Requirement Engineering

อ. ปรัชญ์ ปิยะวงศ์วิศาล

Pratch Piyawongwisal

Today

- ตรวจความคืบหน้าบน Trello + Daily Scrum
- Requirement Engineering
- System Modeling

HW: Project Progress

- ทุกคนในกลุ่มจะต้องเริ่มทำงานอย่างน้อยคนละ 1-2 tasks
- สัปดาห์นี้จะตรวจความคืบหน้าใน daily scrum
- จะต้องปรากฏชื่อทุกคนในคอลัมน์ In Progress หรือ Done
- scrum board ต้องสอดคล้องกับโค้ดเวอร์ชันล่าสุด ห้ามใส่เกินที่ทำจริง

Trello Tips

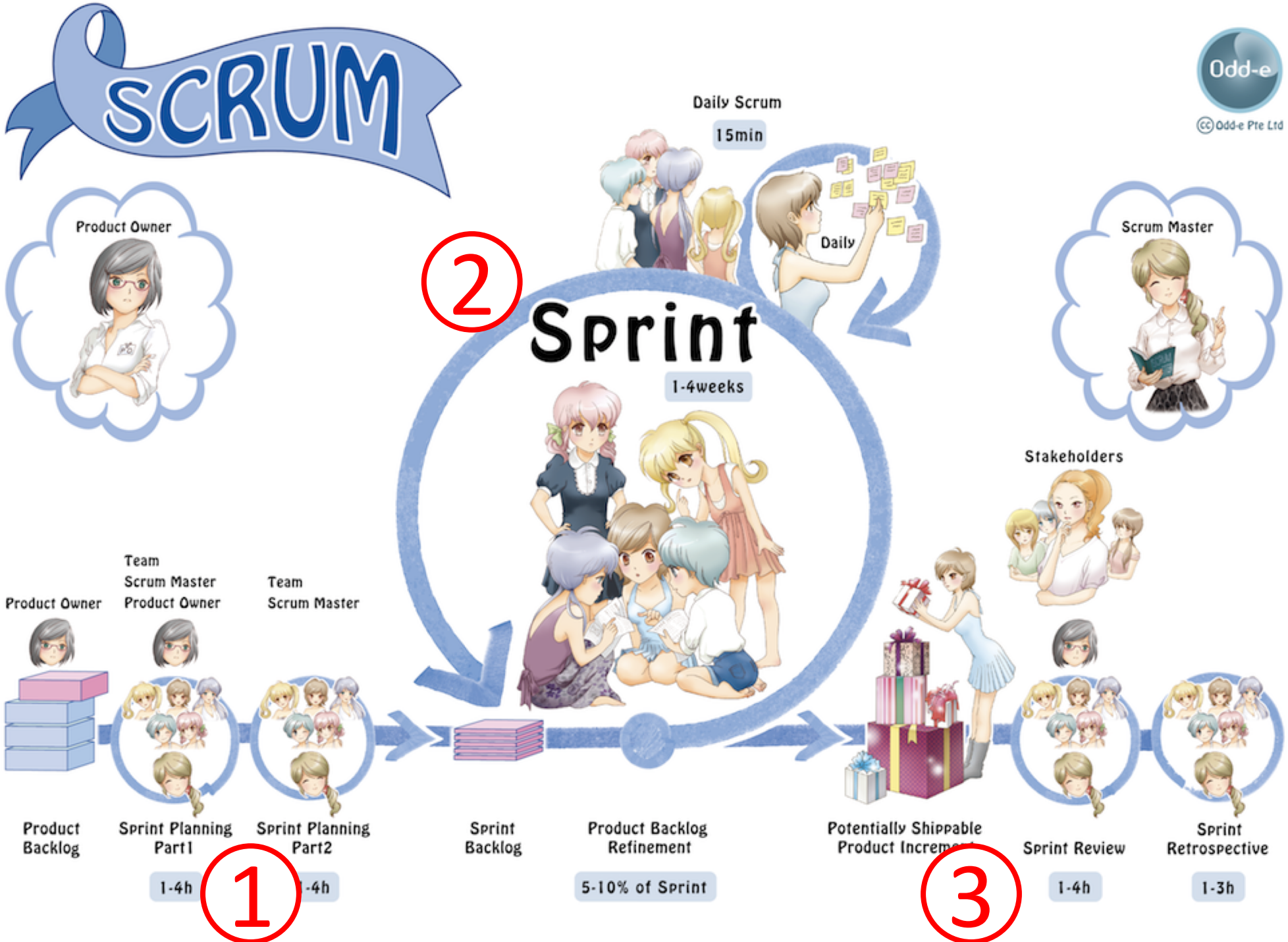
- การกำหนดผู้รับผิดชอบ Task
  - ลากไอคอน user มาวางบน card ได้
- การเชื่อม Task เข้ากับ User Story
  - ใน user story card กดเพิ่ม attachment -> Trello
  - เสิร์ชหา Task เพื่อสร้างลิงค์ story -> Task
  - กด connect cards… เพื่อให้เชื่อม Task -> story กลับด้วย
- แสดงหมายเลข card id
  - https://trello.com/power-ups/59c3d177178a761767b49278/card-numbers-by-reenhanced
- กำหนด story point ให้กับ user story
  - https://chrome.google.com/webstore/detail/scrum-for-trello/jdbcdblgjdpmfninkoogcfpnkjmndgje?hl=en

