Last Update : 11/01/2568

OOAD & UML

วิชาวิเคราะห์และออกแบบระบบเชิงวัตถุ

Object-Oriented Analysis and Deisgn

Lec03 : Unified Process

ผู้ช่วยศาสตราจารย์ .ดร. นัฐพงศ์ ส่งเนียม
http://www.siam2dev.com
siam2dev@hotmail.com

สาขาวิชา วิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏพระนคร

แหล่งข้อมูลเพิ่มเติม : : http://www.lumpaya.com/sdlc01.htm

ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่งเนียม

- **Website : http://www.siam2dev.com**
- **E-mail1 : dr.nattapong_s@hotmail.com**
- **E-mail2 : siam2dev@gmail.com**
- **E-mail3 : xnattapong@gmail.com**
- **Facebook : siam2dev@hotmail.com**

Agenda

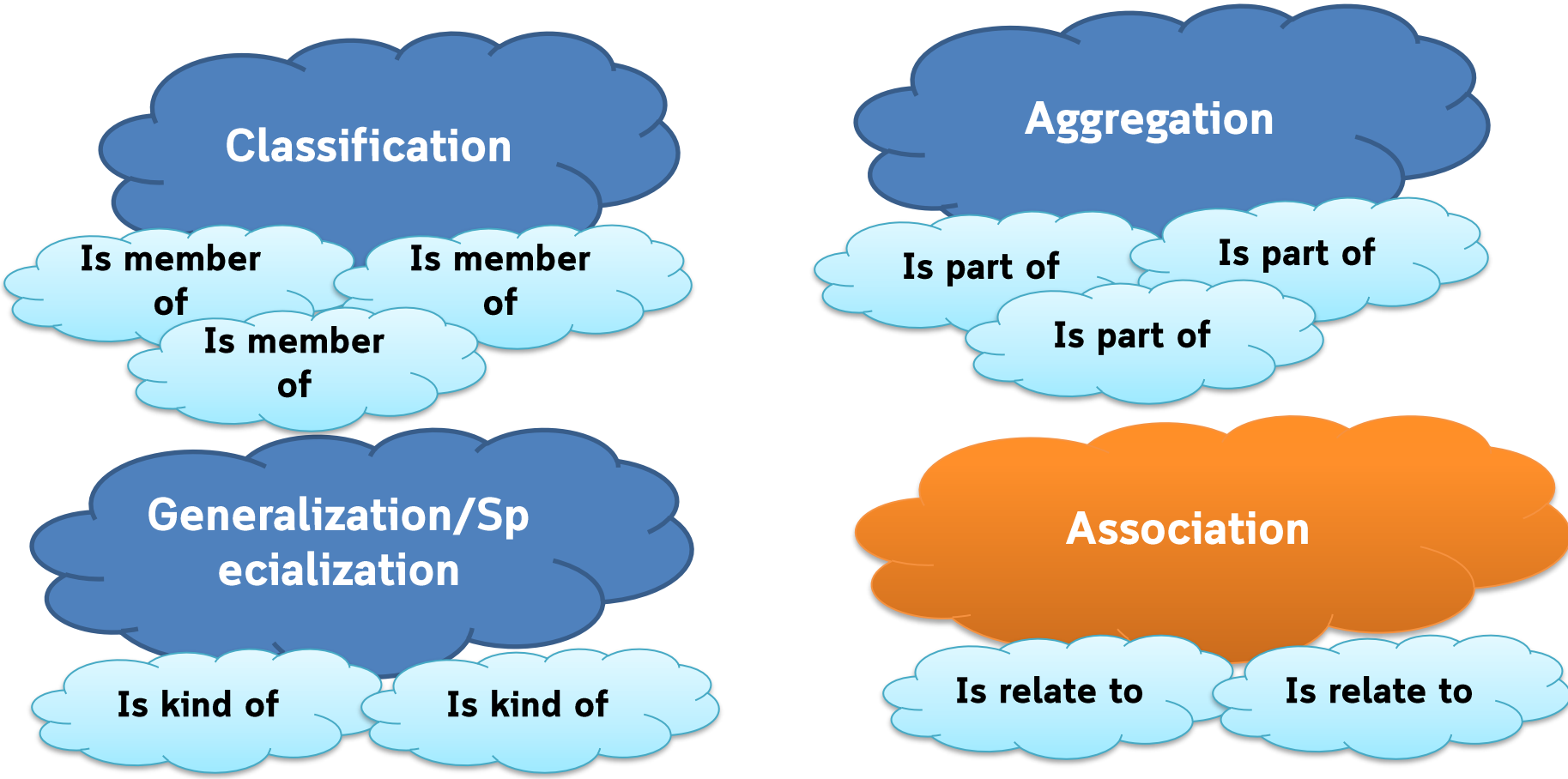
Reviews

Unified Process

1. ซอฟต์แวร์และกระบวนการพัฒนาซอฟต์แวร์ (Software Development Process)
2. กระบวนการ Unified Process
3. รูปแบบของ Unified Process
4. รอบและเฟสใน Unified Process
5. โครงสร้างกระบวนการ Unified Process
6. กระแสงานใน Unified Process

Reviews

แนวคิดเชิงนามธรรม (Abstractions)



Reviews

แนวทางการพัฒนาระบบ

Traditional การพัฒนาระบบแบบดั้งเดิม

SDLC
7 ขั้นตอน

DFD
ERD
NF

OO(Object-Oriented)

UP
4 Phase

UML

ข้อสอบระหว่างภาค 2/2567

- บทที่ 1 – 4
- มีประมาณ 5 – 6 ใหญ่
  - คำศัพท์ที่สำคัญ 10 ข้อ
  - เขียน concept ความคิดรวบยอด
  - การเขียนแผนภาพ abstraction แบบต่างๆ
  - Unified Process
  - UML
- ทั้งหมด 60 คะแนน คิดเป็น 20%
- ข้อสอบแบบอัตนัย ข้อเขียน
  - เป็นข้อสอบเชิงวิเคราะห์
  - จะไม่ถามตรง ๆ
  - เวลาสอบ เวลาทั้งหมด 2 ชั่วโมง

สอบที่ ห้อง ...
เริ่ม 09.00 – 11.30 น.

อนุญาตให้นำเอกสารที่เป็นกระดาษ เข้าห้องสอบได้เท่านั้น ไม่อนุญาตให้เปิดไฟล์

ไม่อนุญาตให้ออกนอกห้องสอบ ในขณะสอบ

3.1 What is the Unified Process ?

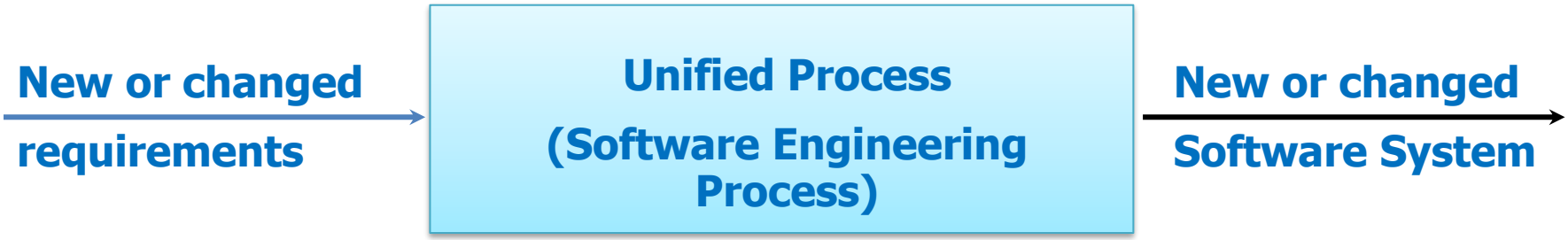
ความหมายของ Unified Process

ความหมาย Unified Process หมายถึง กรรมวิธีพัฒนาซอฟต์แวร์ (Software Development Process) วิธีการพัฒนาซอฟต์แวร์ อธิบายแนวทางสำหรับจัดสร้าง ติดตั้ง การทดสอบ และอาจรวมถึงการบำรุงรักษาซอฟต์แวร์

UP:Unified Process เป็นกรรมวิธีพัฒนาซอฟต์แวร์ที่ได้รับความนิยมสำหรับการพัฒนาระบบซอฟต์แวร์ OO:object-oriented

Unified Process อธิบายการจัดสรรงานและความรับผิดชอบให้กับทีมงานไว้อย่างชัดเจนว่าใคร จะทำอะไร เมื่อไร และทำอย่างไร เพื่อให้เกิดความมั่นใจว่าจะได้ซอฟต์แวร์ที่มีคุณภาพ ตรงกับความต้องการของลูกค้า และการพัฒนาซอฟต์แวร์ตามหลักการของ UP อยู่ภายใต้งบประมาณและค่าใช้จ่ายที่ได้ประมาณการไว้

*** สามารถควบคุมค่าใช้จ่ายในโครงการได้



*** UP เป็นหนึ่งในกระบวนการพัฒนาซอฟแวร์ หรือ วิศวกรรมซอฟต์แวร์

3.1 ความหมายของ Unified Process

Unified Process คือกรรมวิธีพัฒนาซอฟต์แวร์ (Software Development Process) หรือพัฒนาระบบ วิธีการพัฒนาซอฟต์แวร์ตามหลักการ UP จะอธิบายแนวทางสำหรับจัดสร้าง ติดตั้ง และอาจรวมถึงการบำรุงรักษาซอฟต์แวร์ Unified Process เป็นกรรมวิธีพัฒนาซอฟต์แวร์ที่ได้รับความนิยมโดยเฉพาะสำหรับการพัฒนาระบบซอฟต์แวร์แบบ object-oriented

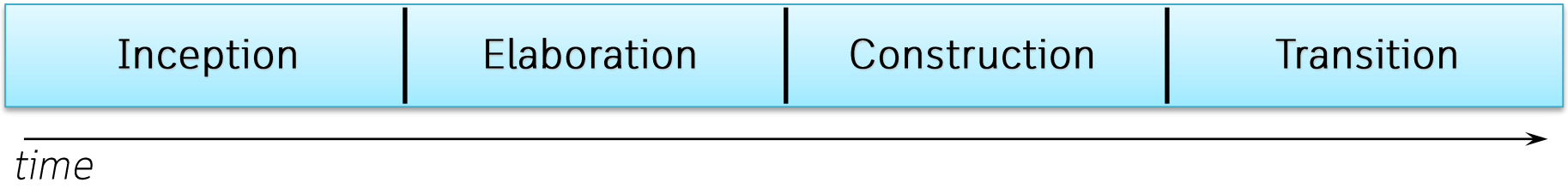
โดย Unified Process จะอธิบายการจัดสรรงานและความรับผิดชอบให้กับทีมงานไว้อย่างชัดเจนว่าใคร จะทำอะไร เมื่อไร และทำอย่างไร เพื่อให้เกิดความมั่นใจว่าจะได้ซอฟต์แวร์ที่มีคุณภาพ ตรงกับความต้องการของลูกค้า และการพัฒนาซอฟต์แวร์อยู่ภายใต้งบประมาณและค่าใช้จ่ายที่ได้ประมาณการไว้

หากสนใจ slide นี้เพื่อการเรียนการสอนโปรดติดต่อ siam2dev@hotmail.com

3.1 ความหมายของ Unified Process

Unified Process (UP) คือกระบวนการพัฒนาซอฟต์แวร์ที่เป็นกระบวนการให้คำแนะนำในการวางแผน การออกแบบ การพัฒนา และการทดสอบซอฟต์แวร์ โดยทำให้ทีมพัฒนาสามารถทำงานร่วมกันอย่างมีประสิทธิภาพ และให้ผลลัพธ์ที่มีคุณภาพสูง โดยเน้นการแยกการพัฒนาเป็นรอบๆ หรือก้าวการพัฒนา (Iteration) และเปลี่ยนแปลงไปตามความเหมาะสมของโครงการและสภาพแวดล้อมการทำงานUP ได้รับความกระจ่างใสจาก Rational Unified Process (RUP) ซึ่งเป็นกระบวนการพัฒนาซอฟต์แวร์ที่ได้รับความนิยมมาก่อนหน้านี้ โดย UP ได้ทำการปรับปรุงและแยกแยะความซับซ้อนของ RUP ให้เหมาะสมกับโครงการขนาดเล็กถึงโครงการใหญ่ข้อดีของ Unified Process คือ:- กระบวนการที่ยืดหยุ่นและสามารถปรับเปลี่ยนได้ตามความเหมาะสมของโครงการและสภาพแวดล้อม- เน้นการทำงานแบบทีม ทำให้ทีมพัฒนาสามารถทำงานร่วมกันอย่างมีประสิทธิภาพ- ให้ความสำคัญกับการตรวจสอบคุณภาพและการทดสอบซอฟต์แวร์ในขั้นตอนต่างๆ และไม่ให้กลับไปทำซ้ำขั้นตอนต่อไปหากพบข้อผิดพลาดUP ถูกออกแบบให้เหมาะสมกับโครงการที่มีความซับซ้อนและต้องการความน่าเชื่อถือในการพัฒนาซอฟต์แวร์ที่ครอบคลุมขั้นตอนต่าง ๆ และมีการควบคุมคุณภาพของผลลัพธ์ที่มีความเสี่ยงสูง และควรนำไปใช้กับโครงการที่มีทีมพัฒนามีขนาดใหญ่และมีการทำงานร่วมกันแบบแยกส่วน (distributed development) ครอบคลุมโครงการที่ใหญ่ ซับซ้อนและก้าวการพัฒนาหลายขั้นตอน โดยให้มีการควบคุมคุณภาพของซอฟต์แวร์ การวางแผนและการควบคุมที่เหมาะสมเพื่อให้สามารถสร้างซอฟต์แวร์ที่คุณภาพสูงโดยทำให้เป็นระบบหลั

3.1 โครงสร้างกรรมวิธี - Lifecycle Phases



Unified process แบ่งการพัฒนาออกเป็น 4 เฟส (phases)

- **เตรียมงาน (Inception) – กำหนดโครงการ , นิยามขอบเขตของโครงการ , ขอบเขตของระบบที่จะพัฒนา .....ทำ เขียน/นำแบบที่เรียกว่า Proposal หรือแบบเสนอโครงการ ให้กับหน่วยงานที่มอบหมาย**
- **การจัดทำรายละเอียด (Elaboration) – วางแผนโครงการ จัดทำรายละเอียดความต้องการหรือข้อกำหนดความต้องการ(Requirement Specification) จัดสร้างสถาปัตยกรรมระบบ**
- **จัดสร้าง (Construction) – สร้าง พัฒนา และทดสอบโปรแกรม / ระบบ**
- **ถ่ายโอน (Transition) – ติดตั้งถ่ายโอนระบบให้กับผู้ใช้ ลงโปรแกรม ติดตั้งระบบ การ config**

ทดสอบระบบ ระยะเวลาหนึ่ง อบรม การใช้งาน จัดทำคู่มือการใช้

*****โครงงานกลุ่ม ที่รับผิดชอบในรายวิชานี้ ให้ใช้ UP**

(Phase I: Inception)

- **เตรียมงาน (Inception)** – ในขั้นตอนนี้จะมีกิจกรรมเกี่ยวกับการวิเคราะห์ความเป็นไปได้และความเป็นไปไม่ได้ของโปรเจคท์ การกำหนดวัตถุประสงค์ของโปรเจกต์ และการระบุขอบเขตของระบบ กำหนดโครงการ ทำเรื่องอะไร , นิยามขอบเขตของโครงการ , ขอบเขตของระบบที่จะพัฒนา .....ทำ เขียน/นำแบบที่เรียกว่า Proposal หรือแบบเสนอโครงการ ให้กับหน่วยงานที่มอบหมาย

Proposal แบบเสนอโครงการ

1. ชื่อโครงการ
2. ผู้รับผิดชอบ
3. ที่มาและความสำคัญของโครงการ
4. วัตถุประสงค์
5. ขอบเขตของโครงการ
6. แผนการดำเนินงาน / ระยะในการดำเนินงาน
7. เครื่องมือ/ซอฟต์แวร์ที่ใช้
8. ประโยชน์ที่คาดว่าจะได้รับ

วางแผน
จัดทำแผนโครงการ
จัดทำแผนความเสี่ยง
กำหนดผู้รับผิดชอบ

จะต้องเขียนบทที่ 1 บทนำ

ระยะที่ 1 เตรียมงาน (Phase I: Inception)

แบบเสนอโครงการ (Proposal)

- ***บทนำ***
  1. ***หลักการและเหตุผล / ที่มา ความสำคัญของปัญหา***
  2. ***วัตถุประสงค์***
  3. ***ขอบเขต***
     1. ***ด้านประชากร***
     2. ***ด้านเนื้อหา***
     3. ***ด้านเทคนิค***
  4. ***วิธีการดำเนินงาน / แผนการดำเนินงาน (Gantt Chart)***
  5. ***เครื่องมือที่ใช้***
  6. ***กำหนดงบประมาณ***
  7. ***ประโยชน์ที่คาดว่าจะได้รับ***
  8. ***นิยามศัพท์***

มีไว้เพื่อว่างาน/โครงการที่ทำ คลอบคลุมแค่ไหน

ระยะที่ 2 ทำรายละเอียด (Phase II : Elaboration)

- **ทำรายละเอียด (Elaboration) – วางแผนโครงการ จัดทำโมเดลทางธุรกิจ (Business Model) เงื่อนไขทางธุรกิจ (Business Rule) รายละเอียดความต้องการ หรือข้อกำหนดความต้องการ (Requirement Specification), จัดสร้างสถาปัตยกรรมระบบ (System Architecture) ด้วยแผนภาพสถาปัตยกรรมของระบบ (Deployment Diagram)**

ระยะที่ 3 จัดสร้าง (Phase III : Construction)

จัดสร้าง (Phase III : Construction) – สร้างและทดสอบระบบ

- การพัฒนาระบบ ขึ้นอยู่กับลักษณะของระบบ
  - เช่น ถ้าระบบเป็นซอฟต์แวร์ ก็หมายถึงการเขียนโปรแกรมและทดสอบโปรแกรม
  - ถ้าเป็นการพัฒนาระบบทั้งฮาร์ดแวร์ และซอฟต์แวร์ จะต้องทั้ง เขียนและ จัดหาหรือจัดซื้อฮาร์ดแวร์ เช่นระบบ จัดการสินค้า อาจจะมือเครื่องอ่านบาร์โค้ด ด้วยก็ตั้งจัดซื้อ อุปกรณ์ เหล่านี้ด้วย หรือ อาจจะ RFID

ต.ย. จงพัฒนาระบบเวชระเบียนของ รพ. แห่งหนึ่งถ้าเป็นเว็บแอปพลิเคชัน ...เขียนเว็บ ...ฐานข้อมูล เช่าโฮส / เช่าเซิร์ฟเวอร์ ...จดโดเมน ด้วย รวมถึงการอัพโหลด ทดสอบ

ระยะที่ 3 จัดสร้าง (Phase III : Construction)

การพัฒนาระบบประกอบไปด้วย

- Software
- Hardware
- Network
- Peripheral อุปกรณ์ต่อพวง ได้แก่ scanner , printer , barcode reader , QR-code , RFID , IoT

ระบบจัดสต๊อก โดย ใช้บาร์โค้ด หรือ RFID

พัฒนาระบบที่อ่านข้อมูลไฟล์ จากกระดาษเข้าคอมพิวเตอร์โดยอัตโนมัติ อ่านจาก Scanner

ระบบรักษาความปลอดภัย ด้วยกล้อง CCTV

ระยะที่ 4 ถ่ายโอน (Phase IV: Transition)

ถ่ายโอน (Phase IV: Transition) – ติดตั้งถ่ายโอนระบบให้กับผู้ใช้

- เมื่อระบบสร้างเสร็จแล้ว จะต้องทำการติดตั้ง config เช่น ถ้าระบบสำหรับทำงานหลายคน หรือ client-server , 2-Tier , N-Tier

