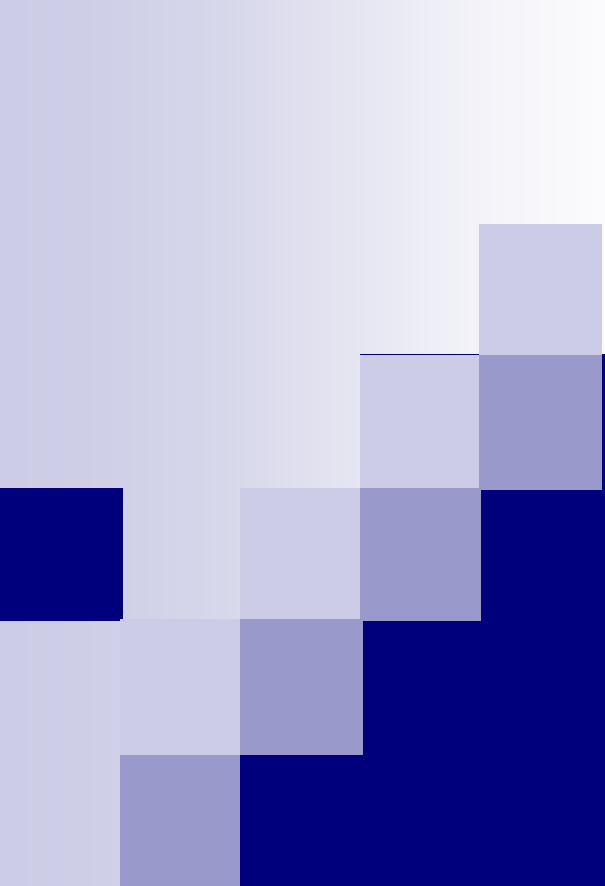


คำสั่งที่ 2

การพัฒนาระบบสารสนเทศ

(Information System Development)



Software in real life



Software in real life

- System software
- AI software
- Data warehouseing
- Application software
- Web Applications
- Wireless networks
- Mobile computing

วงจรการพัฒนาาระบบ

(System Development Life Cycle: SDLC)