ตัวอย่าง User Stories ที่ใช้ได้

Project: Weather Radar App

- As a … I want … so that….
  - user, ดู radar แบบ real-time, ทราบตำแหน่งบนแผนที่ที่มีฝนตก
  - farmer, ดูคำแนะนำพันธุ์พืชที่ควรปลูก, สามารถปลูกพืชได้เหมาะสมตามสภาพอากาศ
  - tourist, ดูคำแนะนำสถานที่ท่องเที่ยว, สามารถเลือกสถานที่เที่ยวได้เหมาะสมตามสภาพอากาศ
  - user, ค้นหา history ของสภาพอากาศได้, สามารถนำสถิติไปใช้ในการพยากรณ์

Daily Scrum


SCRUM
Odd-e
Odd-e Pte Ltd
Daily Scrum
15min
Daily
Product Owner
Scrum Master
2
Sprint
1-4weeks
Stakeholders
Team
Scrum Master
Product Owner
Team
Scrum Master
Product Owner
Product Backlog
Sprint Planning Part1
1-4h
Sprint Planning Part2
1-4h
1
Sprint Backlog
Product Backlog Refinement
5-10% of Sprint
Potentially Shippable Product Increment
3
Sprint Review
1-4h
Sprint Retrospective
1-3h

Mid-Sprint - Daily Scrum

- 15 นาที – ผลัดกันพูดทีละคน
- มักใช้วิธียืนคุย
- อัพเดทความคืบหน้า, ปัญหาและอุปสรรค
- ตอบคำถาม 3 ข้อ
  - เมื่อวานทำอะไรไปบ้าง
  - วันนี้จะทำอะไร
  - ติดขัดอย่างไร

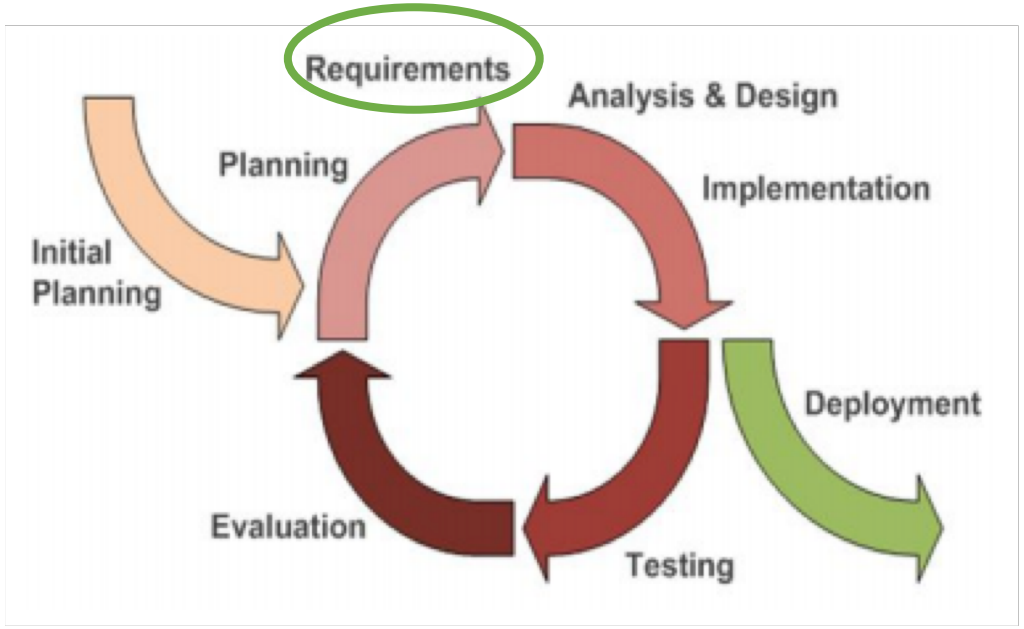
Homework: ทุกสัปดาห์ ให้ถ่าย screenshot Trello board เก็บไว้ใน Github folder