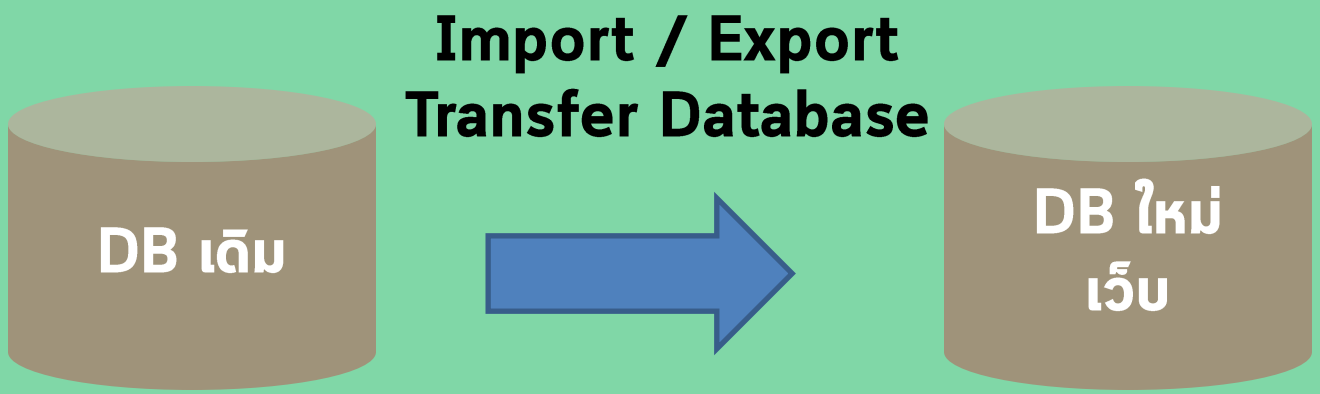
จะต้องมีการ Config , network กำหนด IP กำหนด Proxy กำหนด Protocol , IPX/SPX , TCP/IP, subnet mark

- จัดทำคู่มือ
  - User Manual
  - Programmer Manual / Technical Document
- Training หรือ จัดอบรมให้กับผู้ใช้
- การทดสอบ Beta ...หลังจากติดตั้ง ความเข้ากันของฐานข้อมูลเก่าและใหม่

ระยะที่ 4 ถ่ายโอน (Phase IV: Transition)

การทดสอบ Beta ...หลังจากติดตั้ง ความเข้ากันของฐานข้อมูลเก่าและใหม่

- ระบบเดิม เป็น Desktop / Windows Application ต่อมา ให้คุณพัฒนาเป็น
- Web Application ซึ่งจะต้องเป็นระบบออนไลน์



วงจรชีวิตของ Unified Process

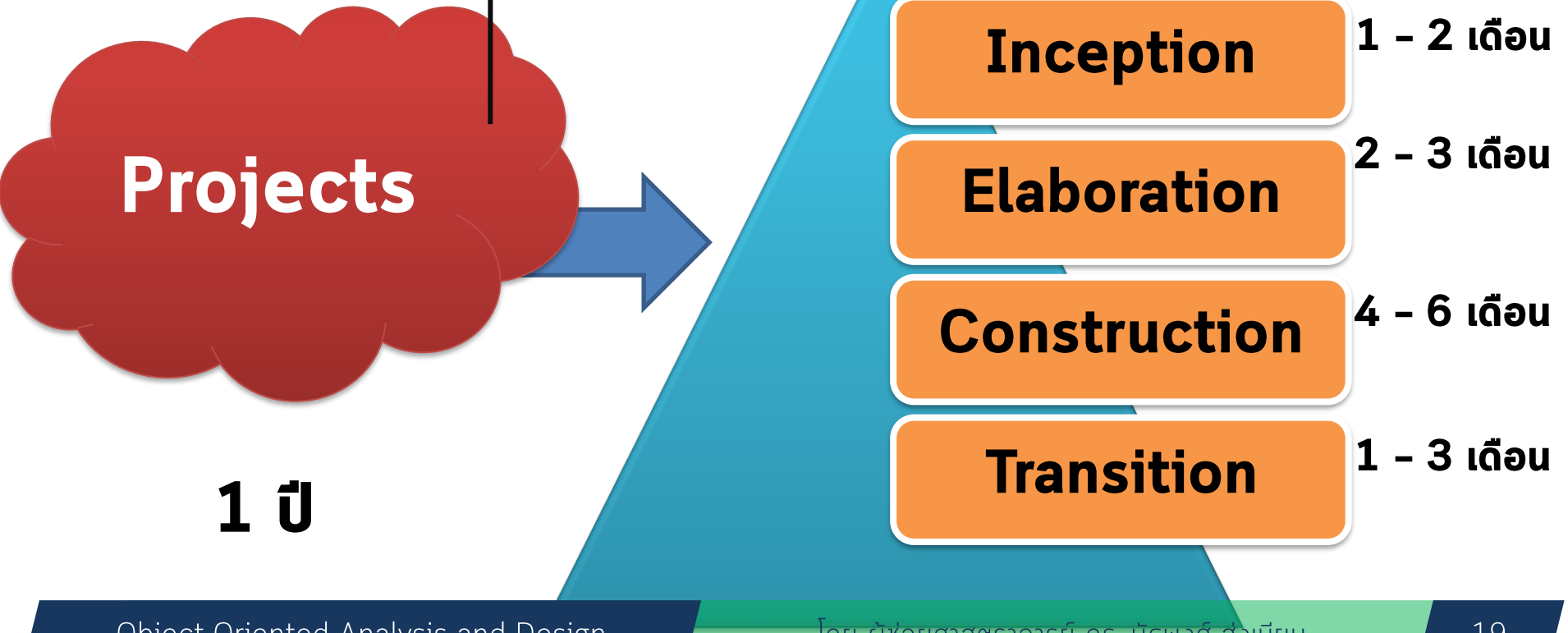
วงจรชีวิตของ Unified Process

ตัวอย่าง

PM: Project manager

- **องค์กร /ภายใน (Inhouse)**
  เจ้าของ บ. เป็นผู้มอบหมายงานให้
- **หน่วยงานภายนอก**
  รับงาน/จ้าง (Outsource)

In house / Out Source



Unified Process : Staff / Team

- PM: Project Manager ผู้จัดการโครงการ
- SA นักวิเคราะห์ระบบ 1- 3 คน
- Programmer / Dev นักพัฒนาโปรแกรม
  - Front-End
  - Back-Office
- Designer นักออกแบบ
  - UX/UI
  - Mockup
  - Engineer
- Tester นักทดสอบระบบ
  - Test case
  - Unit Test
  - Integrate Test

Unified Process : Staff / Team

งานกลุ่ม หลังจากสอบ ระหว่างภาค

จัดแบ่งเป็นกลุ่ม เพื่อทำการเก็บรวบรวมข้อมูล Information Gathering

- สมมติ กลุ่ม ทำเรื่อง CarCare ไปสัมภาษณ์ เก็บข้อมูล จากอีก บ. หนึ่ง
- ให้ จัดแบบห้องประชุม แบบออนไลน์ zoom , google meet
- เขียนป้าย ชื่อ บ. ชื่อพนักงาน ตำแหน่ง ทั้ง บ. ที่สัมภาษณ์
- เตรียมคำถาม – เตรียมคำตอบ
- บันทึก/ถ่ายเป็นวิดีโอ ตั้งแต่ขั้นตอนแรก
  - แนะนำรายวิชา อาจารย์ คณะ มหาวิทยาลัย
  - แนะนำสมาชิก
  - แนะนำ ชื่อกลุ่มงาน
  - สัมภาษณ์
  - สิ้นสุด

Unified Process : Staff / Team

In house / Outsource

- In house ทีมงาน ที่อยู่ภายใน บ. /องค์กร นั้น อาจคนเดียวหรือทั้งแผนก
- Outsource บุคคลภายนอก ที่จ้างขึ้นมาที่ทำเฉพาะบางโครงการ

********** จงเปรียบเทียบข้อดีข้อเสีย

3.3 รูปลักษณ์ของ Unified Process

ลักษณะหรือ หรือ หน้าตาของ UP

- เป็นกรรมวิธีวนรอบเพิ่มพูน (An Iterative and Incremental Process)
- เป็นกรรมวิธีที่มี Use-Case เป็นตัวขับเคลื่อน (A Use-Case Driven Process)
- เป็นกรรมวิธีที่มีสถาปัตยกรรมเป็นศูนย์กลาง (An Architecture-Centric Process)

3.3 รูปลักษณ์ของ Unified Process

Unified Process ขับเคลื่อนด้วย Use Case (Use Case Driven)

Use Case เป็นฟังก์ชันการทำงานในระบบ ซึ่งการทำงานนั้นให้ผลลัพธ์ที่มีคุณค่าต่อผู้ใช้

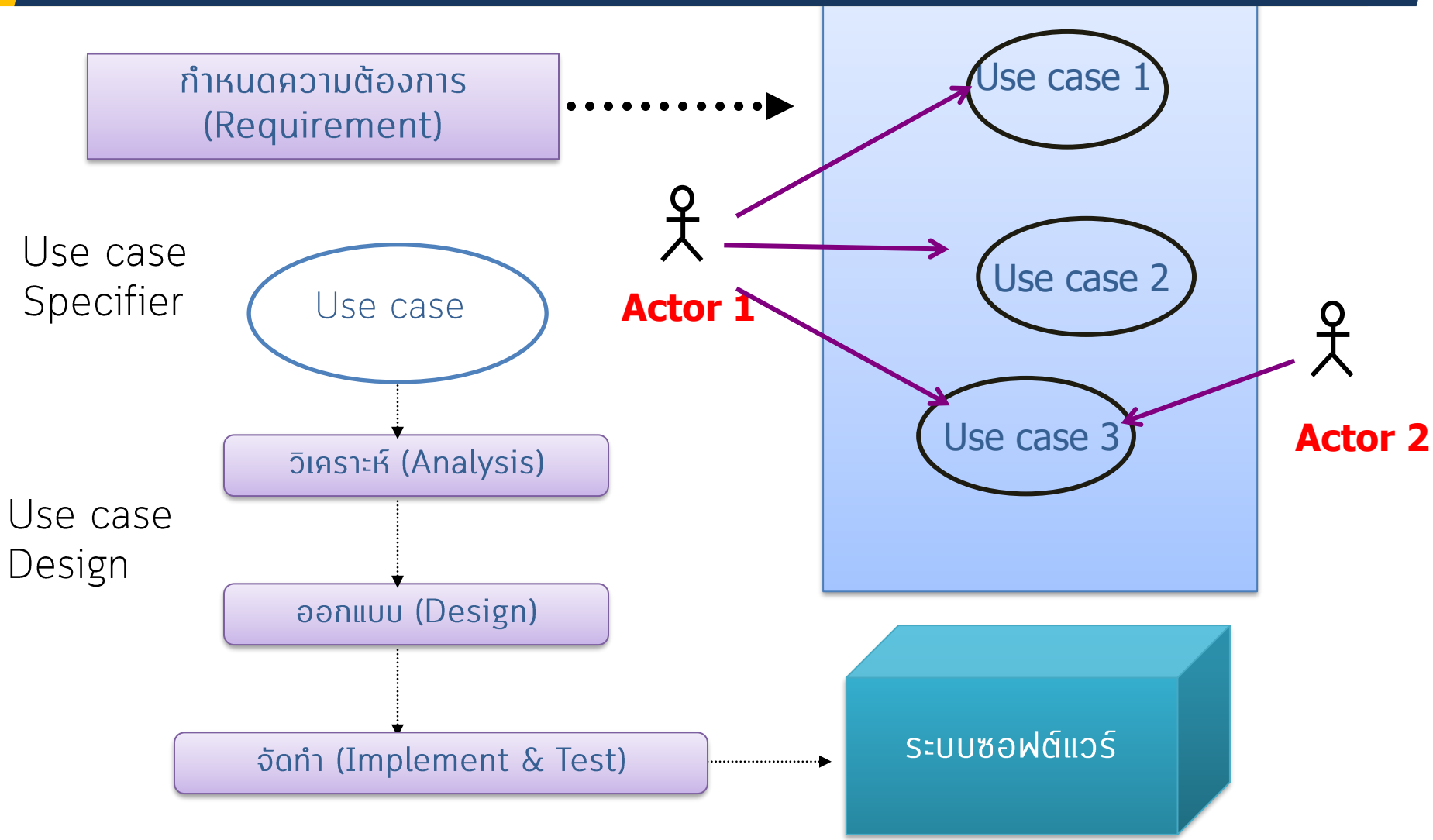
แต่ละ Use Case เป็นการทำงานที่ผู้ใช้คาดหวังเป็นความต้องการของผู้ใช้

Use Case Model อธิบายการทำงานทั้งหมดที่มีในระบบ

เนื่องจากความต้องการคือ use case การพัฒนาระบบจึงต้องกระทำกับ use case ดังนั้น use case เป็นตัวขับเคลื่อนการออกแบบ การจัดสร้าง และการทดสอบระบบ

use case driven หมายถึงการพัฒนาระบบทำตามขั้นตอนโดยใช้ use case เป็นหลัก เริ่มจากหาความต้องการระบุเป็น use case จากนั้นทำ use case ให้เป็นจริง โดยนำ use case ไปวิเคราะห์ ออกแบบ สร้างเป็น code แล้วทำการทดสอบว่า code ที่สร้างขึ้นทำงานได้ถูกต้องตามที่ระบุไว้ใน use case

3.3 รูปลักษณ์ของ Unified Process



3.3 รูปลักษณ์ของ Unified Process

Unified Process ทำงานเป็นวนรอบเพิ่มพูนผล

(Iterative and Incremental)

โครงการพัฒนาระบบถูกจัดออกเป็นหลายโครงการย่อย

Project 1.1

Project 1.2

Project 1.3

แต่ละโครงการย่อยมีกำหนดเวลา แผนงานของตัวเอง เรียกแต่ละ โครงการย่อยว่า รอบงาน (Iteration)

แต่ละรอบงาน (Iteration) กระทำการพัฒนาระบบตามขั้นตอนวิเคราะห์ ออกแบบ จัดสร้างของตัวเอง ได้ผลลัพธ์เป็นระบบที่นำไปประมวลผลได้

ระบบจะถูกเพิ่มพูนให้โตขึ้นตลอดเวลาตามรอบงานแต่ละรอบงานที่ทำเสร็จ จนกระทั่งได้ระบบที่สมบูรณ์ เป็นการพัฒนาระบบที่เรียกว่าวนรอบเพิ่มพูนผล (Iterative and Incremental Development)

3.3 รูปลักษณ์ของ Unified Process

Unified Process ทำงานเป็นวนรอบเพิ่มพูนผล

(Iterative and Incremental)

