรรมความต้องการ

- “แบบจำลอง” คือ การนำเสนอแนวคิด/วิธีการ ส่วนมากใช้ภาพในการนำเสนอ
  - แนวคิดในการนำกระบวนการวิศวกรรมความต้องการไปใช้
- มีหลายแบบจำลอง หนึ่งในนั้นคือ Spiral process model
  - Software Engineer มีหน้าที่ศึกษาแนวคิดต่าง ๆ และนำมาประยุกต์ใช้กับโปรเจกต์ที่ได้รับมอบหมาย

# Spiral process model



# Requirement engineering in practice



# อ้างอิง