Requirement Engineering

Recap: Product Management/Requirement

- เฟสที่สองที่ Product manager จะมีบทบาทมาก
- ในการระบุ requirement จะจัดทำเป็นเอกสาร (artifact) ในรูปแบบของ
  - **User Stories**
  - Use case description
  - UML use case diagrams

- **User Story Examples**
  - User Story #71:
    As a Shopper, I want to print a receipt, so that I have a record of my purchase.
  - User Story #65:
    As a Store Manager, I want to offer limited time discount, so that I can increase sales and traffic.



Agile methods and requirements

- Many agile methods argue that producing detailed system requirements is a waste of time. Requirements change quickly, thus the requirements document is always out of date.
- This is problematic for systems that require pre-delivery analysis or systems developed by several teams.

Agile methods and requirements

- การเขียน requirement ด้วย User Stories สั้นๆ นั้นเป็นที่นิยมในวิธีการแบบ Agile
- เพราะแนวคิดแบบ Agile เชื่อว่าในเมื่อ requirement จะต้องเปลี่ยนบ่อยๆ ก็ไม่จำเป็นที่จะต้องเสียเวลาเขียน requirement ให้ละเอียด

- แต่ User Stories คงไม่เหมาะกับซอฟต์แวร์ใหญ่ๆ ที่จะต้องวิเคราะห์ requirement ให้ละเอียดก่อนเริ่มดำเนินการ หรืออย่างซอฟต์แวร์ที่เป็นตัวชี้ชะตาบริษัท
- ในคาบนี้ เราจึงจะศึกษากระบวนการ “Requirement Engineering”
- สำหรับงานที่ User Stories อย่างเดียวนั้นไม่พอ

Requirement Engineering

- เป็นกระบวนการในการระบุข้อกำหนด (requirement) ของระบบ
- requirement คือ คำอธิบายเกี่ยวกับ service, ฟังก์ชันการทำงานต่าง ๆ ที่ผู้ใช้ต้องการจากระบบ รวมถึงขีดจำกัด (constraint) ต่างๆ ของระบบ
- ประเภทของ requirement โดยแบ่งตาม target audience
  - User requirements
  - System requirements
- ประเภทของ requirement โดยแบ่งตามสิ่งที่ระบุ
  - Functional requirements
  - Non-Functional requirements

Types of Requirements

- แบ่งตาม target audience ได้ 2 แบบ
  - **User requirements**
    - สำหรับลูกค้า
    - high-level
    - อธิบายด้วยภาษาพูด ที่เข้าใจง่าย (อาจใช้แผนภาพ diagram ช่วย)
    - * นึกถึง product backlog ใน scrum
  - **System requirements**
    - สำหรับ Dev
    - low-level
    - เป็นเอกสารที่มีระบบ (structured) อธิบายถึง function การทำงานของระบบโดยละเอียด
    - * นึกถึง tasks ใน scrum