Iterative and Incremental

ตัวอย่าง จงพัฒนาระบบ MIS ของโรงพยาบาล

Form 30-40 Forms

Project 1

Project 1.1

ระบบเวชระเบียน

Project 1.2

HRM

Project 1.3

Purchase Orders

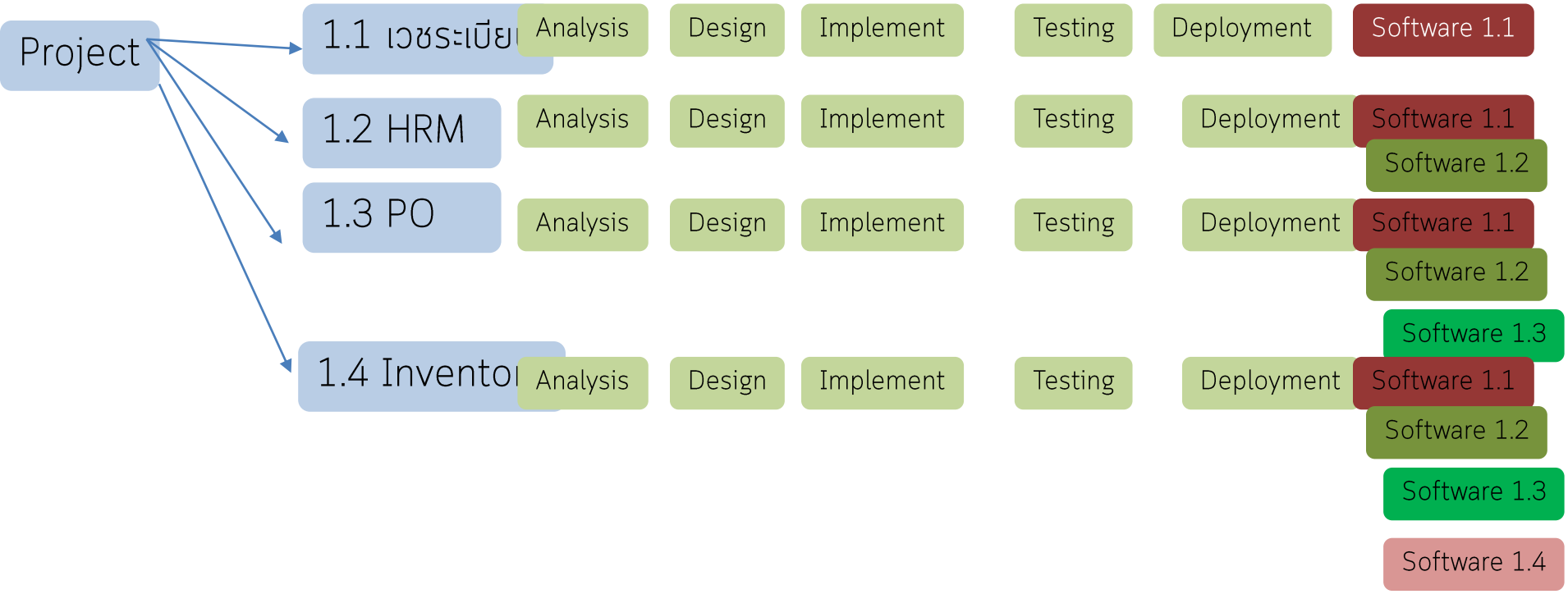
Project 1.4

Inventory Control

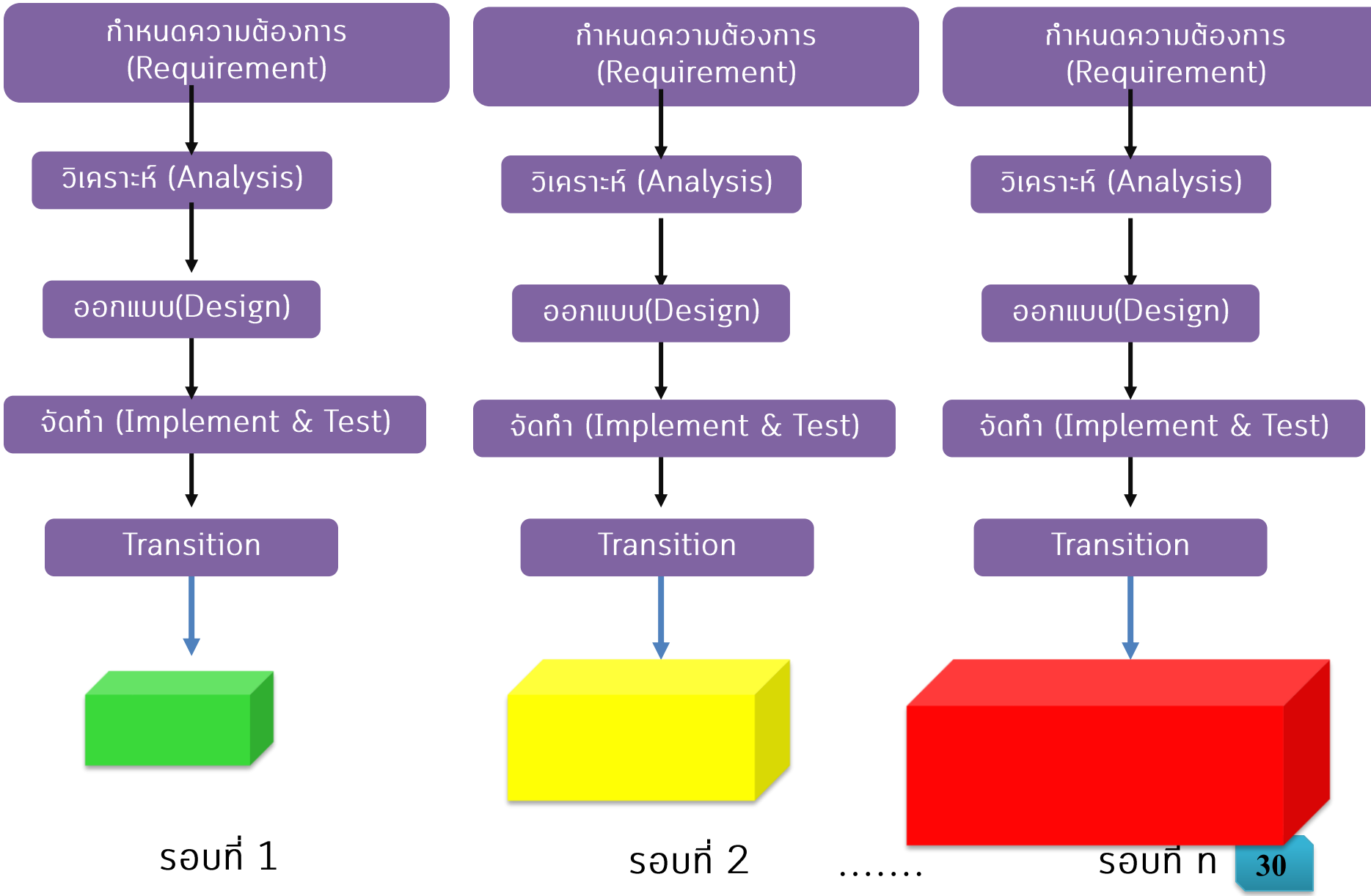
Database 20-30 Table

Report 20-30 Reports

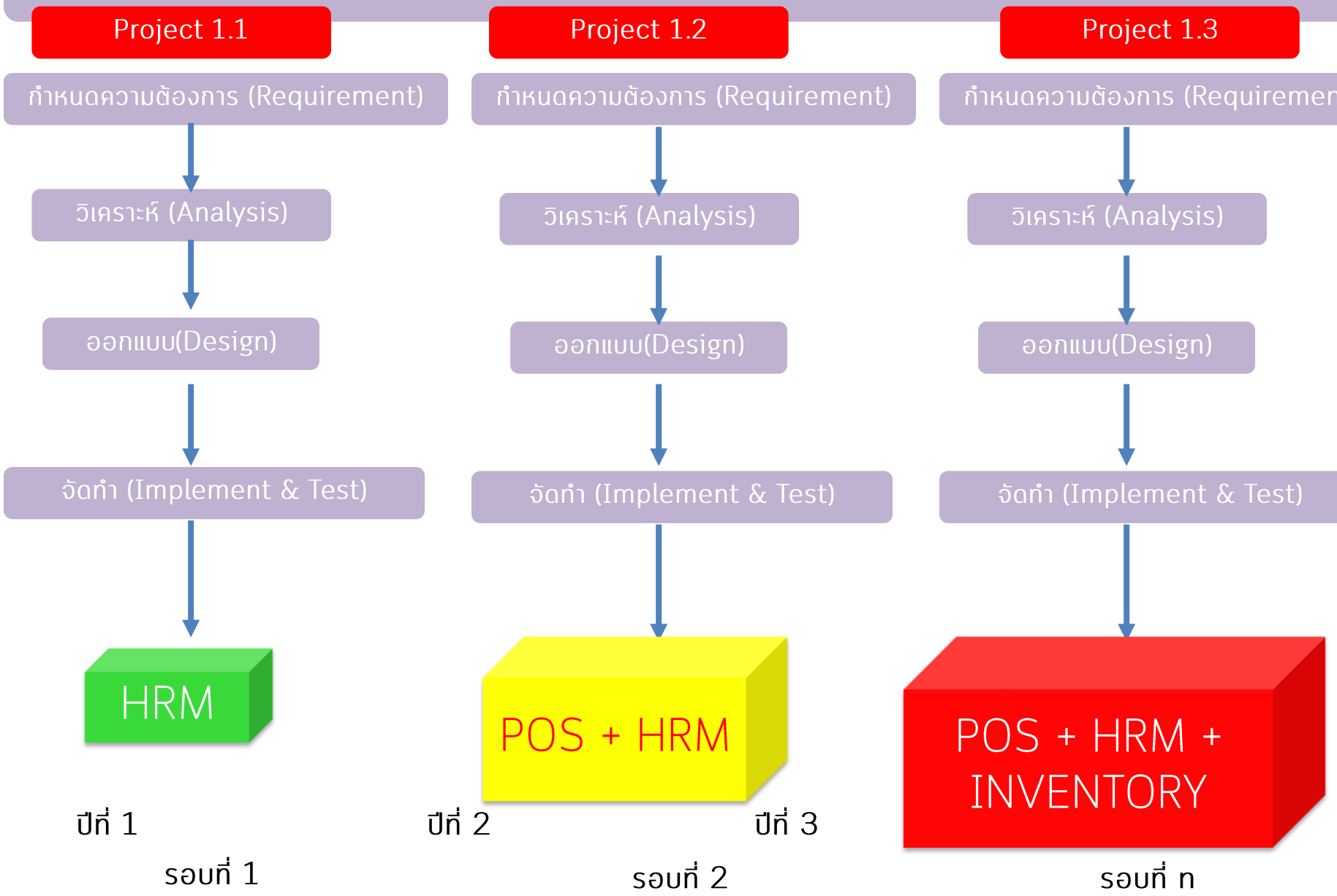
Iterative and Incremental



Iterative and Incremental



Iterative and Incremental Model

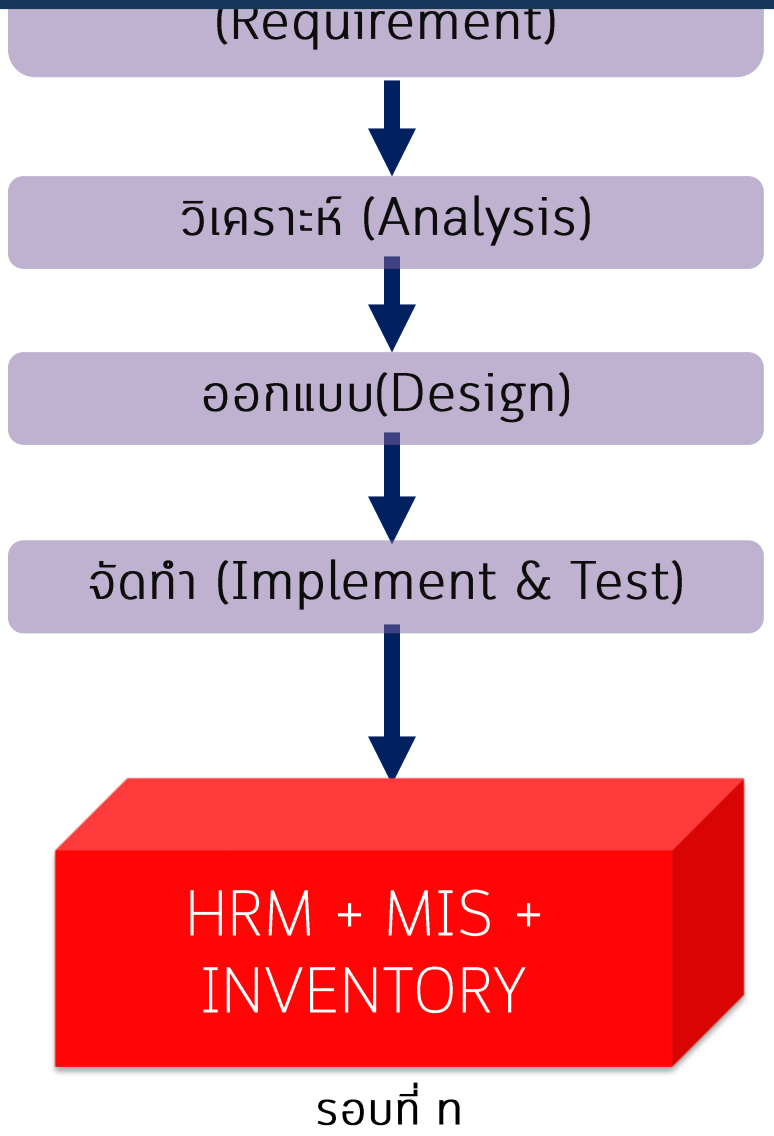


3.3 รูปลักษณ์ของ Unified Process

ทำไมต้องพัฒนาแบบวนรอบเพิ่มพูนผล

- เพื่อจัดการกับความเสี่ยงที่มีผลกระทบสูงตั้งแต่ช่วงต้น
- เพื่อสร้างสถาปัตยกรรมระบบไว้ก่อน เป็นแนวทางในการพัฒนาระบบ
- เป็นกรอบงานที่สามารถจัดการกับการเปลี่ยนความต้องการที่ไม่สามารถเลี่ยงได้เป็นอย่างดี
- เป็นการสร้างระบบโดยการเพิ่มพูนผลตลอดเวลาในแต่ละรอบงาน แทนที่จะทำระบบในครั้งเดียวให้เสร็จสมบูรณ์ ตอนใกล้จบโครงการ ซึ่งถ้ามีการเปลี่ยนแปลงเกิดขึ้นจะมีค่าใช้จ่ายมาก
- เป็นการจัดกรรมวิธีที่ทำให้ทีมงานทำงานได้อย่างมีประสิทธิภาพ

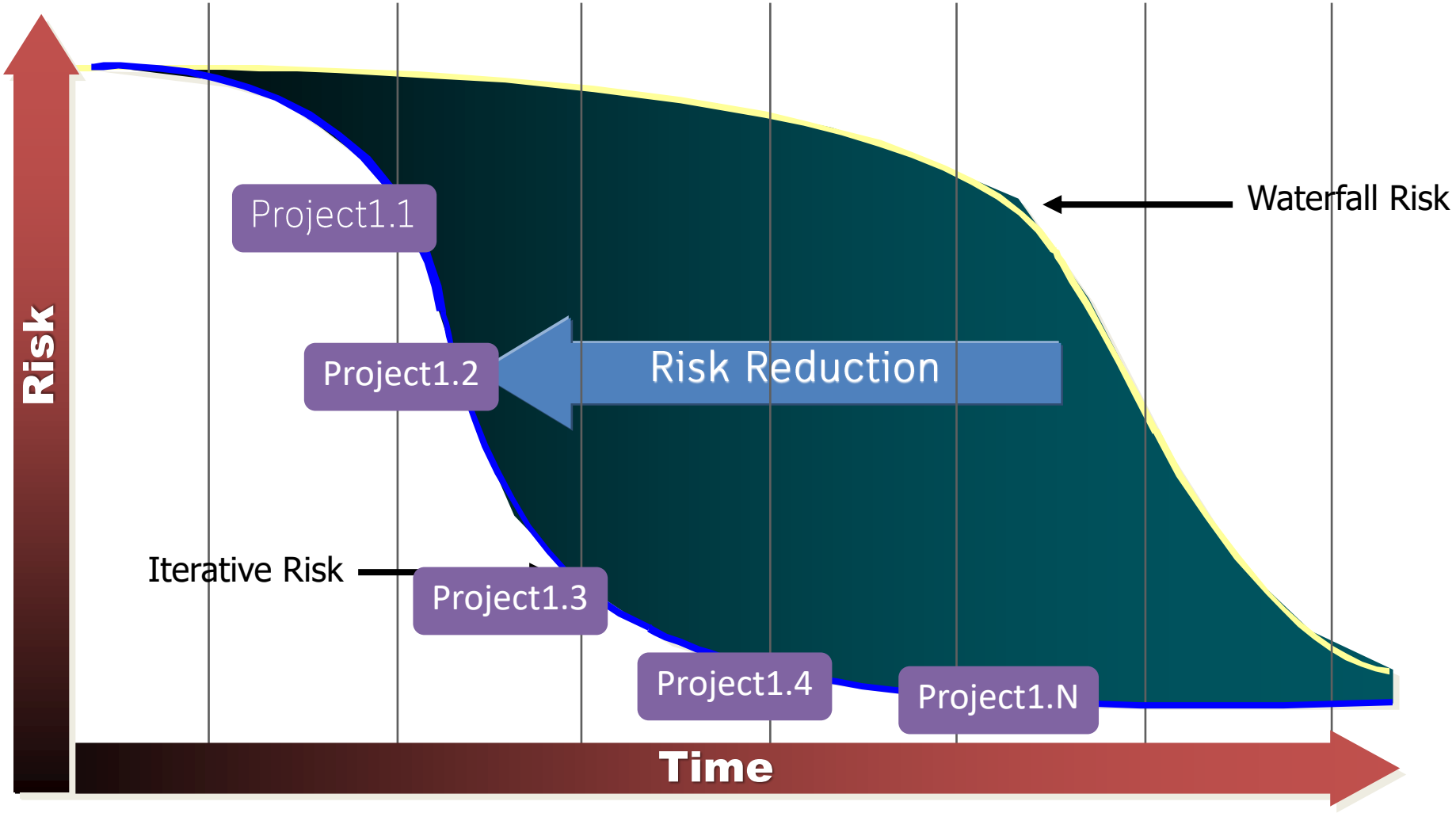
3.3 รูปลักษณ์ของ Unified Process



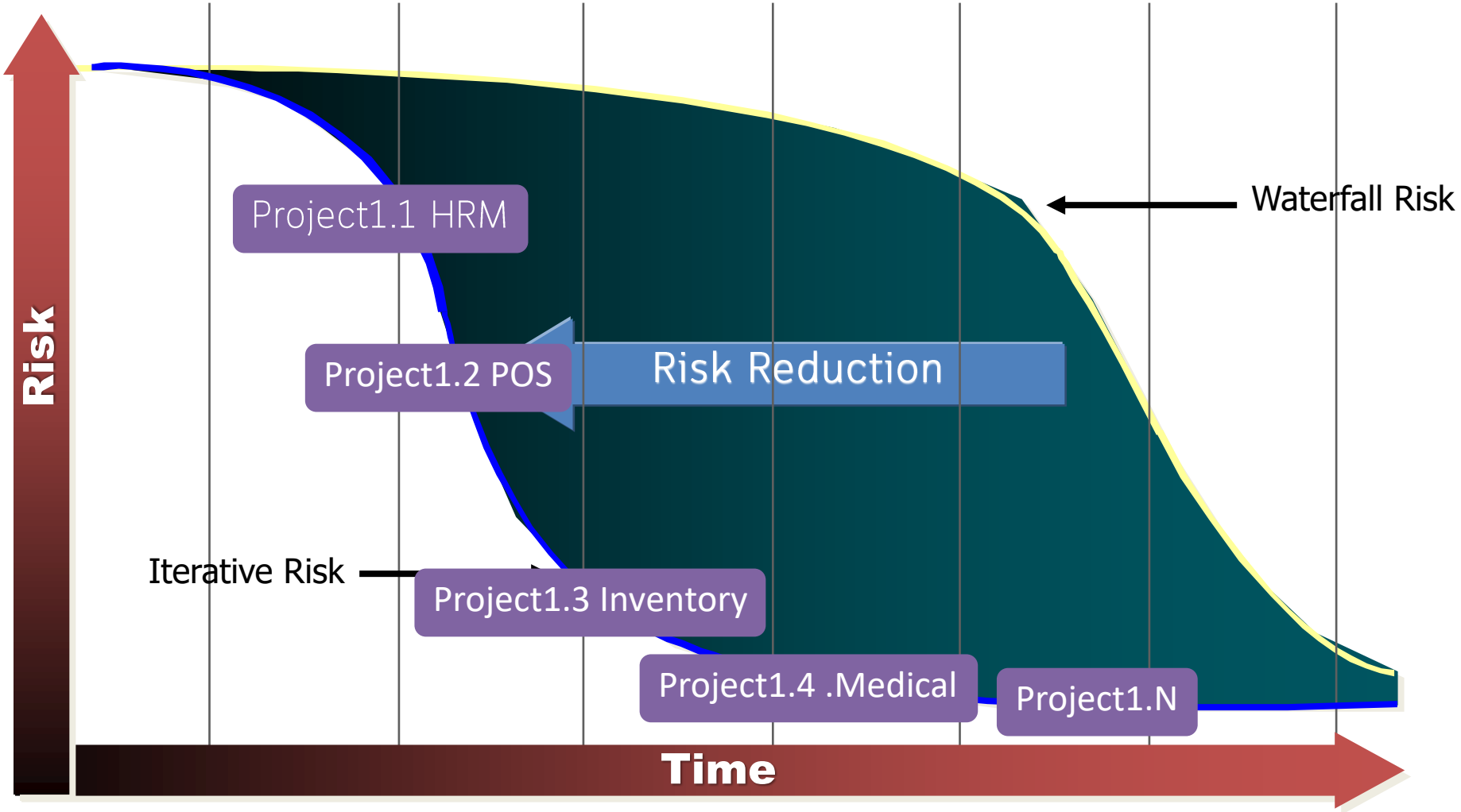
Waterfall Model develop whole system in the once process

5 ปี

Risk Profile



Risk Profile : ระบบบริหารจัดการโรงพยาบาล



3.3 รูปลักษณ์ของ Unified Process

ก่อนที่จะพัฒนาระบบใด ๆ ก็ตาม จำเป็นต้องมีการศึกษาความเป็นไปได้

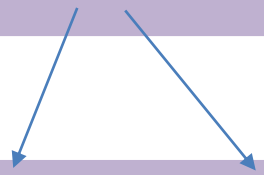
ซึ่งการศึกษาความเป็นไปได้ มี 3 ด้าน ดังนี้

1. **ด้านเทคนิค โปรแกรมที่ใช้ เทคนิค ที่ใช้ ทำได้หรือไม่ มีความเป็นได้หรือไม่**
2. **ด้านการปฏิบัติงาน เมื่อนำไปใช้แล้วสะดวกหรือไม่**
3. **ด้านเศรษฐศาสตร์ งบประมาณ**

จงพัฒนาระบบจองคิวในการทำธุรกรรมธนาคารผ่านมือถือ

แต่ก่อนทำไม่ได้ เพราะเทคโนโลยีมือถือยังไม่สะดวกเท่าปัจจุบัน

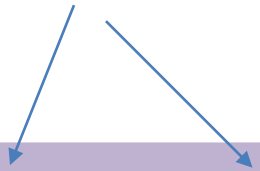
จงศึกษาความเป็นไปได้ทางเทคนิค ....ผลลัพธ์ คือ ทำได้หรือไม่
เพราะอะไร ทำอย่างไร
แก้ปัญหาอย่างไร (5 คะแนน)



- **ใช้ภาษา อะไร เครื่องมืออะไร ....มีค่า license ?**
- **สมมุติใน ทีมงานมีแต่ android .... ไม่มีโปรแกรมเมอร์ iOS เลย**

จงพัฒนาระบบจองคิวในการทำธุรกรรมธนาคารผ่านมือถือ

จงศึกษาความเป็นไปได้ทางเทคนิค ....ผลลัพธ์ คือ ทำได้หรือไม่ เพราะอะไร ทำอย่างไร
แก้ปัญหาอย่างไร (5 คะแนน)



จงศึกษาความเป็นไปได้ทางปฏิบัติงาน ....ผลลัพธ์ คือ ทำได้หรือไม่ เพราะอะไร ทำอย่างไร
แก้ปัญหาอย่างไร (5 คะแนน)

ทำแอปพลิเคชันซื้อข้าวราดแกงในโรงอาหาร มีความเป็นได้หรือไม่ ในแง่ .....เทคนิค การปฏิบัติงาน เศรษฐศาสตร์

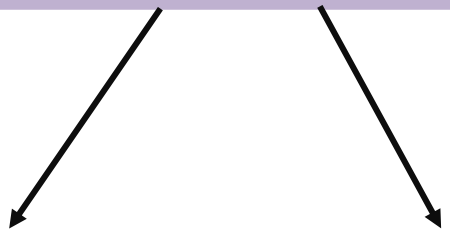
มีแม่ค้าคนเดียว แต่ สั่งทั้ง มหาวิทยาลัย ?