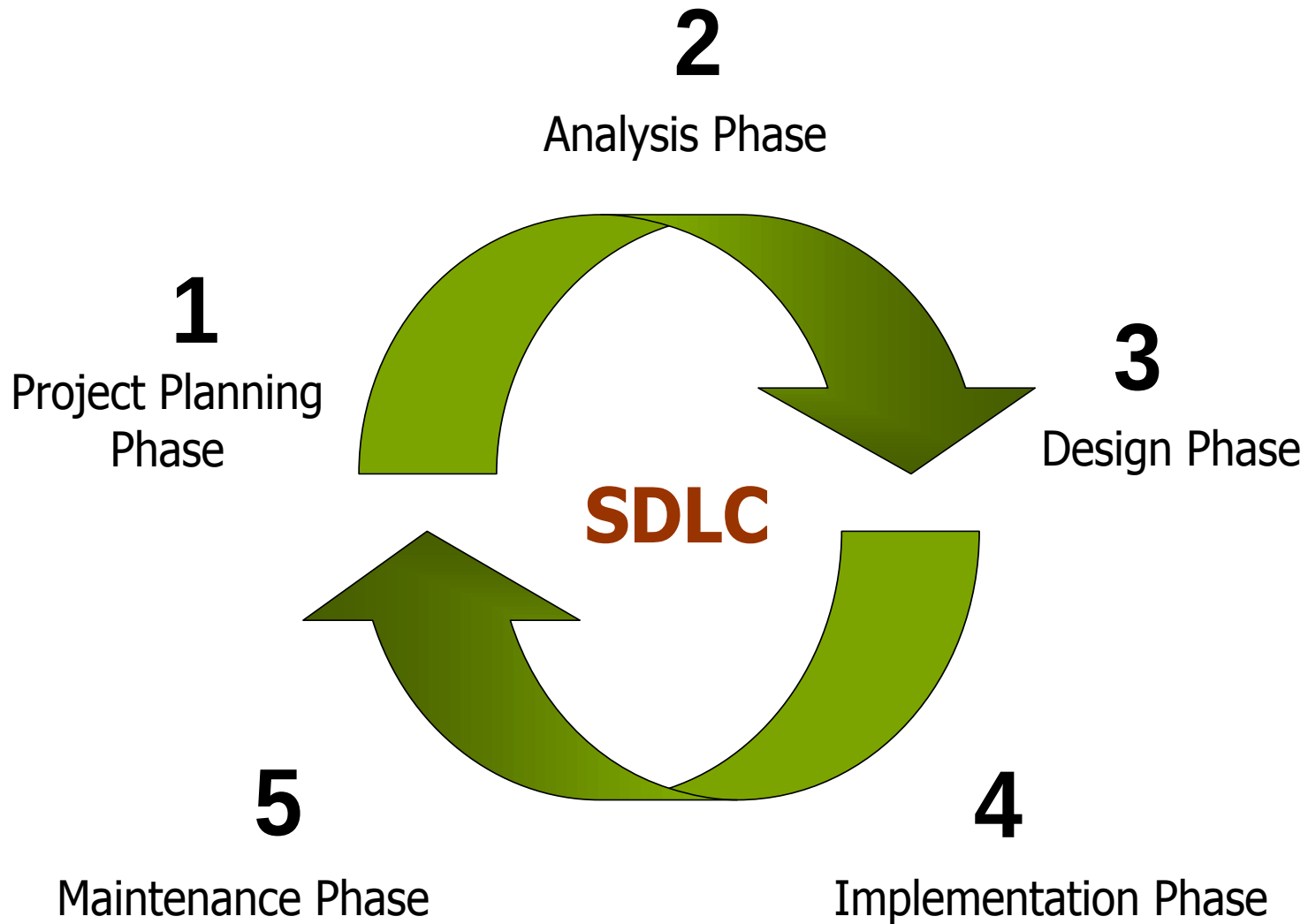
ระยะที่ 1: การวางแผนโครงการ

ระยะที่ 2: การวิเคราะห์

ระยะที่ 3: การออกแบบ

ระยะที่ 4: การนำไปใช้

ระยะที่ 5: การบำรุงรักษา



1. Project Planning Phase

SA เข้าไปดูระบบเพื่อศึกษาความเป็นไปได้
ของระบบว่าระบบที่พัฒนาขึ้นมาใหม่นั้น
สามารถแก้ไขปัญหานั้นได้หรือไม่

2. Analysis Phase

ระยะการวิเคราะห์ว่าใคร (Who) เป็นผู้ใช้ระบบ
และมีอะไรบ้าง (What) ที่ระบบต้องทำ

เป็นการรวบรวมความต้องการ (Requirements
Gathering) จากผู้ใช้ (User Requirements)

สรุปเป็นข้อกำหนดที่ชัดเจน และพัฒนา
แบบจำลองกระบวนการ (Process Model)

3.Design Phase

ระยะการการออกแบบเป็นการพิจารณาว่าระบบ
จะดำเนินการไปได้อย่างไร (How)

มุ่งเน้นถึงการดำเนินการแก้ปัญหา ด้วยการนำ
ผลลัพธ์จากแบบจำลอง Logical Model ที่ได้จากการ
วิเคราะห์มาพัฒนาเป็นแบบจำลองทาง Physical Model

4.Implementation Phase

ในระยะการนำไปใช้ จะทำให้ระบบเกิดผลขึ้นมา
ด้วยการสร้างระบบ ทดสอบระบบ และการติดตั้งระบบ
รวมถึงการฝึกอบรมเพื่อใช้งานระบบ

5. Maintenance Phase

ระยะการบำรุงรักษา สามารถเพิ่มเติมคุณสมบัติระบบให้มีประสิทธิภาพสูงขึ้น รวมถึงการแก้ไขปรับปรุงโปรแกรมในกรณีที่เกิดข้อผิดพลาดที่พบบ่อย และการเขียนโมดูลการทำงานเพิ่มเติม

System Development Methodology

1. Models

- Flowchart
- Data Flow Diagram
- ER Diagram
- Structure Chart
- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Gantt Chart/PERT

System Development Methodology

2. Tools

เครื่องมือที่ช่วยในการพัฒนาระบบประกอบด้วย software หรือ Program ที่ช่วยสนับสนุนการใช้งาน เครื่องมือเหล่านี้สามารถนำมาใช้งานเพื่อสร้างแบบจำลองหรือโมเดลต่าง ๆ

System Development Methodology

3. Techniques

คือกลุ่มแนวทางที่ช่วยในการชี้แนะ (Guidelines) เพื่อให้ นักวิเคราะห์ระบบนำเทคนิคไปดำเนินกิจกรรมการพัฒนา ระบบเพื่อให้เกิดความสมบูรณ์ยิ่งขึ้น