User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Types of Requirements

- หรือแบ่งตามสิ่งที่ระบุได้ 2 แบบ
  - **Functional requirements**
    - ระบุ function/feature ต่าง ๆ ที่ระบบจะต้องทำได้
    - อธิบายถึง business rule และ process ต่าง ๆ ของระบบ
    - อาจแต่งเป็นประโยค
    - หรือใช้ use case diagram เพื่อแสดงถึงปฏิสัมพันธ์ระหว่าง user กับระบบ
  - **Non-Functional requirements**

ตัวอย่าง Functional Requirements

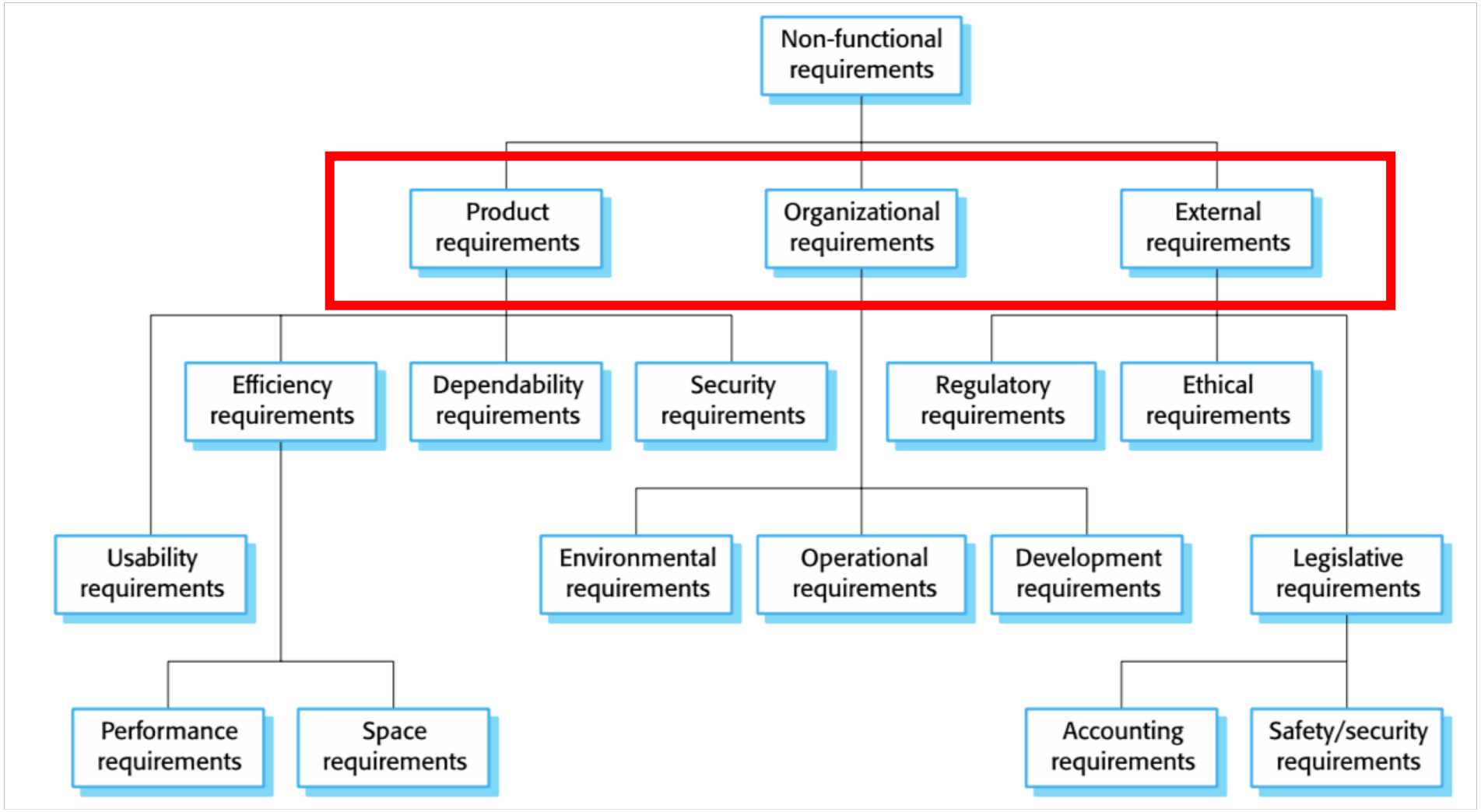
Mentcare system

- **A user shall be able to** search the appointments lists for all clinics.
- **The system shall** generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- **Each staff member using the system shall** be uniquely identified by his or her 8-digit employee number.