ทางเศรษฐศาสตร์ เช่น โปรแกรม ราคา 200000 แต่ราคาข้าวจานละ 30 บาท คุ้มหรือไม่ ต้องใช้วิธีทางเศรษฐศาสตร์คำนวณ

- Grab
- GET
- Panda Food
- Line Man

จงพัฒนาระบบจองคิวในการทำธุรกรรมธนาคารผ่านมือถือ

จงศึกษาความเป็นไปได้ทางเทคนิค ....ผลลัพธ์ คือ ทำได้หรือไม่ เพราะอะไร ทำอย่างไร แก้ปัญหาอย่างไร (5 คะแนน)

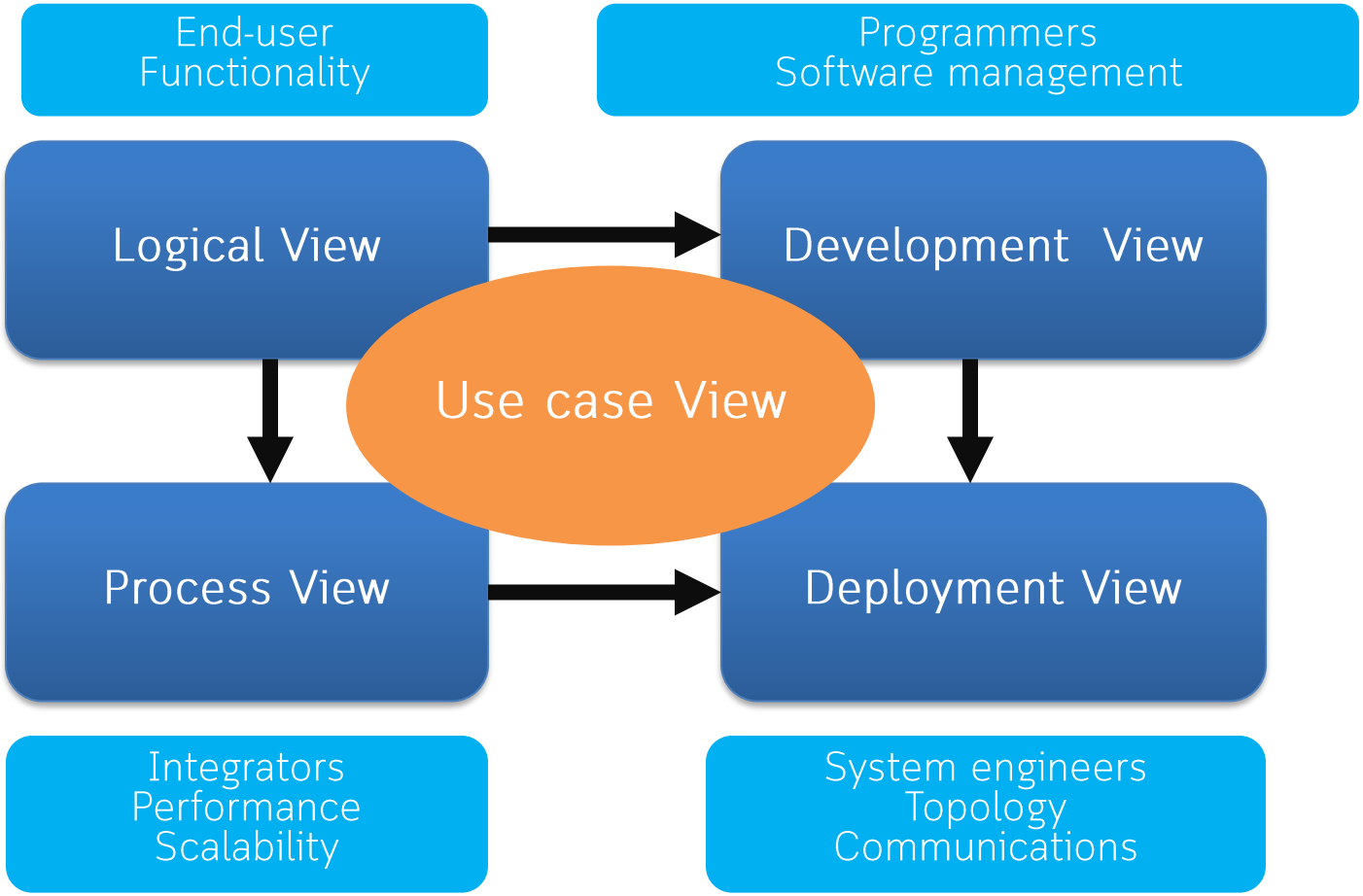


- สมมุติ 10000 ทำได้หรือไม่ ? ต้องคิดวิเคราะห์ประเมินค่าใช้จ่ายเป็น ....แบ่งเป็นค่าอะไรบ้าง ...
- ใช้ inhouse หรือ outsource
- มี network / server หรือไม่
- ใช้ OS อะไร
- ใช้ plate form อะไร
- ใช้ framework อะไร

3.3 รูปลักษณ์ของ Unified Process

Unified Process มีสถาปัตยกรรมเป็นศูนย์กลาง (Architecture-Centric)

UP ใช้สถาปัตยกรรมเป็นศูนย์กลางในการพัฒนาระบบ แสดงภาพของระบบในมุมมองต่างๆ

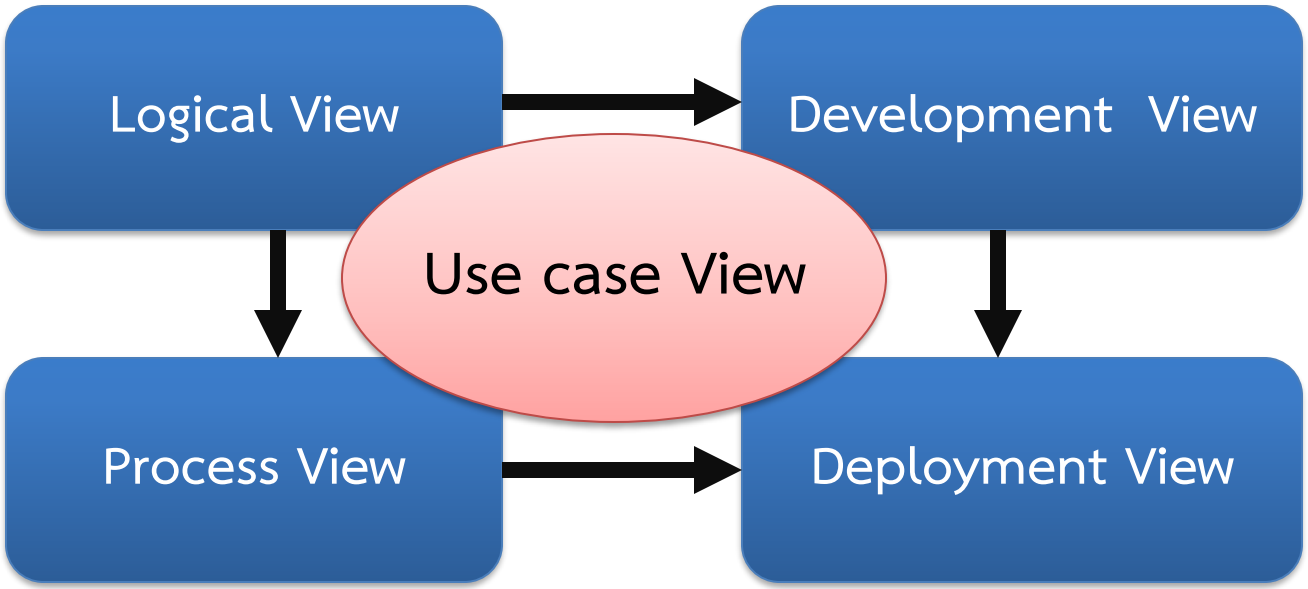


Unified Process มีสถาปัตยกรรมเป็นศูนย์กลาง (Architecture-Centric)

UP ใช้สถาปัตยกรรมเป็นศูนย์กลางในการพัฒนาระบบ แสดงภาพของระบบในมุมมองต่างๆ

End-user
Functionality

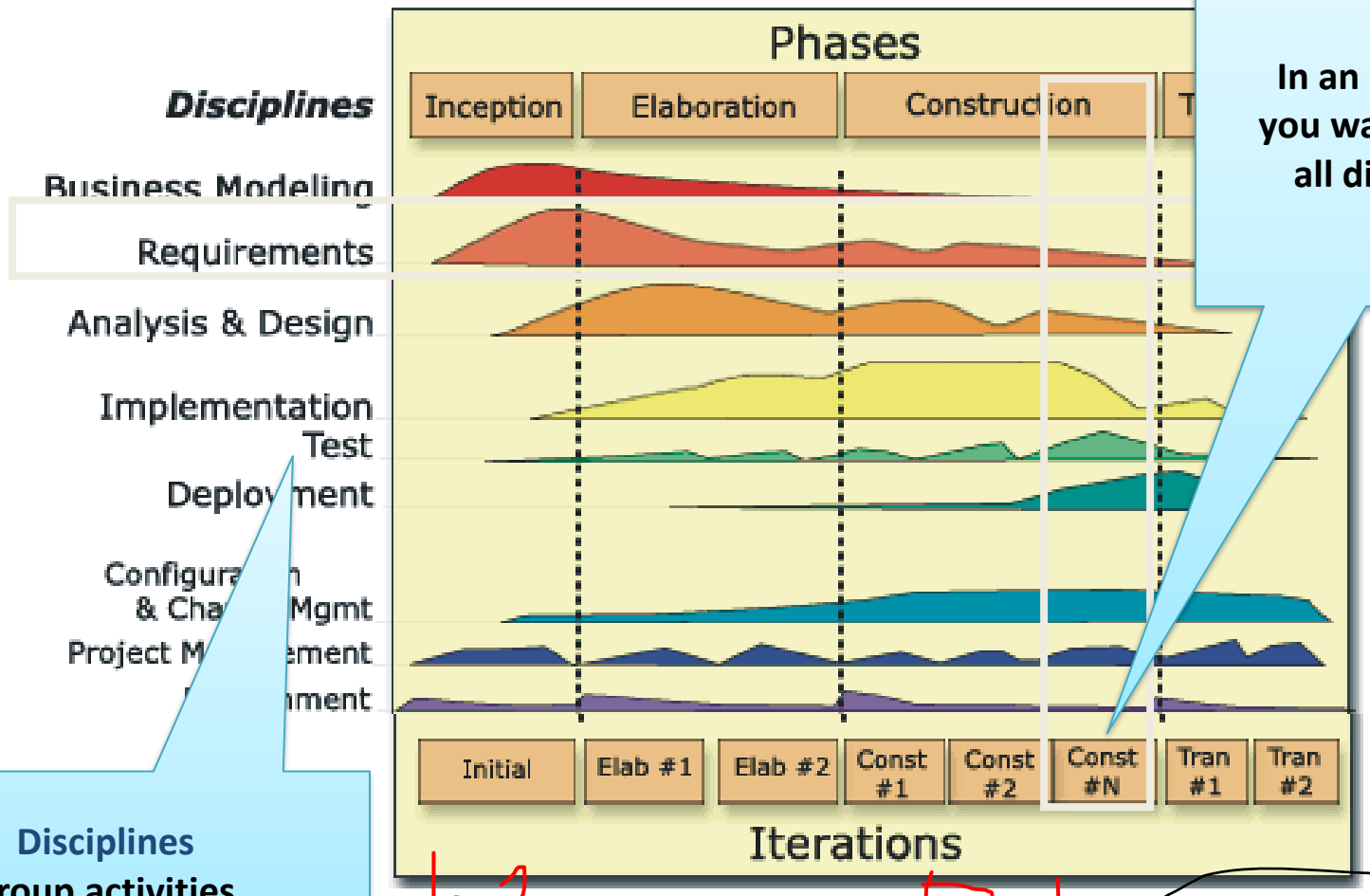
Programmers
Software management



Integrators
Performance
Scalability

System engineers
Topology
Communications

The Iterative Approach



In an iteration, you walk through all disciplines

Disciplines group activities logically

3.3 รูปลักษณ์ของ Unified Process

ระดับของซอฟต์แวร์

เวลา นับ User นับอย่างไร ?

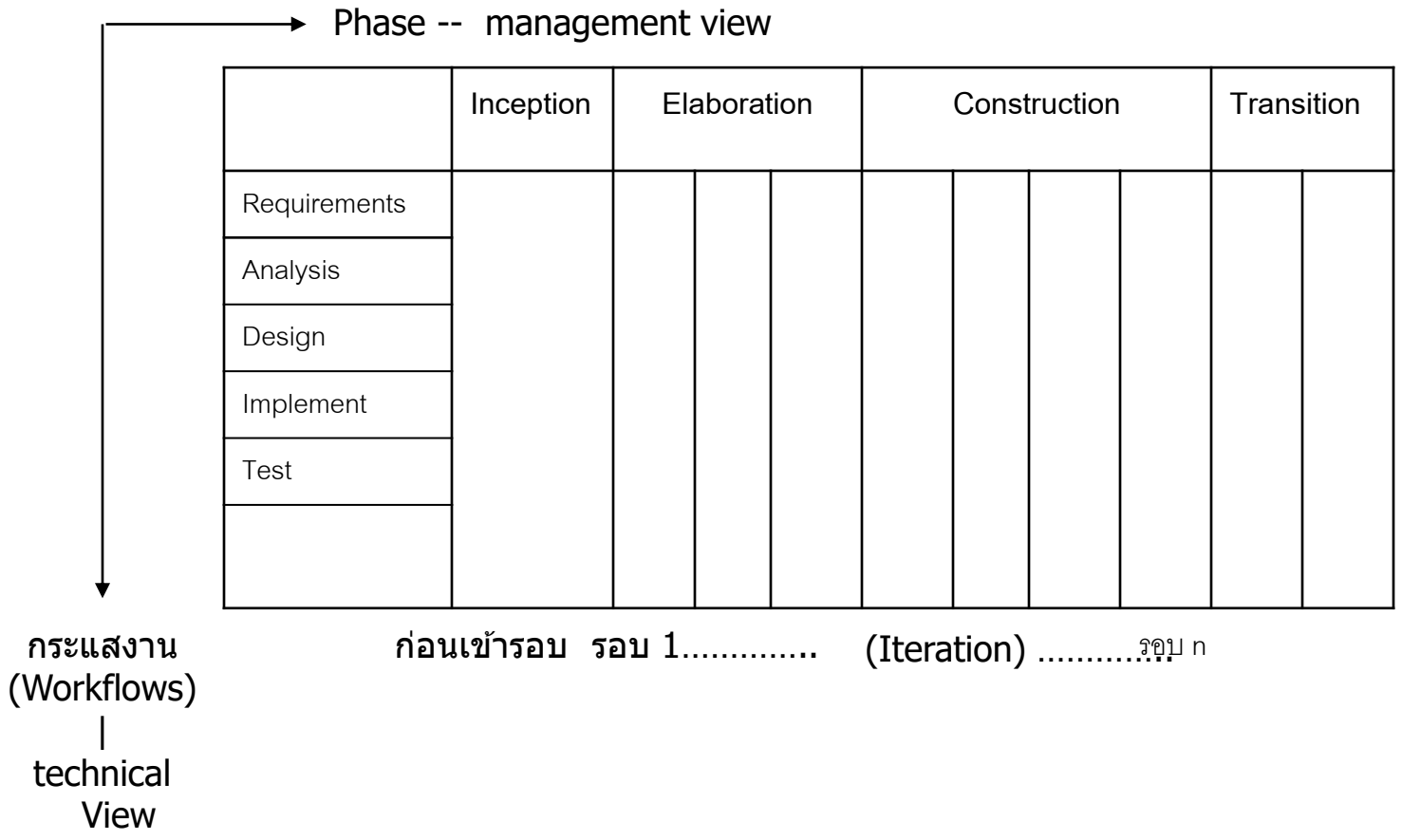
Stand Alone
Client-Server
2 Tier
N Tier

3.3 รูปลักษณ์ของ Unified Process

ไม่ถึงแสน ระดับเล็ก
หลายแสน – ล้านต้น ระดับกลาง
MIS ของ PNRU 5 -10 ล้าน
หลายล้านขึ้นไป ระดับใหญ่

3.4 รอบงานและระยะงาน (Iterations and Phases)

โครงสร้างของ Unified Process ถูกจัดออกเป็น 2 มิติ คือมิติทางด้านเวลา (Time)หรือระยะของการทำงาน (Phase) และมิติกระแสงาน (Workflows)



3.4 รอบงานและระยะงาน (Iterations and Phases)

รอบงานและเฟส ในแต่ละเฟสแบ่งการทำงานออกเป็นรอบงาน (Iteration) จำนวนรอบงานที่มีในแต่ละเฟสจัดแบ่งตามความเหมาะสมของโครงการ

รอบงาน (Iteration) ในแต่ละรอบงานจะประกอบไปด้วยหลายชุดของของกิจกรรม (กระแสงาน) ที่ผู้พัฒนาระบบจะต้องกระทำทั้งกระแสงานหลักและกระแสงานสนับสนุน การทำกิจกรรมเสร็จสิ้นในแต่ละรอบงานจะได้ผลลัพธ์เป็นงานที่สามารถนำไปประมวลผลซึ่งอาจจะเป็นประมวลผลภายในองค์กรของผู้พัฒนาระบบ หรือจัดส่งไปให้ลูกค้าเพื่อนำไปติดตั้งใช้งาน

กระแสงานหลัก (Main workflow)

- **Requirements เก็บรวบรวมความต้องการ**
- **Analysis การวิเคราะห์ เขียนยูสเคส คำอธิบายยูสเคส**
- **Design ออกแบบ แผนภาพคลาส แผนภาพกิจกรรม**
- **Implement การเขียนโปรแกรม Coding (OOP)**
- **Test ทดสอบ ขั้น alpha และขั้น beta**

กระแสงานสนับสนุน (Supporting workflow)

- **Deployment ...ทำเอกสารประกอบ การจัดอบรม การติดตั้ง วิธีการใช้งาน**
- **Project Management การบริหารโครงการ การแบ่งความรับผิดชอบ ประเมินความเสี่ยง การบริหารงบประมาณ หรือการควบคุมค่าใช้จ่าย (Gantt Chart, PERT Chart)**
- **Configuration Management การกำหนดค่า setting**

3.4 รอบงานและระยะงาน (Iterations and Phases)

Configuration Management (CM)

Configuration Management (CM) เป็นกระบวนการที่ใช้ในการจัดการและควบคุมการเปลี่ยนแปลงในระบบหรือผลิตภัณฑ์ โดยเฉพาะในบริบทของซอฟต์แวร์และระบบเทคโนโลยีสารสนเทศ กระบวนการ CM ช่วยให้ทีมพัฒนาและบริหารโครงการสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ และควบคุมการเปลี่ยนแปลงเพื่อให้ผลิตภัณฑ์หรือระบบมีคุณภาพและความเสถียรสูงสุด นี่คือบางความสำคัญเกี่ยวกับ Configuration Management:

1. การจัดเก็บและการเยี่ยมชม (Configuration Identification and Baseline):** CM ช่วยในการระบุและเก็บข้อมูลเกี่ยวกับส่วนประกอบและคุณสมบัติของผลิตภัณฑ์หรือระบบ นี่รวมถึงการกำหนดรุ่นของซอฟต์แวร์และการบันทึกข้อมูลการเปลี่ยนแปลง
2. การควบคุมการเปลี่ยนแปลง (Change Control):** CM ช่วยในการกำหนดกระบวนการและอนุญาตในการเปลี่ยนแปลงในผลิตภัณฑ์หรือระบบ เพื่อให้มั่นใจว่าการเปลี่ยนแปลงมีการประเมินความเสี่ยงและอนุมัติโดยทีมที่เกี่ยวข้อง
3. การการดูแลรักษา (Configuration Management and Maintenance):** CM ช่วยในการดูแลและรักษารายการของผลิตภัณฑ์หรือระบบ แต่ละรายการควรมีการติดตามและรายงานการเปลี่ยนแปลง
4. การการติดตามและรายงาน (Configuration Status Accounting and Reporting):** CM ช่วยในการติดตามสถานะของรายการและการเปลี่ยนแปลง รายงานเหล่านี้มีความสำคัญในการตัดสินใจและการเชื่อมโยงกับกิจกรรมพัฒนาอื่นๆ
5. การควบคุมการสร้าง (Build and Release Control):** CM ช่วยในการควบคุมกระบวนการสร้างและการปล่อยผลิตภัณฑ์หรือระบบ เพื่อให้มั่นใจว่าการสร้างเป็นไปตามแผนและมีคุณภาพ
6. การจัดการกับการสั่งซื้อ (Supplier CM): ถ้าผลิตภัณฑ์หรือระบบรวมองในมุมของการจัดหาจากผู้ผลิตภัณฑ์อื่น ๆ การจัดการกับการสั่งซื้อ (Supplier CM) ช่วยในการติดตามการเปลี่ยนแปลงที่ผู้ผลิตภัณฑ์ทำ