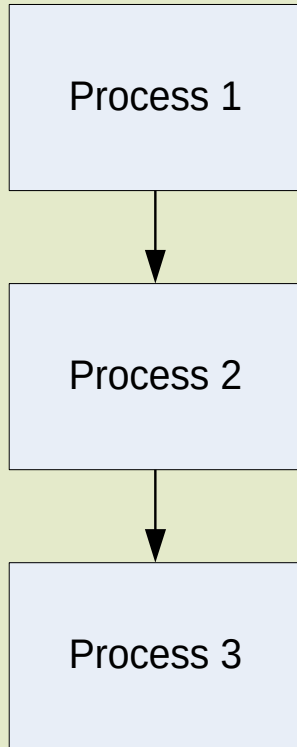
วิธีการพัฒนาระบบ

1. การพัฒนาระบบแบบดั้งเดิม
2. การพัฒนาระบบเชิงวัตถุ

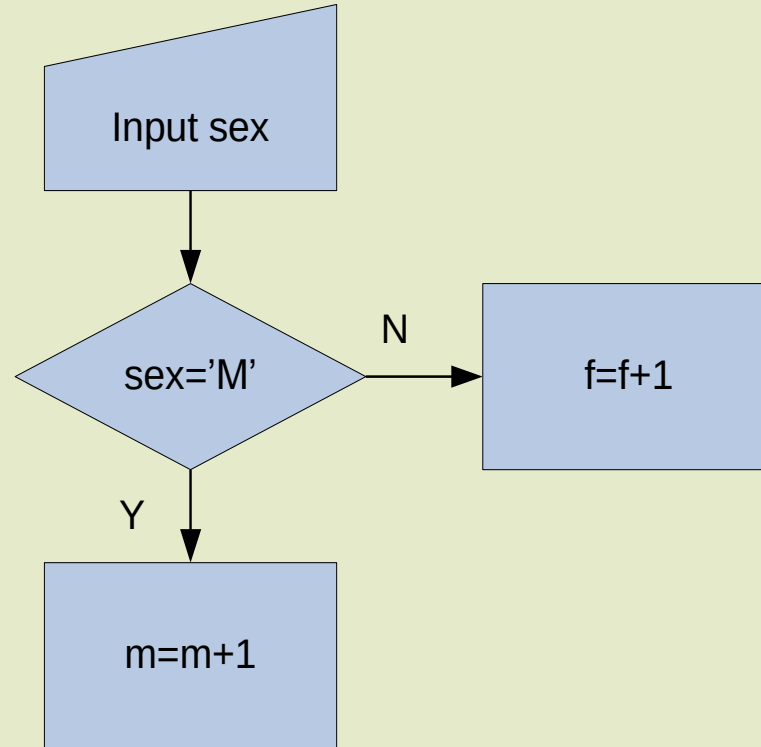
การพัฒนาาระบบแบบดั้งเดิม

1. การโปรแกรมเชิงโครงสร้าง (Structured Programming)

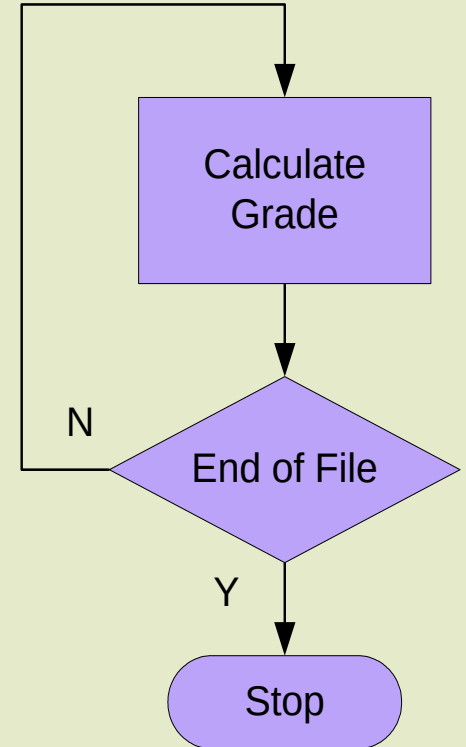
มีวัตถุประสงค์เพื่อให้เกิดความเป็นไปได้ในการ
รวมส่วนโปรแกรมต่าง ๆ ที่แยกจากกันนี้เข้าไปในระบบ
สารสนเทศที่มีความซับซ้อนมากยิ่งขึ้นได้ และในส่วนของ
เทคนิคการวิเคราะห์เชิงโครงสร้างจะช่วยอธิบายความ
ต้องการให้มีความชัดเจนยิ่งขึ้น