Types of Requirements

- หรือแบ่งตามสิ่งที่ระบุได้ 2 แบบ
  - **Functional requirements**
  - **Non-Functional requirements**
    - ระบุข้อจำกัด (constraint) ต่าง ๆ ของระบบ*โดยรวม*
    - เช่น response time, throughput, failure rate, ease of use, protocols, power consumption, storage requirement
    - แบ่งเป็น 3 ประเภท ตามแหล่งที่มา
      - Product requirement ความต้องการเกี่ยวกับผลิตภัณฑ์
      - Organizational requirement ความต้องการภายในองค์กร
      - External requirement ความต้องการจากภายนอกองค์กร

ประเภทย่อยของ Non-Functional Requirement



ตัวอย่าง Non-Functional Requirements

Product requirement

The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

Organizational requirement

Users of the Mentcare system shall authenticate themselves using their health authority identity card. (Some nonfunctional requirements inform the need for specific functional requirements)

External requirement

The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Common Problem with Non-Functional Req.

- The users or managers may write unclear goals that are difficult to test, such as

  The system should be very easy to use by medical staff and should be organized in such a way that user errors are minimized

- We can rewrite the above requirement as follows, so that the goal can be objectively verified:

  Medical staff shall be able to use all the system functions after four hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use.

Metrics for specifying non-functional req.

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Requirement Engineering Process

- การทำ RE มีกระบวนการอยู่ 4 ขั้นตอนหลัก ๆ โดยในทางปฏิบัติมักทำเป็นแบบ iterative มีดังนี้
  - Requirement elicitation*
  - Requirement analysis and specification
  - Requirement validation
  - Requirement management

* To elicit true feeling from someone = ทำให้นาย ก แสดงความรู้สึกที่แท้จริงออกมา
To elicit requirements from the customer = ล้วงเอาความต้องการที่แท้จริงจากลูกค้ามา