การใช้งาน Configuration Management ช่วยให้ทีมพัฒนามีการควบคุมและควบคุมการเปลี่ยนแปลงในระบบหรือผลิตภัณฑ์ของพวกเขา นี้ช่วยลดความเสี่ยงในการผิดพลาด และให้ความมั่นใจในคุณภาพและความเสถียรของผลิตภัณฑ์หรือระบบที่สร้างขึ้น

I. Inception Phase

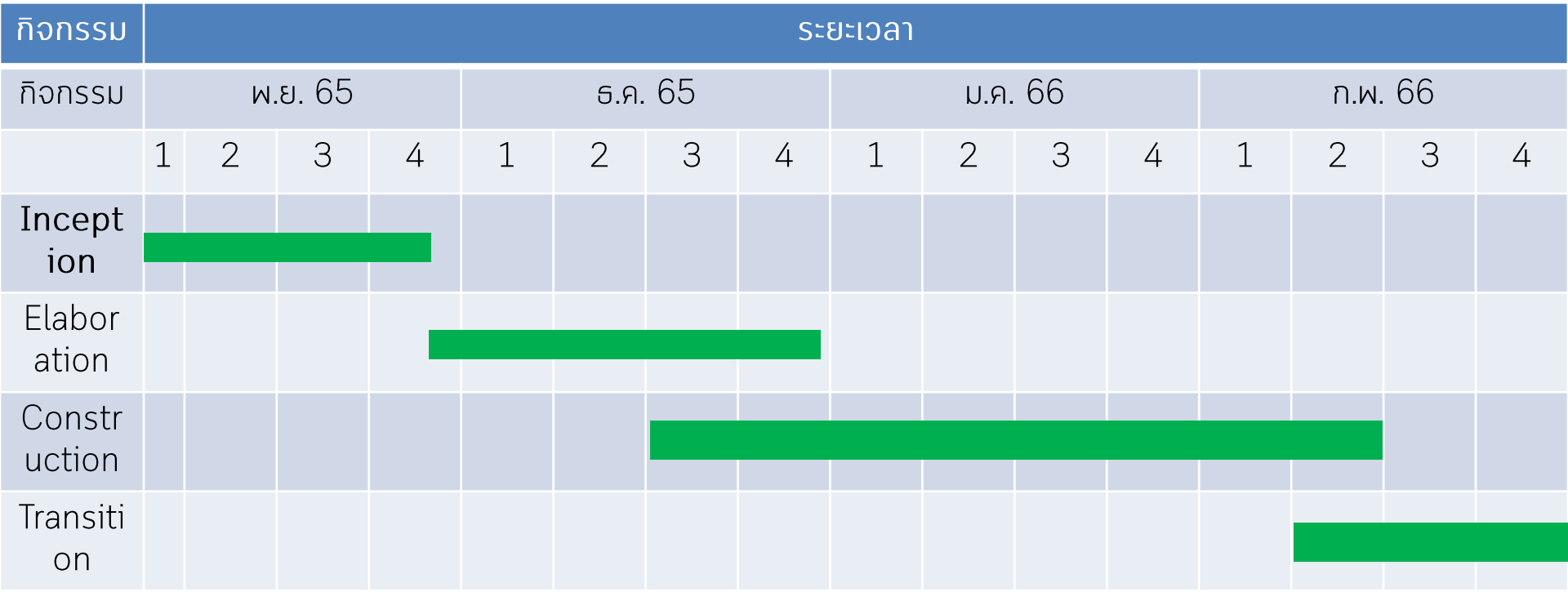
Outcome :

- **เอกสารวิสัยทัศน์ (vision document)**
- **Use case model เบื้องต้น (10-20%)**
- **พจนานุกรมคำศัพท์โครงการ (Project glossary) เบื้องต้น**
- **กรณีการทำธุรกิจ (Business Model/Rule) ครอบคลุมบริบทธุรกิจ เงื่อนไข**
- **การจัดการความเสี่ยงเบื้องต้น (Risk Plan)**
- **แผนโครงการแสดงระยะงานและรอบงาน (Project Management) Gantt Chart , PERT Chart**
- **โมเดลธุรกิจ (ถ้าจำเป็น)**
- **ต้นแบบระบบ (prototypes) (UX/UI)**

การตระเตรียมงาน Inception

ถ้ากำหนดโครงไว้ 1 ปี ระยะ 1. - 2 . เดือน แรกเสร็จ ตามข้างบน

Gantt Chart



I. การตระเตรียมงาน Inception

II. Elaboration Phase

II. Elaboration Phase

ถ้ากำหนดโครงไว้ 1 ปี ระยะ 3. – 4,5 . เดือน แรกเสร็จ ตามข้างบน

Outcome :

- **Use case โมเดลสมบูรณ์อย่างน้อย 80%**
- **Supplementary requirement (non-functional requirements)**
- **คำอธิบายสถาปัตยกรรมซอฟต์แวร์ (Software Architecture Description)**
  - **Hardware & software**
  - **Deployment Diagram , Component Diagram**
- **ต้นแบบที่สามารถนำไป execute ได้ (Prototype สำหรับ Demo , mockup , wireframe)**
- **ปรับรายการความเสี่ยง (Change/Adjust Risk Plan)**
- **แผนการพัฒนาระบบทั้งโครงการ (Complete)**
- **user manual (Opt.)**

III. Construction Phase

ถ้ากำหนดโครงไว้ 1 ปี ระยะ 3. – 5,6,7,8,9 . เสร็จ ตามข้างล่าง

Construction, Implementation / Testing

ทดสอบระบบ Alpha ทดลองโดยทีมผู้พัฒนา, Beta เวอร์ชันทดลองใช้ , Release

Outcome :

- **ระบบซอฟต์แวร์ที่ตรงกับแพลตฟอร์มเป้าหมาย**
- **คู่มือใช้งาน (user manuals)**
- **คำอธิบายระบบที่ออกวางจำหน่าย**
  - **license**

IV. Transition Phase

ถ้ากำหนดโครงไว้ 1 ปี ระยะ 3. – 11,12 . เสร็จ ตามข้างล่าง

เป็นเฟส การถ่ายโอนระบบไปยังผู้ใช้

Outcome :

- **ทดสอบเพื่อรับระบบ**
- **ทดสอบการ Integrate ของระบบ ความเข้ากันได้ของระบบย่อยต่าง ๆ**
- **แปลงข้อมูลลงสู่ฐานข้อมูลที่ใช้งาน / การอัตเดตฐานข้อมูลเก่า**
- **อบรมผู้ใช้และผู้บำรุงรักษาระบบ**
- **ออกวางระบบสู่ท้องตลาด (สำหรับจำหน่าย)**

3.5 โครงสร้างการทำงานของ UP

(Static structure of the process)

UP กำหนดการปฏิบัติงานของทีมงานไว้อย่างชัดเจนว่าใคร (who) ทำอะไร (what) ทำเมื่อไร (when) และทำอย่างไร (how) เพื่อบรรลุเป้าหมายที่ตั้งไว้

UP แสดงโครงสร้างการปฏิบัติงานเป็นโมเดลที่มี 4 สัญญลักษณ์พื้นฐาน

สัญญลักษณ์	ความหมาย
 ผู้ปฏิบัติ (worker)	**ผู้ปฏิบัติงาน (worker)** - who หมายถึงบทบาทในโครงการที่กระทำโดยบุคคลใดบุคคลหนึ่งหรือทีม
	ชิ้นงาน (artifact) - what หมายถึงงานที่ผลิตโดยผู้ปฏิบัติในรูปของข้อมูลสารสนเทศ โปรแกรม รหัสโปรแกรม หรือเอกสาร (document) **กิจกรรม (activity)** - how หมายถึงกิจกรรมที่ผู้ปฏิบัติกระทำ
 กระแสงาน	**กระแสงาน (workflow)** - when หมายถึงชุดของกิจกรรมที่เกี่ยวเนื่อง

3.5 โครงสร้างการทำงานของ UP

3.5.1 ผู้ปฏิบัติงาน (Worker)

ผู้ปฏิบัติงานหรือ worker นิยามถึงพฤติกรรมและความรับผิดชอบ (บทบาทความรับผิดชอบ) ของตัวบุคคลหรือกลุ่มบุคคลที่ทำงานเป็นทีม

1. CEO
2. PM
3. SA
4. ...
5. ...
6. ...
7. Programmer / DEV .....Source Code / Executable
    1. Front-end
    2. Back-end
8. UXUI Designer
9. Tester

3.5 โครงสร้างการทำงานของ UP

3.5.1 ผู้ปฏิบัติงาน (Worker)

ผู้ปฏิบัติงานหรือ worker นิยามถึงพฤติกรรมและความรับผิดชอบ (บทบาทความรับผิดชอบ) ของตัวบุคคลหรือกลุ่มบุคคลที่ทำงานเป็นทีม

Resource	Worker	Activities
Somchai	Designer	Object Design
Somying	Use-Case Author	Detail a Use-Case
Sombat	Use-Case Designer	Use-Case Design
Somboon	Design Review	Review Design
Somruk	Architect	Architectural Analysis Architectural Design

3.5 โครงสร้างการทำงานของ UP

3.5.1 ผู้ปฏิบัติงาน (Worker)

ผู้ปฏิบัติงานหรือ worker นิยามถึงพฤติกรรมและความรับผิดชอบ (บทบาทความรับผิดชอบ) ของตัวบุคคลหรือกลุ่มบุคคลที่ทำงานเป็นทีม

Resource	Worker	Activities
อติการ	Designer	Object/class Design
ชโลธร	Use-Case Author	Detail a Use-Case
ณัฐพล	Use-Case Designer	Use-Case Design
อมรชัย	Design Reviewer	Review Design
ชาลิสา	Architect	Architectural Analysis Architectural Design

3.5 โครงสร้างการทำงานของ UP

3.5.2 กิจกรรม (Activity)

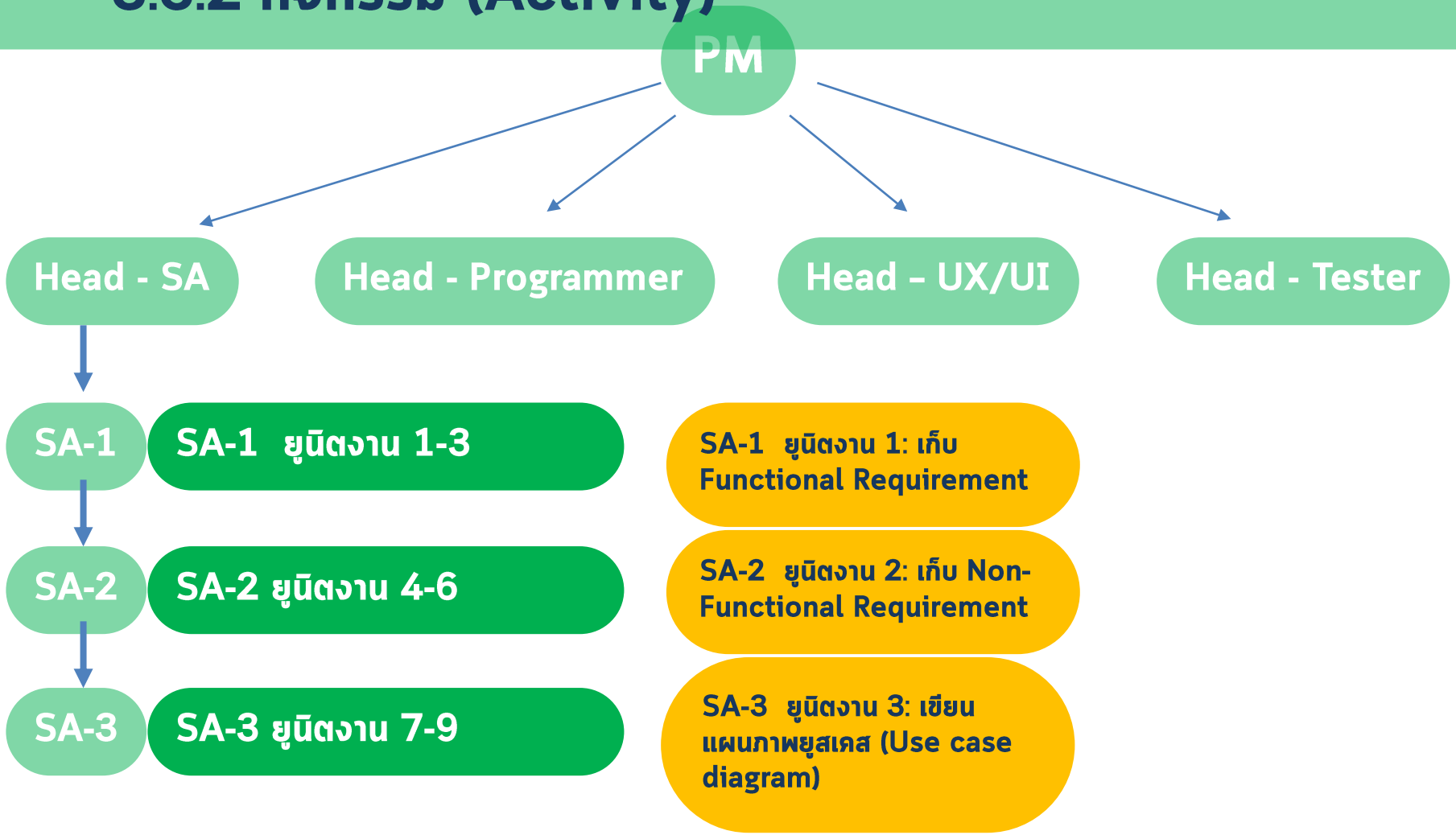
กิจกรรม (activity) ของผู้ปฏิบัติงานใด ๆ คือยูนิตงาน ที่ผู้ปฏิบัติงานนั้นได้รับการมอบหมายให้ทำ(จาก Project Manager) แต่ละกิจกรรมมีจุดหมายที่ชัดเจน โดยทั่วไปจะเป็นกิจกรรมในการสร้างหรือปรับปรุงชิ้นงาน (artifact) เช่น การสร้างและปรับปรุงโมเดลหรือ class การจัดทำและปรับแผน

ตัวอย่าง Activities :

- Plan an iteration, for the Worker: Project Manager
- Find use cases and actors, for the Worker: System Analyst
- Review the design, for the Worker: Design Reviewer
- Execute performance test, for the Worker: Performance Tester

3.5 โครงสร้างการทำงานของ UP

3.5.2 กิจกรรม (Activity)



3.5 โครงสร้างการทำงานของ UP

3.5.2 กิจกรรม (Activity)

ชโลธร เป็น Project Manage จะทำหน้าที่ มอบหมาย ควบคุม ติดตาม ประเมินผล worker
กำหนดรอบให้ ณัฐพล เก็บ Requirement ใช้เวลา 2 สัปดาห์

ชโลธร เป็น Project Manage จะทำหน้าที่ มอบหมาย ควบคุม ติดตาม ประเมินผล worker
กำหนด รอบให้ ณัฐพล เก็บ Requirement ใช้เวลา 1-2 สัปดาห์

สมชาย เป็น Object Designer ทำหน้าที่หา Object จากโดเมน เขียนเป็น Class

3.5 โครงสร้างการทำงานของ UP

3.5.3 ชิ้นงาน (Artifact)

ชิ้นงาน (Artifact) เป็นส่วนของสารสนเทศที่ถูกจัดสร้าง บำรุงรักษา หรือถูกใช้ในการพัฒนาซอฟต์แวร์ ชิ้นงานเป็นผลิตภัณฑ์เกิดขึ้นระหว่างการพัฒนาระบบซึ่งนำไปสู่ผลิตภัณฑ์เป้าหมายของโครงการคือระบบซอฟต์แวร์ในที่สุด

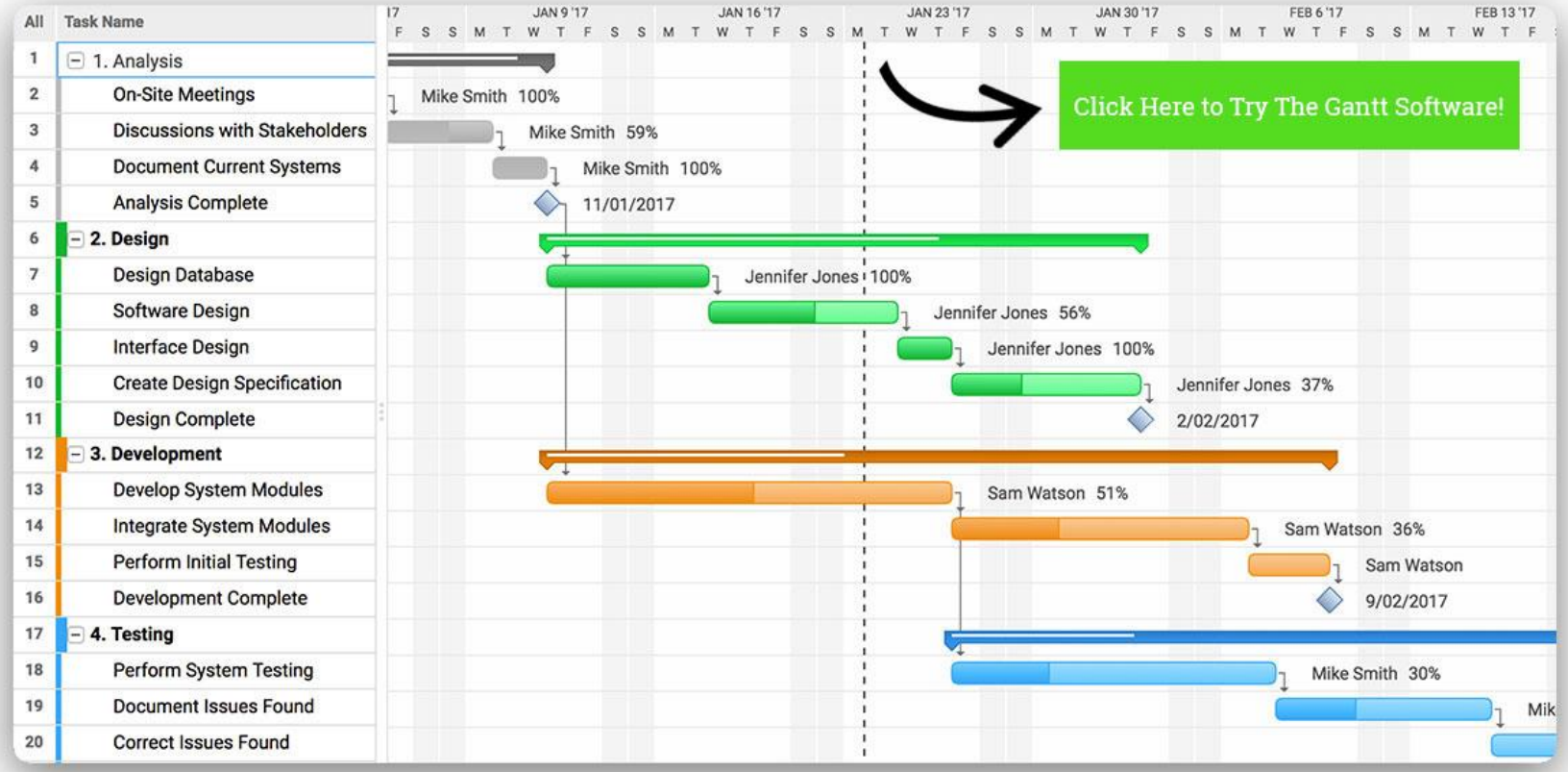
ชิ้นงานมีได้หลายรูปแบบ

- **A model, such as the Use-case Model or the Design Model**
- **A model element, i.e. an element within a model, such as a class, a use case or a subsystem**
- **A document, such as business Case or Software Architecture Document**
- **Source Code เช่น การเข้าสู่ระบบ การสมัครสมาชิก การสั่งซื้อ การโอนเงิน การฝากเงิน การลงทะเบียนเรียน เป็น**
- **Executables เช่น โปรแกรมที่คอมไพล์แล้ว *.exe**
- **Mockup ,wireframe การออกแบบจอต่าง ๆ เช่น หน้าจอ login หน้าจอสั่งซื้อสินค้า หน้าจอถอนรายวิชา**

3.5 โครงสร้างการทำงานของ UP

3.5.3 ชิ้นงาน (Artifact)