(a) Sequence



(b) Decision



(c) Repetition

```
graph TD; CM[Control Module] --> M1[Module 1]; CM --> M2[Module 2]; CM --> M3[Module 3];
```

Control Module
start
call module 1
call module 2
call module 3
stop

Module 1
begin
do 1
do 2
do 3
return

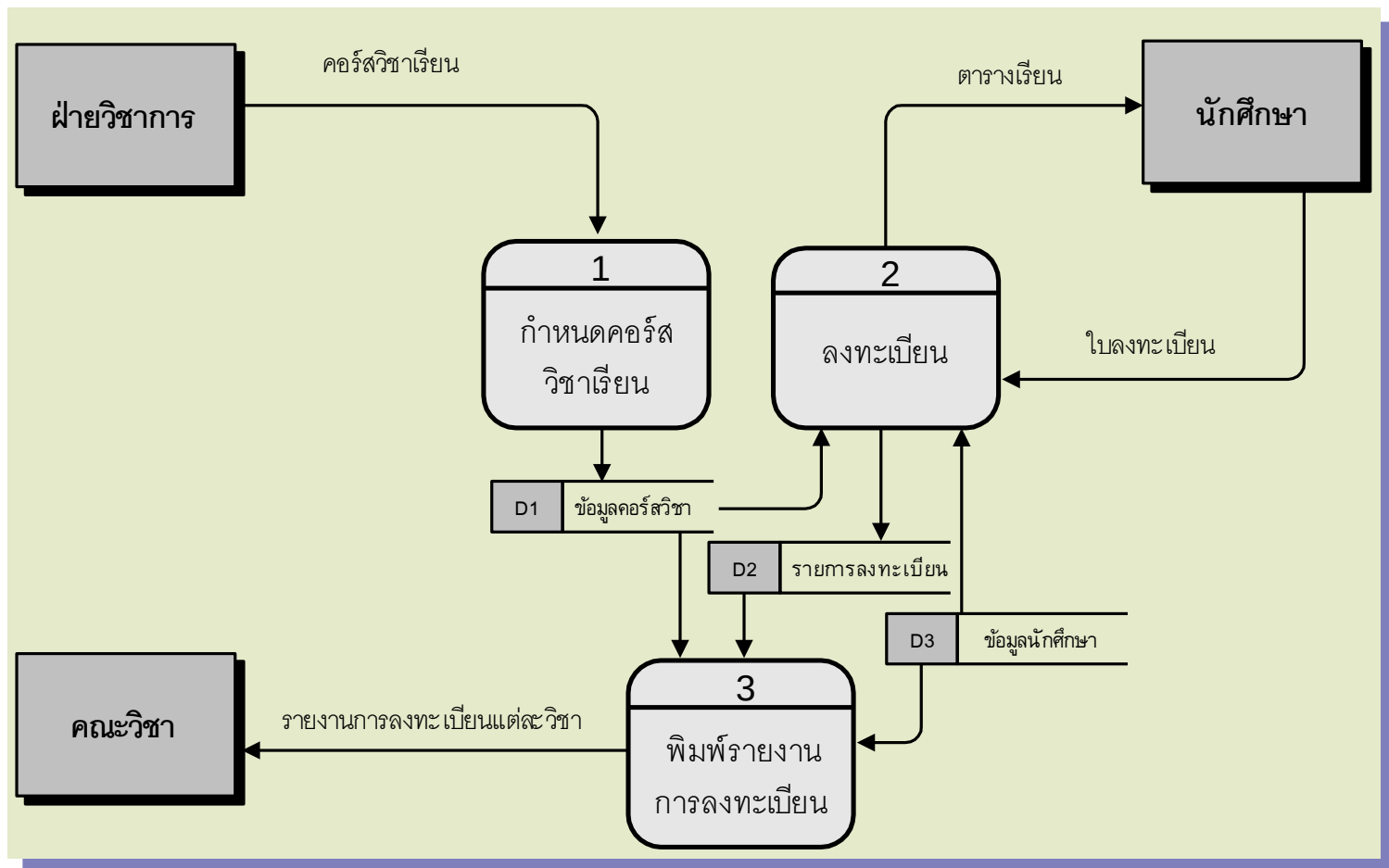
Module 2