Iterative Requirement Engineering Process

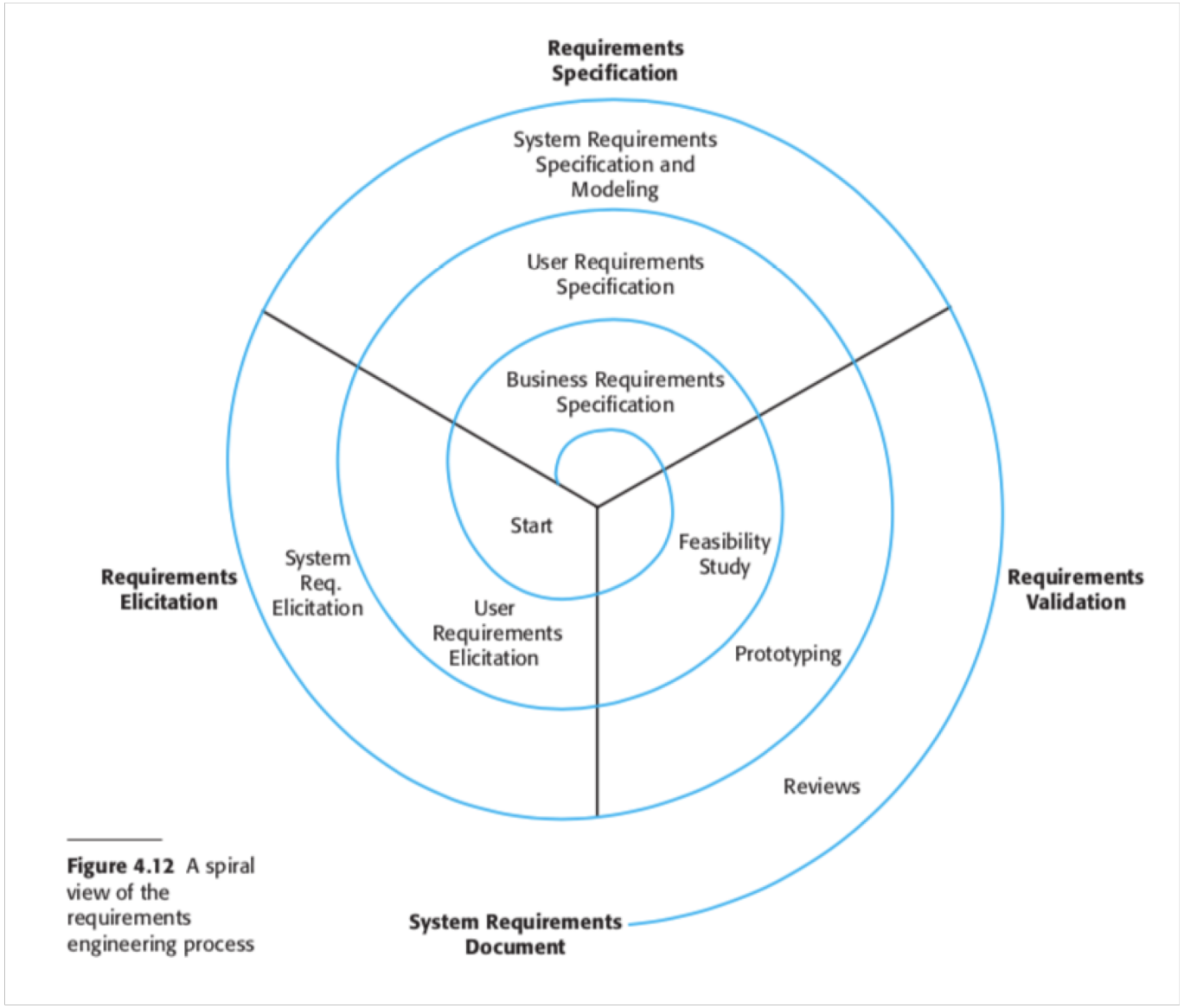


Figure 4.12 A spiral view of the requirements engineering process

Requirement Engineering Process

- **Requirement elicitation**
  - สอบถามความต้องการจาก stakeholder โดยใช้ user stories, use case เป็นเครื่องมือ อาจใช้วิธีสัมภาษณ์, ค้นหาข้อมูล, เยี่ยมชมสถานที่, สังเกตการทำงาน, จัดทำ prototype, ทำแบบสอบถาม
- **Requirement analysis and specification**
  - วิเคราะห์แล้วลงมือเขียนทั้ง User requirement และ System requirement
- **Requirement validation**
  - ตรวจสอบว่า requirement ตรงความต้องการลูกค้าจริง, ไม่มี conflict ใด ๆ ระหว่างแต่ละ requirement, มีความเป็นไปได้, สามารถทดสอบได้
- **Requirement management**
  - ติดตาม requirement ที่ทำสำเร็จ, การเปลี่ยนแปลงของ requirement

ลองคิดดูว่าใน **scrum** เราได้ทำกระบวนการเหล่านี้อย่างไร โดยใช้ **artifact** ใดบ้าง

Software Requirements Document

- Requirement documents เป็นเอกสารทางการที่ควรจัดทำขึ้นเพื่อชี้แจงสิ่งที่ผู้พัฒนาจะต้องลงมือทำโดยจะประกอบด้วยทั้งในส่วนของ User Requirement และ System Requirement
- มีรายละเอียดมากน้อยขึ้นกับงาน เช่น
  - ซอฟต์แวร์สำหรับควบคุม engine เครื่องบินคงต้องละเอียดมากๆ
  - งานที่จ้างอีกบริษัททำ (outsource) ก็ควรจะละเอียดและเป๊ะ เมื่อเทียบกับงานที่ทำเอง
- ในการเขียน อาจใช้ format ที่เป็นที่ยอมรับ เช่น ตามมาตรฐาน IEEE

ตัวอย่างโครงสร้างเอกสาร Requirement ตามมาตรฐาน IEEE

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The non-functional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.
System requirements specification	This should describe the functional and non-functional requirements in more detail. If necessary, further detail may also be added to the non-functional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components, the system, and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.