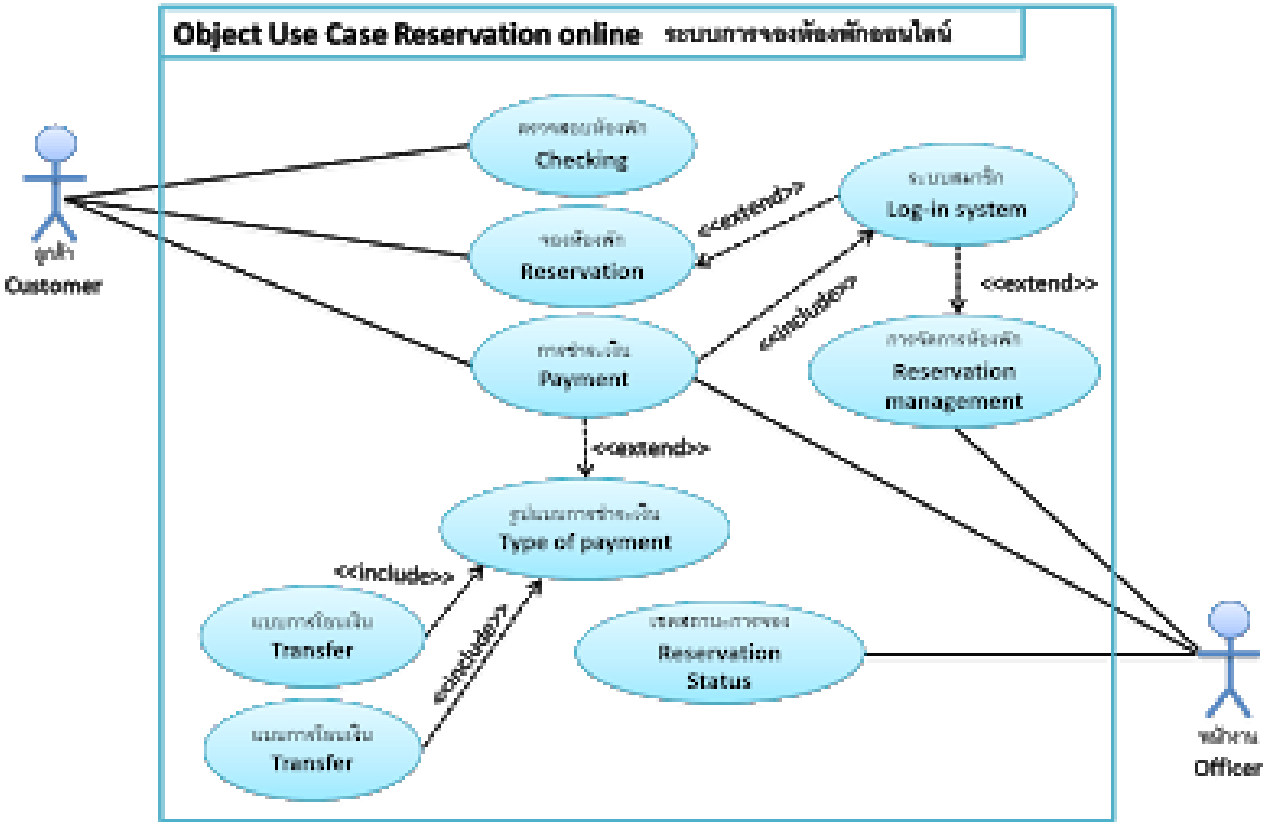
ชิ้นงาน Gantt Chart แผนการดำเนินโครงการ หน้าที่ PM



3.5 โครงสร้างการทำงานของ UP

3.5.3 ชิ้นงาน (Artifact)

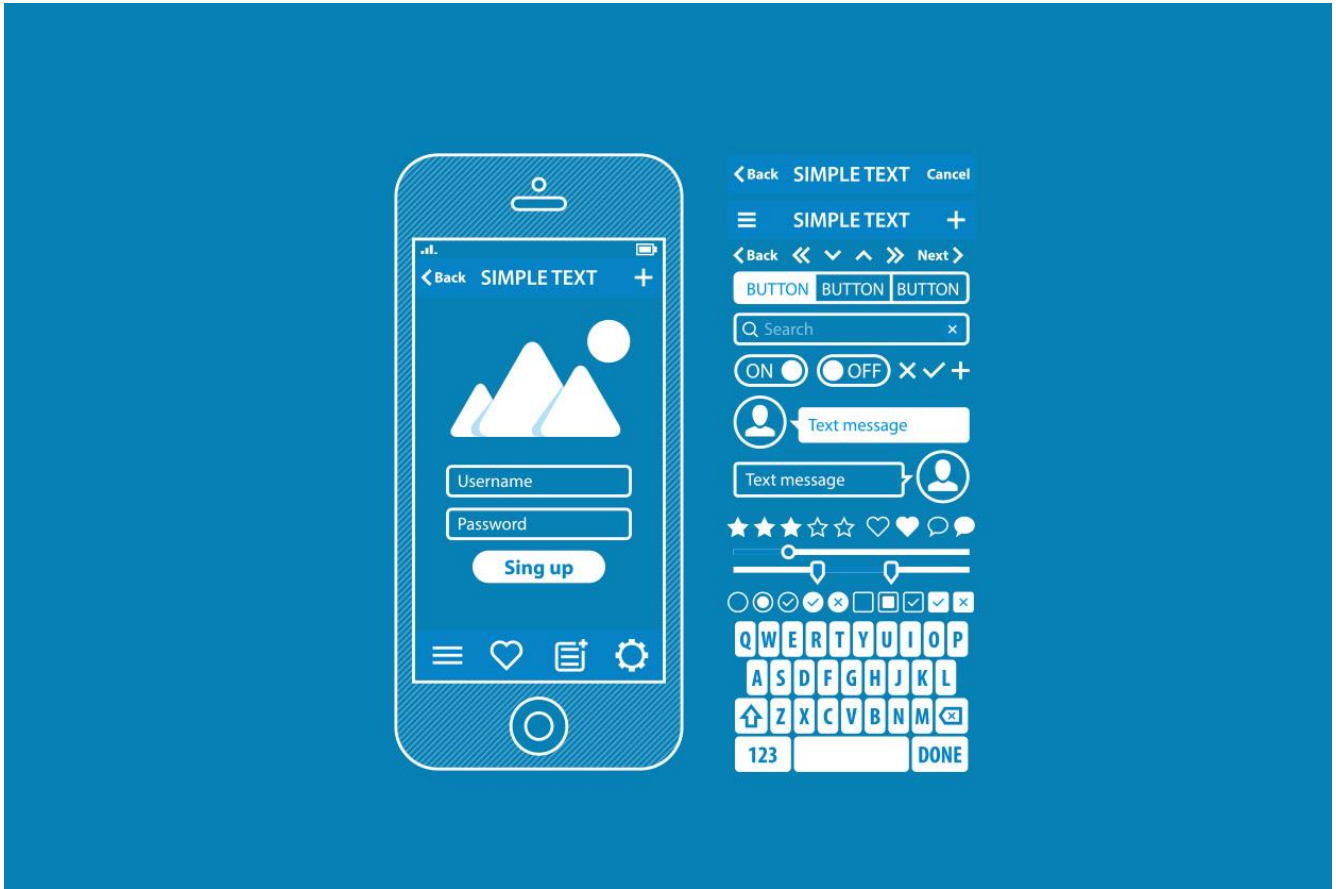
ชิ้นงาน Use case diagram หน้าที่ของ SA use case designer



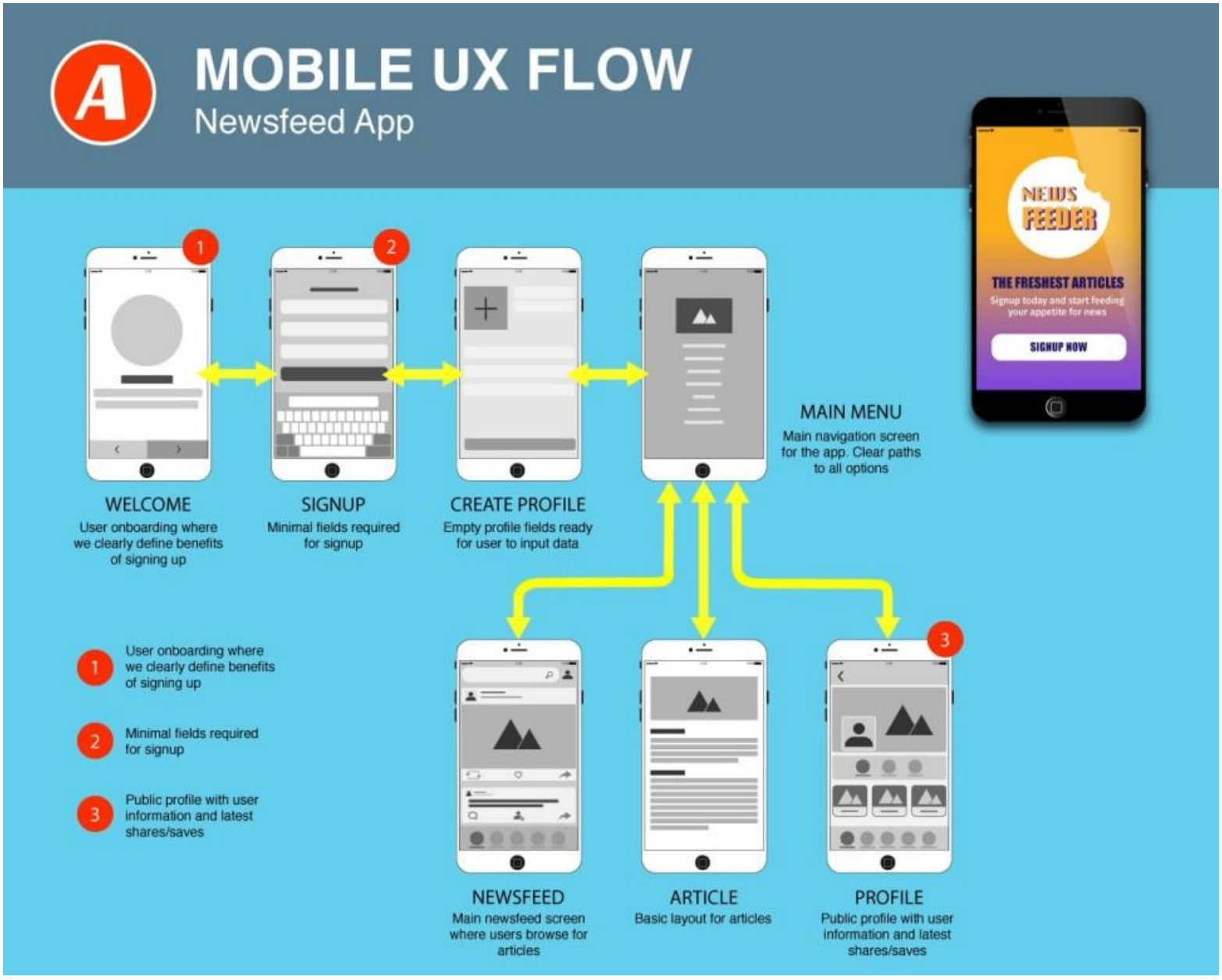
3.5 โครงสร้างการทำงานของ UP

3.5.3 ชิ้นงาน (Artifact)

UXUI Designer



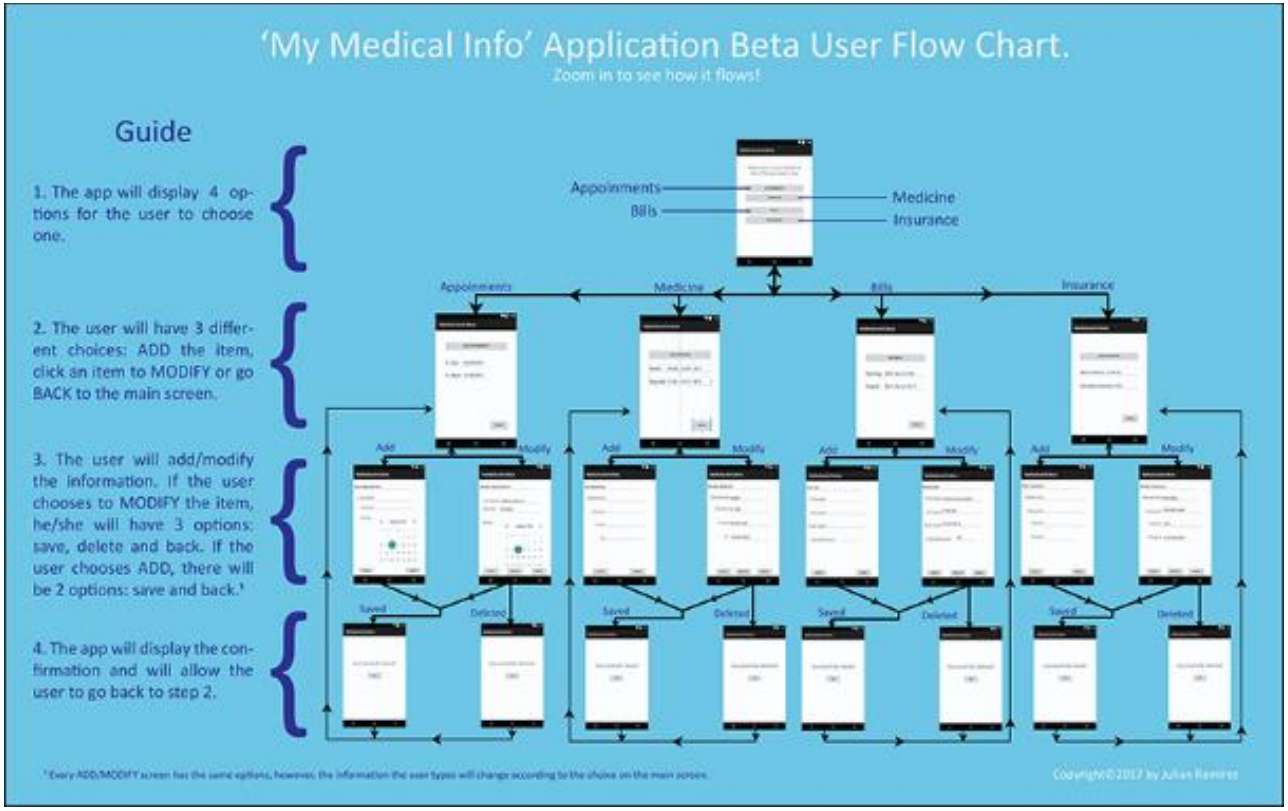
UXUI Designer



3.5 โครงสร้างการทำงานของ UP

3.5.3 ชิ้นงาน (Artifact)

I worked on this project during a semester. It was interesting how my UX / UI skills improved during this time. The project was a mix of sketches, personas, storyboards, user flow charts and user testing.



3.5 โครงสร้างการทำงานของ UP

การแบ่งงานเป็นยูนิตงาน (Unit of Work)

การแบ่งงานเป็นยูนิตงาน (Unit of Work) เป็นหนึ่งในแนวคิดที่สำคัญของ Unified Process (UP) ซึ่งเป็นกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ แนวคิดนี้กำหนดให้งานพัฒนาซอฟต์แวร์ทั้งหมดถูกแบ่งออกเป็นยูนิตงานย่อย ๆ ที่เล็กและจัดการได้ง่าย แต่ละยูนิตงานจะรับผิดชอบงานเฉพาะอย่าง การทำงานร่วมกันของยูนิตงานต่าง ๆ จะทำให้งานพัฒนาซอฟต์แวร์ทั้งหมดสำเร็จลุล่วง

ยูนิตงานใน UP มีลักษณะดังนี้

- เล็กและจัดการได้ง่าย
- รับผิดชอบงานเฉพาะอย่าง
- ทำงานร่วมกันได้อย่างอิสระ
- มีการทดสอบและตรวจสอบความถูกต้องแยกกัน

3.5 โครงสร้างการทำงานของ UP

การแบ่งงานเป็นยูนิตงาน (Unit of Work)

การแบ่งงานเป็นยูนิตงานมีประโยชน์หลายประการ ดังนี้

- ทำให้งานพัฒนาซอฟต์แวร์มีความยืดหยุ่นและปรับแต่งได้ง่าย
- ทำให้การทำงานร่วมกันของทีมพัฒนาซอฟต์แวร์มีประสิทธิภาพมากขึ้น
- ทำให้กระบวนการพัฒนาซอฟต์แวร์มีความน่าเชื่อถือมากขึ้น

การแบ่งงานเป็นยูนิตงานสามารถทำได้หลายวิธี ขึ้นอยู่กับขนาดและลักษณะของโครงการ โดยทั่วไปแล้ว ยูนิตงานจะแบ่งตามหน้าที่การทำงาน (functionality) ของซอฟต์แวร์ ตัวอย่างเช่น ยูนิตงานหนึ่งอาจรับผิดชอบงานด้านการแสดงผล (UI) อีกยูนิตงานหนึ่งอาจรับผิดชอบงานด้านฐานข้อมูล เป็นต้น ยูนิตงานยังสามารถแบ่งตามคุณสมบัติของซอฟต์แวร์ (feature) ของซอฟต์แวร์ ตัวอย่างเช่น ยูนิตงานหนึ่งอาจรับผิดชอบคุณสมบัติการลงทะเบียนผู้ใช้ อีกยูนิตงานหนึ่งอาจรับผิดชอบคุณสมบัติการสั่งซื้อสินค้า เป็นต้น

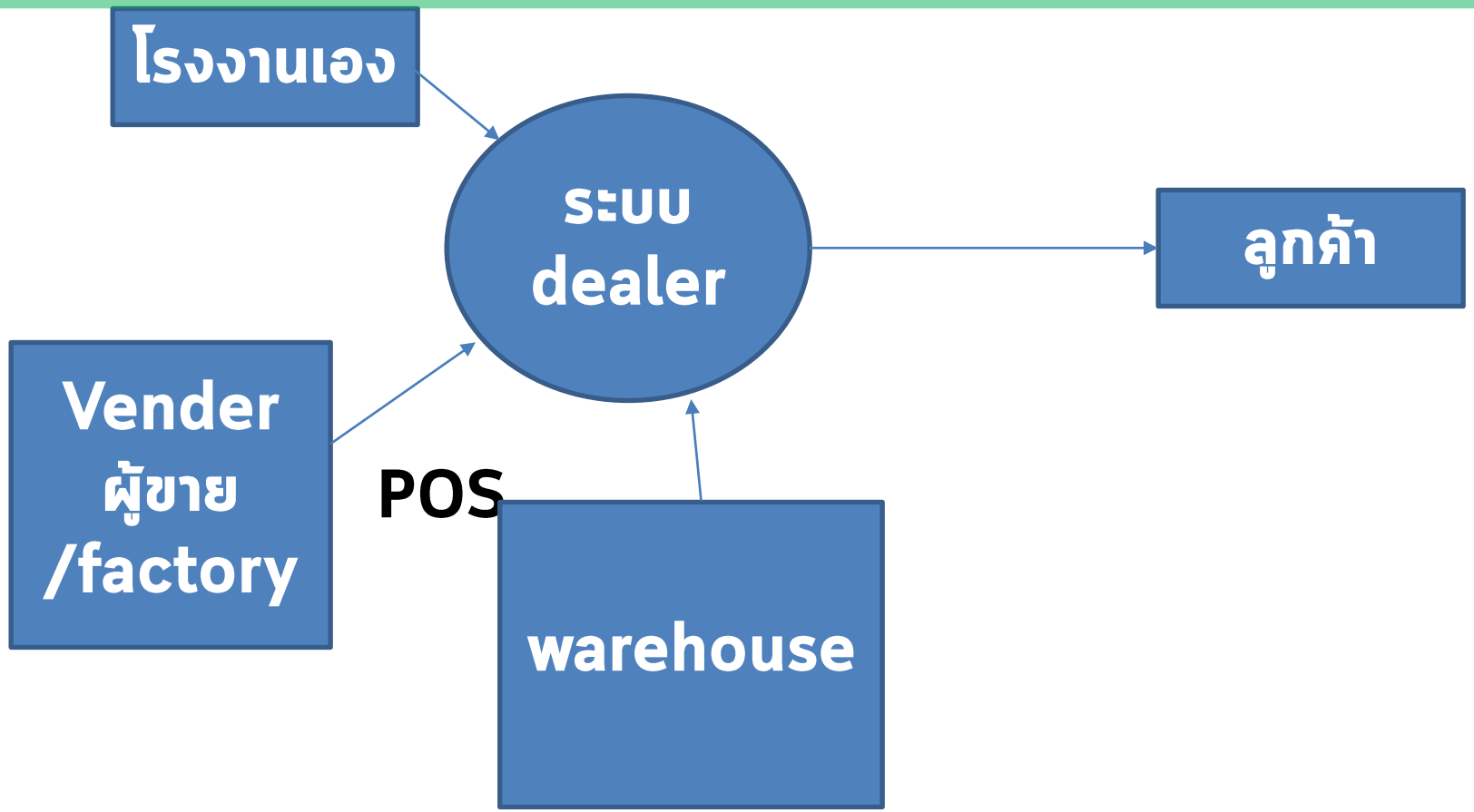
การแบ่งงานเป็นยูนิตงานเป็นแนวคิดที่สำคัญที่ช่วยให้กระบวนการพัฒนาซอฟต์แวร์มีประสิทธิภาพมากขึ้น ช่วยให้ทีมพัฒนาซอฟต์แวร์ทำงานร่วมกันได้อย่างราบรื่น และช่วยให้ซอฟต์แวร์ที่พัฒนาขึ้นมีคุณภาพดี

3.5 โครงสร้างการทำงานของ UP

การแบ่งงานเป็นยูนิตงาน (Unit of Work)

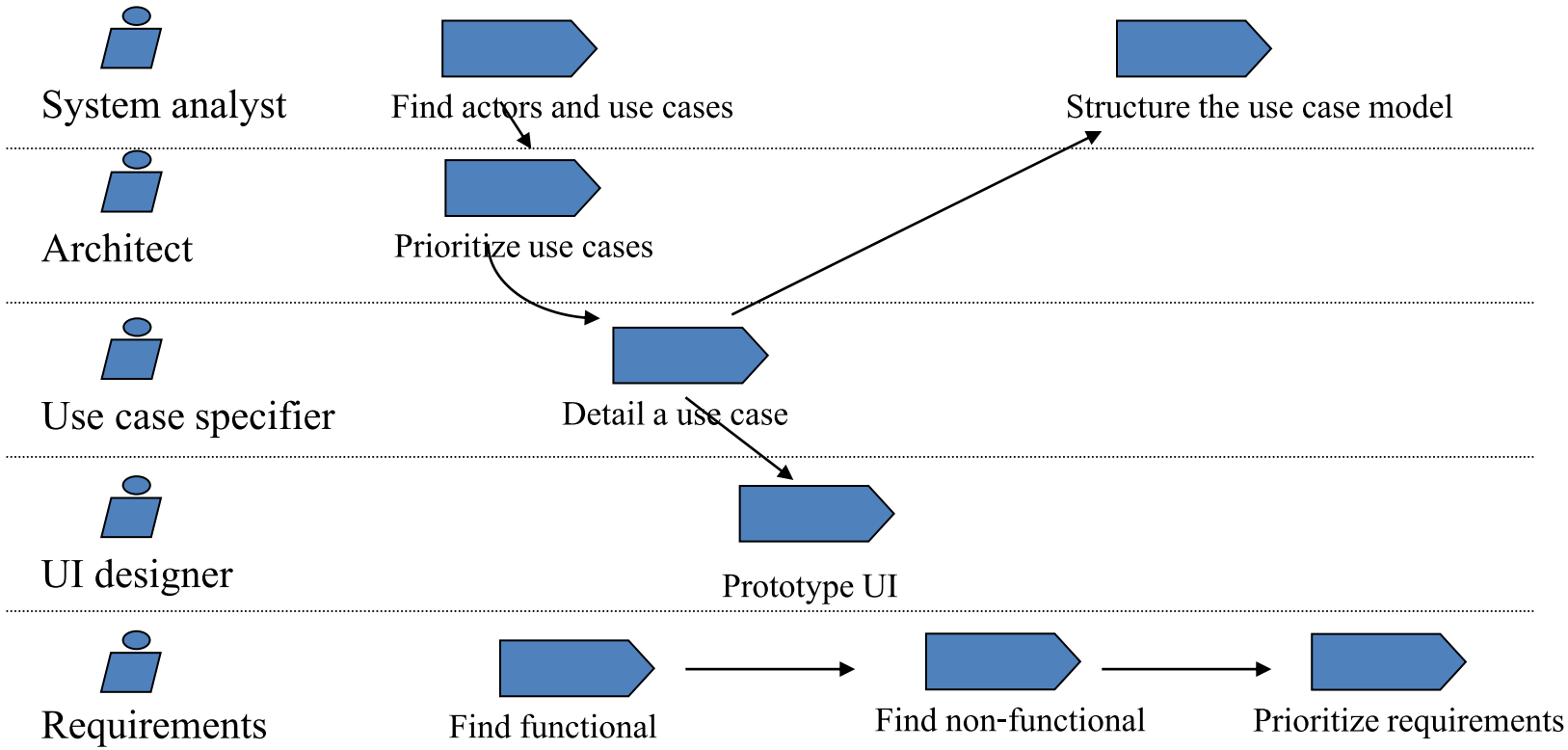
- แต่ละยูนิตงาน ต้องเท่ากัน
  - หน้าจอ login
  - การหา actor และ usecae
- คนหนึ่ง ๆ จะทำหน้าที่ (worker) ได้หลายหน้าที่
  - สมชาย เป็น Object designer, Class designer , class Reviewer

การแบ่งงานเป็นยูนิตงาน (Unit of Work)



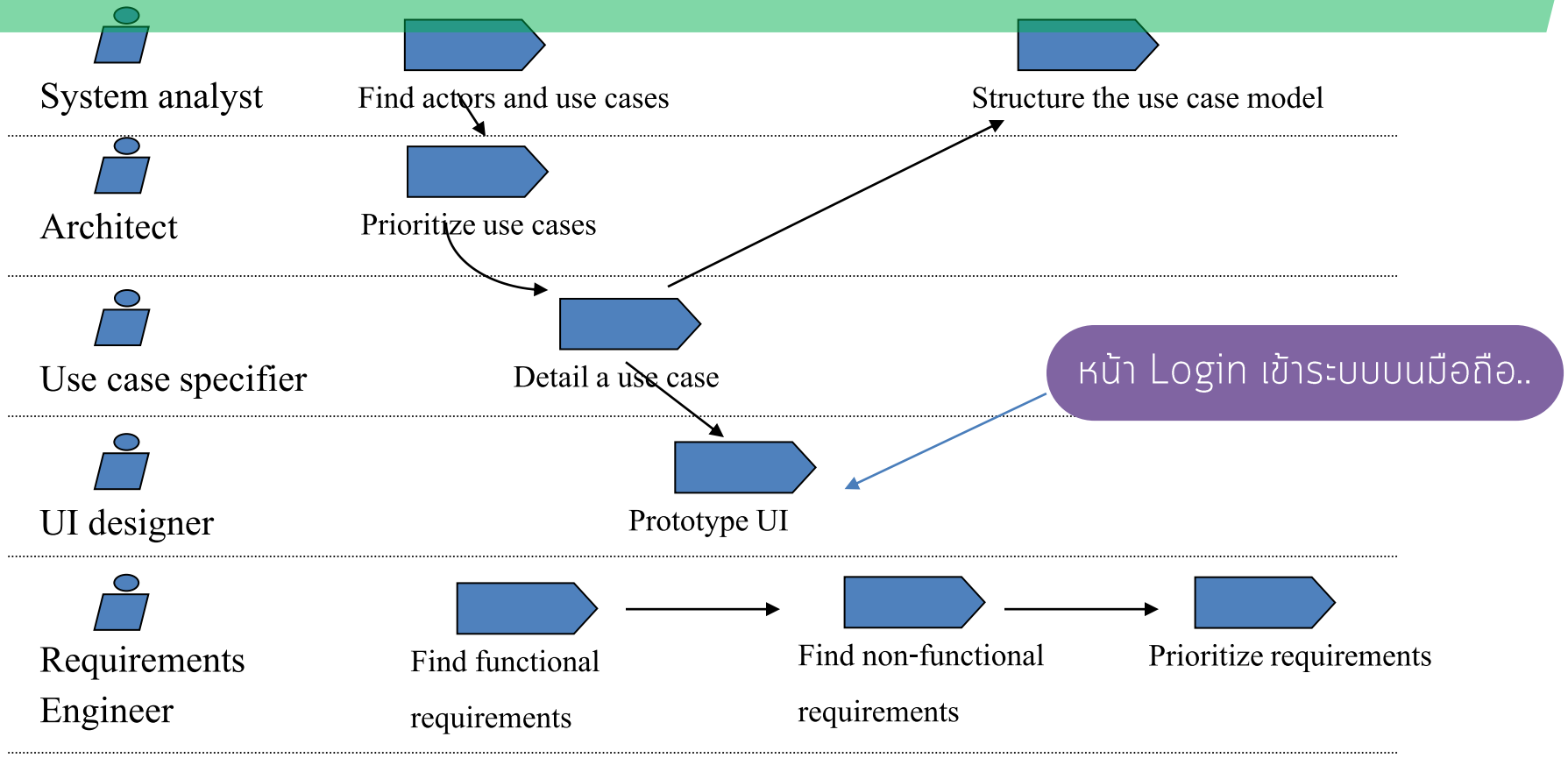
4.6 กระแสงาน (Workflows)

กระแสงาน (workflow) เป็นชุดของกิจกรรมที่สร้างผลลัพธ์ที่มีคุณค่าต่อโครงการ



3.6 กระแสงาน (Workflows)

กระแสงาน (workflow) เป็นชุดของกิจกรรมที่สร้างผลลัพธ์ที่มีคุณค่าต่อโครงการ



กระแสงานหลัก (Core workflows)

The core process workflows are divided into 5 core "engineering" workflow :

1. **Requirements workflow**
2. **Analysis workflow**
3. **Design workflow**
4. **Implementation workflow**
5. **Test workflow**

And three core "supporting" workflow :

1. **Project Management workflow**
2. **Configuration and Change Management workflow**
3. **Environment workflow**

Requirements Workflow

The goal of the Requirements workflow is to describe what the system should do and allows the developers and the customer to agree on that description. To achieve this, we elicit, organize, and document required functionality and constraints; track and document tradeoffs and decisions.

Goals

- Reach agreement on the system context
- List candidate requirements
- Identify & negotiate requirements
- Specify nonfunctional requirements

Artifacts

- Domain model
- Business model
- Glossary
- Actors & Use Case
- User-Interface Prototype
- Use Case model

Workers

- System analyst
- Use Case specifier
- User-Interface designer
- Architect

Activities

- Build the domain model
- Build the business model
- Find actors and use cases
- Prototype the user interface
- Prioritize the use cases
- Detail a use case
- Structure the model

Analysis Workflow

The goal of the Analysis workflow is to gain real understanding of the customer's requirements and to use that understanding to build momentum as the project heads into design and implementation.

Goals

- Gain real understanding of the customers' requirements

Artifacts

- Analysis class (Boundary class, Entity class, Control class)
- Use Case Realization - Analysis
- Analysis package
- Analysis model

Workers

- Architect
- Use Case engineer
- Component engineer

Activities

- Perform architectural analysis
- Analyze a use case
- Analyze a class
- Analyze a package

Design Workflow

The goal of the Design workflow is to build a blueprint of the system that the team can rely on going forward into implementation.

Goals

- Build a blueprint of the system

Artifacts

- Design class
- Use Case Realization - Design
- Interface
- Design Subsystem
- Design Model
- Architectural Description
- Deployment Model

Workers

- Architect
- Use Case engineer
- Component engineer

Activities

- Perform architectural design
- Design a use case
- Design a class
- Design a subsystem

Implementation Workflow

The goal of the Design workflow is to build a working version of the system that it can deliver to beta customers for evaluation.

Goals

- Build a working version of the system

Artifacts

- Component
- Interface
- Implementation subsystem
- Implementation model
- Architecture description
- Integration build plan

Workers

- Architect
- Component engineer
- System integrator

Activities

- Perform architectural implementation
- Implement a class
- Perform unit test
- Implement a subsystem
- Integrate the system

Test Workflow

The goal of the test workflow is to ensure that the system offers a high degree of quality before it's delivered to customers.

Goals

- To verify the interaction between objects.
- To verify the proper integration of all components
- To verify that all requirements have been correctly implemented
- To identify and ensure defects are addressed prior to the deployment of the software

Artifacts

- Test Case
- Test Procedure
- Test Component
- Test Model
- Test Plan
- Defect
- Test Evaluation

Workers

- Test Engineer
- Component Engineer
- Integration Tester
- System Tester

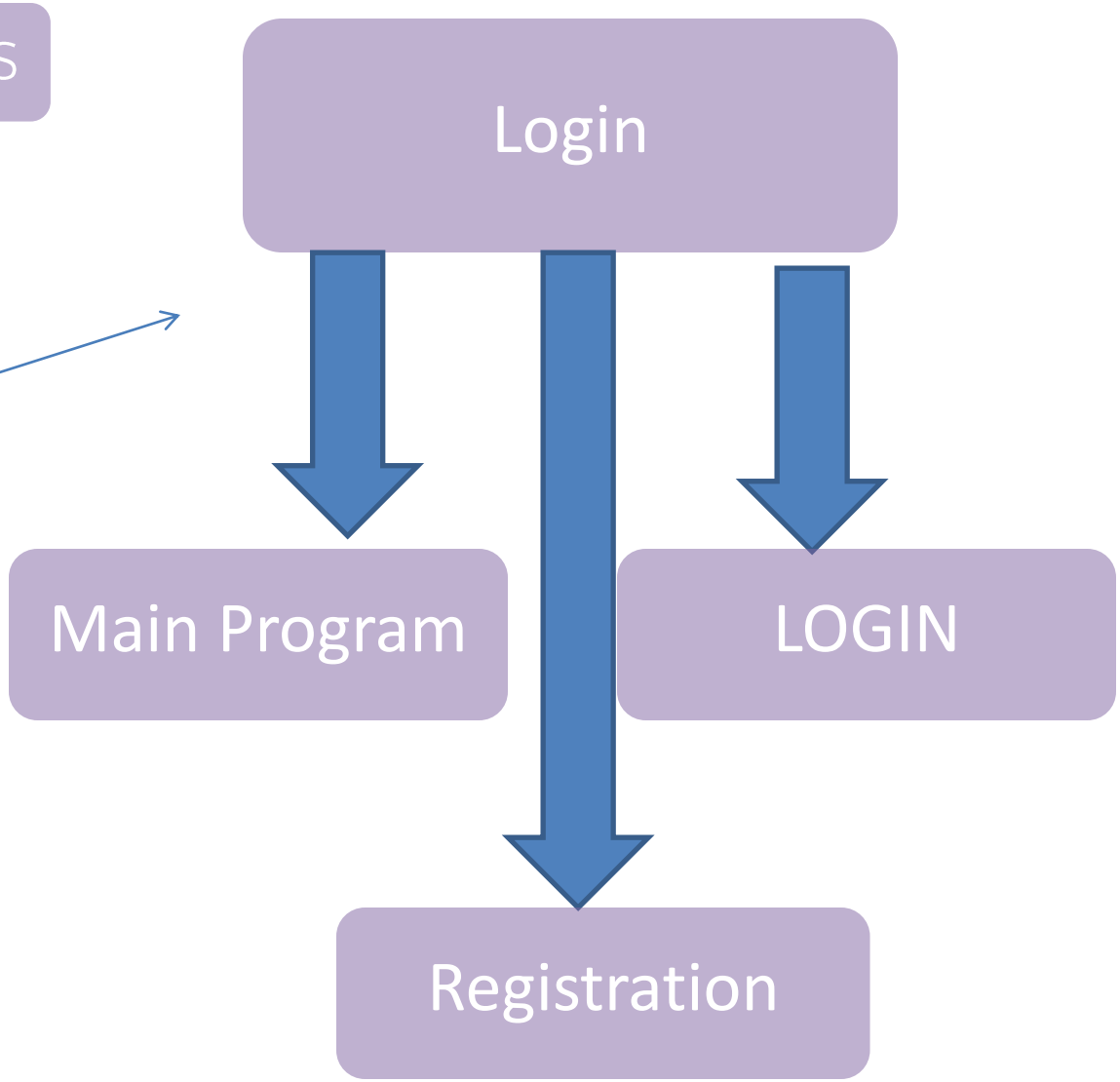
Activities

- Plan test
- Design test
- Implement test
- Perform Integration test
- Perform System test
- Evaluate test

ทดสอบกระบวนการ

Artifacts

- Test Case
- Test Procedure
- Test Component
- Test Model
- Test Plan
- Defect
- Test Evaluation



3.6 กระแสงาน (Workflows)

Deployment

The purpose of the deployment workflow is to successfully produce product releases, and deliver the software to its end users. It covers a wide range of activities including :

- **Producing external releases of the software.**
- **Packaging the software.**
- **Distributing the software.**
- **Installing the software.**
- **Providing help and assistance to users.**
- **In many cases, this also includes activities such as:**
- **Planning and conduct of beta tests.**
- **Migration of existing software or data.**
- **Formal acceptance.**

Project Management

- **A framework for managing software - intensive projects.**
- **Practical guidelines for planning, staffing, executing, and monitoring projects.**
- **A framework for managing risk. จัดทำแผนความเสี่ยง**

Project Management

Project Management (การบริหารโครงการ) เป็นกระบวนการที่ใช้ในการวางแผน ดำเนินการ และควบคุมการทำงานเพื่อสร้างผลิตภัณฑ์หรือผลลัพธ์ที่มีคุณภาพตามเป้าหมายที่กำหนดไว้ โดยใช้ทรัพยากรที่มีอยู่อย่างมีประสิทธิภาพ รวมถึงการใช้เครื่องมือและเทคนิคที่เหมาะสมในการบริหารโครงการ
รายละเอียดของ Project Management ประกอบด้วย:

1. การวางแผนโครงการ (Project Planning):
- **กำหนดเป้าหมายและขอบเขตของโครงการ รวมถึงการกำหนดระยะเวลาและงบประมาณที่เหมาะสม**
- **วางแผนการทำงานและกำหนดรายละเอียดของขั้นตอนการดำเนินงานในโครงการ**

2. การกำหนดทรัพยากร (Resource Allocation):
- **แบ่งส่วนทรัพยากรที่มีอยู่ เช่น งบประมาณ บุคลากร อุปกรณ์ และเทคโนโลยี เพื่อให้เหมาะสมกับความต้องการของโครงการ**
- **กำหนดการใช้ทรัพยากรให้เกิดผลสูงสุดในการดำเนินงานในโครงการ**

Project Management

3. การติดตามและควบคุม (Monitoring and Control):

- ติดตามความก้าวหน้าของโครงการและการดำเนินงานเพื่อตรวจสอบว่าโครงการดงอยู่ในเส้นทางที่ถูกต้องหรือไม่

- ควบคุมกระบวนการในการทำงานของโครงการและการปรับปรุงเพื่อให้ทันสถานการณ์และเป้าหมายที่กำหนดไว้

4. การจัดทำรายงาน (Reporting):

- สร้างรายงานเกี่ยวกับความก้าวหน้า ปัญหา และผลการดำเนินงานของโครงการ ให้ผู้ที่เกี่ยวข้องทราบเพื่อให้มีการตัดสินใจที่เหมาะสม

5. การจัดการความเสี่ยง (Risk Management):

- วิเคราะห์และระบุความเสี่ยงที่อาจเกิดขึ้นในโครงการ และกำหนดวิธีการบริหารจัดการกับความเสี่ยงเหล่านั้น

- ให้ความสำคัญในการควบคุมและลดความเสี่ยงที่อาจเกิดขึ้นในโครงการ

Project Management

6. การควบคุมคุณภาพ (Quality Control):

- ตรวจสอบและประเมินคุณภาพของผลลัพธ์ที่ได้จากโครงการเพื่อให้มั่นใจว่าตรงตามมาตรฐานที่กำหนดไว้

- ตรวจสอบและปรับปรุงกระบวนการทำงานเพื่อให้สามารถทำงานได้อย่างมีคุณภาพ

7. การปรับปรุงและสิ้นสุดโครงการ (Project Closure):

- ประเมินผลและสิ้นสุดโครงการอย่างเหมาะสม

- สรุปผลการดำเนินงานและเรียนรู้จากประสบการณ์ในโครงการ

การบริหารโครงการเป็นกระบวนการที่เกี่ยวข้องกับการวางแผน การดำเนินงาน และควบคุมในการพัฒนาระบบหรือผลิตภัณฑ์ที่ต้องการ มีบทบาทสำคัญในการดำเนินงานให้สามารถสร้างผลลัพธ์ที่ต้องการได้ และช่วยลดความเ

Configuration & Change Management

- **Simultaneous Update - When two or more workers work separately on the same artifact, the last one to make changes destroys the work of the former.**
- **Limited Notification - When a problem is fixed in artifacts shared by several developers, and some of them are not notified of the change.**
- **Multiple Versions**

Configuration & Change Management

Configuration & Change Management (การจัดการและการเปลี่ยนแปลงการกำหนดค่า) เป็นกระบวนการที่ใช้ในการจัดการและควบคุมการเปลี่ยนแปลงของซอฟต์แวร์และระบบที่พัฒนาขึ้น เพื่อให้มั่นใจว่าระบบมีความเสถียรและสอดคล้องกับความต้องการของลูกค้าและองค์กร โดยให้ความสำคัญในการเก็บรักษาข้อมูลและควบคุมการเปลี่ยนแปลงของซอฟต์แวร์

รายละเอียดของ Configuration & Change Management ประกอบด้วย:

1. **Configuration Management (การจัดการและการกำหนดค่า):**

- การจัดทำรายการของซอฟต์แวร์และระบบ ซึ่งรวมถึงรุ่น การติดตั้ง และการกำหนดค่าที่ใช้ในระบบ

- การเก็บรักษาและบริหารจัดการรุ่นของซอฟต์แวร์ รวมถึงการทำความเข้าใจและควบคุมการเปลี่ยนแปลงของรุ่นต่างๆ

2. **Version Control (การควบคุมรุ่น):**

- การติดตามและควบคุมรุ่นของซอฟต์แวร์และระบบ เพื่อให้ทีมพัฒนาสามารถทำงานร่วมกันในรุ่นเดียวกัน

- การจัดเก็บระบบรุ่นก่อนหน้าเพื่อสามารถเรียกใช้และทำการเปรียบเทียบกับรุ่นปัจจุบัน

3. **Change Management (การจัดการและการเปลี่ยนแปลง):**

- การวิเคราะห์และการอนุมัติการเปลี่ยนแปลงของซอฟต์แวร์และระบบ ซึ่งอาจเป็นการเพิ่มความสามารถ การแก้ไขข้อบกพร่อง หรือการปรับปรุงที่ต้องการ

- การควบคุมและบริหารจัดการขอบเขตของการเปลี่ยนแปลง และสิ่งที่ต้องการนำเสนอก่อนที่จะนำเข้าสู่ระบบ

Configuration & Change Management

4. Documentation Management (การจัดการเอกสาร):
   - การบันทึกและเก็บรักษาเอกสารที่เกี่ยวข้องกับการเปลี่ยนแปลงของระบบ ซึ่งอาจเป็นเอกสารการออกแบบ รายงานการทดสอบ รายงานการวิเคราะห์ความต้องการ เป็นต้น
5. Risk Management (การบริหารจัดการความเสี่ยง):
   - การตรวจสอบและประเมินความเสี่ยงที่อาจเกิดขึ้นจากการเปลี่ยนแปลงของระบบ และการกำหนดวิธีการป้องกันและการจัดการกับความเสี่ยงนั้น
6. Release Management (การจัดการการเผยแพร่):
   - การวางแผนและการจัดการการเผยแพร่รุ่นของซอฟต์แวร์และระบบ ซึ่งรวมถึงการทดสอบและการตรวจสอบความพร้อมก่อนการเผยแพร่

การจัดการและการเปลี่ยนแปลงการกำหนดค่าเป็นขั้นตอนสำคัญในกระบวนการพัฒนาระบบซอฟต์แวร์ เพื่อให้ระบบมีความเสถียรและเป็นไปตามความต้องการของลูกค้าและองค์กร และเป็นเครื่องมือสำคัญในการบริหารจัดการโครงการและพัฒนาระบบในสถานการณ์ที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง

References

- **Larman, C. (2001). Applying UML and patterns (2nd ed.). Prentice Hall PTR.**
- **Rational Software Corporation. (1997). Analysis and design with UML.**
- **Jacobson, I. (1998). Applying UML in the unified process.**
- **สมนึก ดีรีโต. (n.d.). ผู้อำนวยการสถาบันนวัตกรรมไอที, ผู้ช่วยศาสตราจารย์, ภาควิชาวิศวกรรมคอมพิวเตอร์, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเกษตรศาสตร์. Authorized SCAMPI Lead Appraiser, Authorized SCAMPI B&C Team Leader. sk@ku.ac.th**

หากสนใจ slide นี้เพื่อการเรียนการสอนโปรดติดต่อ siam2dev@hotmail.com

References

- ดร. สมนึก คีรีโต
  - ผู้อำนวยการ สถาบันนวัตกรรมไอที
  - ผู้ช่วยศาสตราจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์
  - คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

- *Authorized SCAMPI Lead Appraiser*
- *Authorized SCAMPI B&C Team Leader*
- sk@ku.ac.th

ตัวอย่าง การนำ Unified Process มาประยุกต์ใช้กับการพัฒนาระบบงานร้านสะดวกซื้อ

เมื่อนำ Unified Process มาประยุกต์ใช้กับการพัฒนาระบบงานร้านสะดวกซื้อ จะมีขั้นตอนการทำงานเหมือนกับกระบวนการพัฒนาซอฟต์แวร์ทั่วไป โดยแบ่งเป็นขั้นตอนหลัก ๆ ดังนี้:

1. Inception (ขั้นตอนเริ่มต้น)
- วิเคราะห์ความต้องการและความเหมาะสมในการพัฒนาระบบร้านสะดวกซื้อ
- ทำความเข้าใจความต้องการของลูกค้าและธุรกิจในการพัฒนาระบบ
- กำหนดขอบเขตระบบ และกำหนดวัตถุประสงค์ในการพัฒนาระบบ

2. Elaboration (ขั้นตอนในการแก้ไขปัญหา)
- ทำการวิเคราะห์และออกแบบระบบโดยละเอียด
- กำหนดสถาปัตยกรรมของระบบและเทคโนโลยีที่ใช้
- ทำการสร้างพื้นฐานของระบบ และเตรียมความพร้อมในการพัฒนา

3. Construction (ขั้นตอนการก่อสร้าง)
- เริ่มทำการพัฒนาระบบสะดวกซื้อตามที่ออกแบบมาในขั้นตอน Elaboration
- ทดสอบและตรวจสอบความถูกต้องของระบบ

4. Transition (ขั้นตอนการเปลี่ยนแปลง)
- นำระบบสะดวกซื้อที่พัฒนามาใช้งานจริง
- ดำเนินการทดสอบและปรับปรุงระบบตามความต้องการของลูกค้าและธุรกิจ

โดยในแต่ละขั้นตอน จะมีการเปลี่ยนแปลงและปรับปรุงของระบบตามความต้องการ ซึ่งการแยกการพัฒนาเป็นขั้นตอนและทำแบบก้าวการพัฒนา (Iteration) ทำให้ทีมพัฒนาสามารถแก้ไขปัญหาและปรับปรุงระบบในขั้นตอนต่าง ๆ ได้โดยทันสมัย และเปลี่ยนแปลงไปตามความเหมาะสมของโครงการและสภาพแวดล้อมการทำงาน หนึ่งในสิ่งที่แตกต่างจากกระบวนการพัฒนาอื่น ๆ คือ UP มีการให้ความสำคัญกับการตรวจสอบคุณภาพและการทดสอบซอฟต์แวร์ ในแต่ละขั้นตอนเพื่อให้ได้ระบบที่มีคุณภาพและสามารถสร้างมูลค่าให้กับธุรกิจและลูกค้าได้อย่างเหมาะสม

ตัวอย่าง การนำ Unified Process มาประยุกต์ใช้กับการพัฒนาระบบบริหารศูนย์บริการรถยนต์มีดังนี้:

เมื่อนำ Unified Process มาประยุกต์ใช้กับการพัฒนาระบบบริหารศูนย์บริการรถยนต์ทำให้การพัฒนาเป็นกระบวนการที่มีขั้นตอนชัดเจนและเป็นระบบ ตัวอย่างขั้นตอนการนำ Unified Process มาประยุกต์ใช้งานในการพัฒนาระบบบริหารศูนย์บริการรถยนต์ได้แก่:

1. Inception Phase (เริ่มต้น):
   - ศึกษาและวิเคราะห์ความต้องการของระบบบริหารศูนย์บริการรถยนต์ เช่น การจัดการนัดหมาย การบริหารจัดการคิว ระบบติดตามสถานะรถยนต์ การแจ้งเตือนลูกค้า เป็นต้น
   - กำหนดเป้าหมายและขอบเขตของโครงการพัฒนาระบบ
   - ประเมินความเสี่ยงและการตอบสนองต่อความต้องการของระบบ
2. Elaboration Phase (ขยายความเข้าใจ):
   - ออกแบบระบบบริหารศูนย์บริการรถยนต์รายละเอียด ซึ่งอาจประกอบด้วยแผนผังฐานข้อมูล แผนภูมิกระบวนการ และอินเตอร์เฟสผู้ใช้งาน
   - สร้างโมเดลระบบและออกแบบข้อมูลที่เกี่ยวข้องกับระบบ เช่น ข้อมูลลูกค้า ข้อมูลรถยนต์ การบันทึกประวัติซ่อมแซม เป็นต้น
   - กำหนดสถาปัตยกรรมและเทคโนโลยีที่ใช้ในการพัฒนาระบบ
3. Construction Phase (การสร้าง):
   - เขียนโค้ดและพัฒนาระบบบริหารศูนย์บริการรถยนต์ตามการออกแบบและข้อมูลที่กำหนด
   - ทดสอบระบบเพื่อตรวจสอบความถูกต้องและประสิทธิภาพ ตรวจสอบการทำงานของระบบร่วมกับสถานะการดำเนินการของรถยนต์ตามหลังร้าน
   - แก้ไขและปรับปรุงระบบให้สอดคล้องกับความต้องการและสถานะการดำเนินการของรถยนต์
4. Transition Phase (การเปลี่ยนแปลง):
   - ปรับปรุงและแก้ไขระบบหากมีข้อเสนอแนะหลังจากการทดสอบและการใช้งานจริง
   - ฝึกอบรมผู้ใช้งานเกี่ยวกับการใช้งานระบบ
   - นำระบบบริหารศูนย์บริการรถยนต์ไปใช้งานจริงและดูแลรักษาระบบเพื่อให้มีประสิทธิภาพในการบริหารจัดการร้านบริการรถยนต์

การนำ Unified Process มาประยุกต์ใช้กับการพัฒนาระบบบริหารศูนย์บริการรถยนต์ทำให้ทีมพัฒนามีขั้นตอนและเครื่องมือในการกำหนดขอบเขต วางแผน ออกแบบ พัฒนา และประเมินผลการพัฒนาระบบ ช่วยให้การพัฒนาระบบมีความเป็นระบบ

ตัวอย่าง การนำ Unified Process มาประยุกต์ใช้กับการพัฒนาระบบบริหารศูนย์บริการรถยนต์มีดังนี้:

การนำ Unified Process (UP) มาประยุกต์ใช้ในการพัฒนาระบบบริหารศูนย์บริการรถยนต์สามารถแบ่งเป็นขั้นตอนต่อไปนี้:

1. Inception (เบื้องต้น):
   - กำหนดวัตถุประสงค์และขอบเขตของระบบบริหารศูนย์บริการรถยนต์
   - วิเคราะห์ความเสี่ยงและประโยชน์ที่คาดหวัง
   - สร้าง Business Case เพื่อเปรียบเทียบความคุ้มค่าของโปรเจกต์
2. Elaboration (ขยายความเข้าใจ):
   - พัฒนา Use Case Model เพื่อระบุความต้องการของผู้ใช้
   - ออกแบบและพัฒนา Prototype เพื่อสร้างความเข้าใจในการทำงานของระบบ
   - วางแผนการทดสอบและทำความเข้าใจถึงเทคโนโลยีที่จะถูกนำมาใช้
3. Construction (ก่อสร้าง):
   - พัฒนาระบบบริหารศูนย์บริการรถยนต์ตาม Use Case Model และ Prototype
   - ทดสอบและปรับปรุงระบบตามความต้องการ
   - สร้างเอกสารทางเทคนิคและคู่มือการใช้งาน
4. Transition (การเปลี่ยนแปลง):**
   - ทดสอบระบบในสภาพแวดล้อมที่จริง
   - จัดอบรมและเตรียมพร้อมสำหรับการใช้งานจริง
   - ปรับปรุงระบบตามข้อเสนอแนะและความคิดเห็นจากผู้ใช้
5. **Iterations (การทำซ้ำ):**
   - ทำการวิจารณ์และปรับปรุงระบบในทุกรอบของ UP
   - พัฒนาความเข้าใจและปรับปรุงระบบตามความต้องการที่เปลี่ยนแปลง
   - ให้ความสำคัญกับการพัฒนาแบบ iterative เพื่อให้สามารถปรับปรุงระบบได้อย่างยืดหยุ่น
6. **Use Case-driven Development:**
   - ใช้ Use Case Model เป็นแนวทางในการพัฒนาระบบ

ตัวอย่าง การนำ Unified Process มาประยุกต์ใช้กับการพัฒนาระบบบริหารศูนย์บริการรถยนต์มีดังนี้:

ขั้นตอน Inception Phase สำหรับระบบบริหารศูนย์บริการรถยนต์ เป็นขั้นตอนแรกในกระบวนการพัฒนาและเป็นการเริ่มต้นในการวางแผนและตั้งเป้าหมายของโครงการ ดังนี้:

1. วิเคราะห์ความต้องการ (Requirements Analysis):
   - สำรวจและวิเคราะห์ความต้องการของร้านบริการรถยนต์ ซึ่งอาจประกอบด้วยความต้องการของผู้ใช้งาน เช่น ระบบจองนัดหมาย การทำรายงานสถานะรถยนต์ที่ซ่อมแซม การติดตามสถานะรถยนต์ ระบบติดต่อกับลูกค้า เป็นต้น
   - วิเคราะห์ความต้องการของระบบเพื่อให้เข้าใจความเป็นไปได้และการพัฒนาที่เหมาะสม

2. กำหนดเป้าหมายและขอบเขตของโครงการ (Project Goals and Scope):
   - กำหนดเป้าหมายของโครงการว่าต้องการสร้างระบบบริหารศูนย์บริการรถยนต์ขึ้นมาเพื่อให้บริการลูกค้าอย่างมีประสิทธิภาพและครอบคลุมความต้องการของลูกค้า
   - กำหนดขอบเขตของโครงการเพื่อจำกัดขอบเขตของระบบและการพัฒนาในโครงการนี้

3. วิเคราะห์ความเป็นไปได้และความเสี่ยง (Feasibility and Risk Analysis):
   - วิเคราะห์ความเป็นไปได้ของโครงการว่าสามารถพัฒนาระบบที่ต้องการได้หรือไม่ โดยพิจารณาเรื่องทางเทคนิค ทรัพยากร และเทคโนโลยีที่ใช้
   - ประเมินความเสี่ยงในการพัฒนาโครงการ เช่น ความเสียหายทางเศรษฐกิจ ความเสี่ยงในเทคโนโลยี การตอบสนองของลูกค้า เป็นต้น

4. **กำหนดแนวทางและแผนการทำงาน (Project Direction and Plan):**
   - กำหนดแนวทางในการพัฒนาระบบ และออกแบบแผนการทำงานของทีมพัฒนา
   - วางแผนการทำงานในการพัฒนาและการทดสอบระบบบริหารศูนย์บริการรถยนต์ รวมถึงระยะเวลาที่ใช้ในแต่ละขั้นตอนของโครงการ

5. **เตรียมความพร้อมในการเริ่มโครงการ (Prepare for Project Kickoff):**
   - จัดทำเอกสารและรายงานที่เกี่ยวข้องกับโครงการ เช่น รายงานการวิเคราะห์ความต้องการ แผนการทำงาน และรายงานความเป็นไปได้และความเสี่ยง
   - นำเสนอแผนการพัฒนาและข้อมูลต่างๆ ให้กับทีมพัฒนาและผู้ที่เกี่ยวข้องในโครงการ

ขั้นตอน Inception Phase ช่วยให้ทีมพัฒนามีความเข้าใจและความเหมาะสมในการพัฒนาระบบบริหารศูนย์บริการรถยนต์ และกำหนดเป้าหมายที่ชัดเจนในโครงการ เพื่อให้การพัฒนาเป็นไปอย่างมีประสิทธิภาพและสอดคล้องกับความต้องการของลูกค้าและเป้าหมายขององค์กร

แบบฝึกหัด 3.1 จงตอบคำถามต่อไปนี้

จงบอกชิ้นงาน ของแต่ละ worker ต่อไป มาอย่างน้อยคนละ 3 ชิ้นงาน

- **SA .....**
- **Programmer .....**
- **PM .....**
- **UX/UI Mockup ของหน้า Login**
- **Tester .....**

แบบฝึกหัด 3.1 จงตอบคำถามต่อไปนี้

1. SA (Software Architect):
   - ออกแบบและวางโครงสร้างระบบใหม่เพื่อเพิ่มประสิทธิภาพ
   - พัฒนาวิธีการเชื่อมต่อระบบเดิมกับระบบใหม่
   - ทำความเข้าใจความต้องการของลูกค้าและแปลงเป็นแผนการทำงานของระบบ
2. Programmer:
   - เขียนโปรแกรมส่วนหนึ่งของระบบใหม่ที่ได้รับมอบหมาย
   - แก้ไขบั๊กและปรับปรุงโค้ดตามคำแนะนำจาก QA
   - ทดสอบโปรแกรมเพื่อตรวจสอบความถูกต้องและประสิทธิภาพ
3. PM (Project Manager):
   - วางแผนโปรเจกต์และกำหนดส่วนงานแต่ละอันในทีม
   - ติดตามความคืบหน้าของโปรเจกต์และแก้ไขปัญหาที่เกิดขึ้น
   - ประชุมกับทีมเพื่อรับคำแนะนำและติดตามแผนการทำงาน
4. **UX/UI Mockup ของหน้า Login:**
   - ออกแบบ UX/UI สำหรับหน้า Login ของแอปพลิเคชัน
   - ทำการโมดัพ (Mockup) เพื่อแสดงภาพตัวอย่างของหน้า Login
   - ปรับปรุงตามความคิดเห็นจากทีมและผู้ใช้
5. **Tester:**
   - ทดสอบฟังก์ชันต่าง ๆ ของระบบ
   - รายงานบั๊กและสรุปผลการทดสอบให้กับทีมพัฒนา
   - จัดทำแผนการทดสอบและดูแลความถูกต้องของระบบก่อนการนำสู่การใช้งานจริง

โปรเจกต์นี้แบ่งงานออกเป็นหลายส่วน เพื่อให้ทุกรายที่เกี่ยวข้องมีส่วนร่วมในการพัฒนาและทดสอบระบบให้เสร็จสมบูรณ์และมีประสิทธิภาพตามที่กำหนดไว้ในโปรเจกต์.

แบบฝึกหัด 3.2 จงตอบคำถามต่อไปนี้

1. จงบอกชิ้นงาน ของระบบต่อไปนี้

- **จงบอกชิ้นงาน ของการพัฒนาระบบ สั่งซื้อสินค้า(POS) ..**
- **จงบอกชิ้นงาน ของการพัฒนาระบบ เวชระเบียนของโรงพยาบาล...**
- **จงบอกชิ้นงาน ของการพัฒนาระบบ ระบบคาร์แคร์(CarCare)...**
- **จงบอกชิ้นงาน ของการพัฒนาระบบ ระบบคลังสินค้า (warehouse)...**

2. จงตอบคำถามต่อไปนี้

2.1 การแบ่งยูนิตงานให้แต่ละคนเป็นผู้รับผิดชอบ สิ่งที่ควรคำนึงถึงคืออะไรบ้าง

2.2 จงอธิบายรูปลักษณ์ ของ UP มาพอสังเขป

2.3 ให้อธิบายหรือวิเคราะห์ระบบ การสั่งซื้อสินค้า ด้วยหลักการวนรอบเพิ่มพูน (Iterative and Incremental)