begin
do x
do y
do z
return

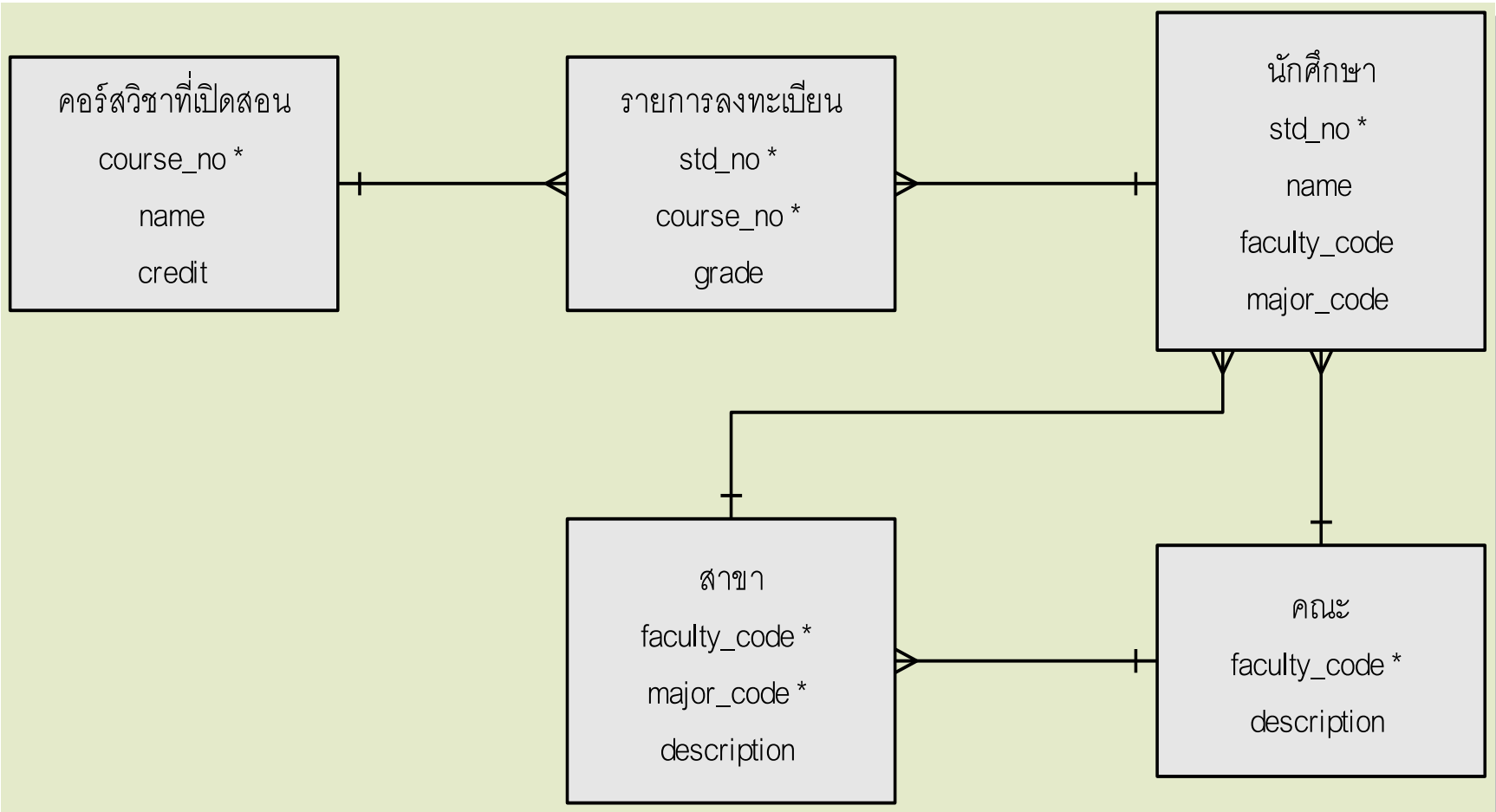
Module 3
begin
if x then y
else z
do abc
return

2. การออกแบบเชิงโครงสร้าง (Structured Design)

เทคนิคการออกแบบเชิงโครงสร้างได้พัฒนาขึ้น
เพื่อเป็นแนวทางในการตัดสินใจว่ามีกลุ่มของโปรแกรม
อะไรบ้างที่ควรจะมี มีโปรแกรมอะไรที่ได้ทำเสร็จ และ
โปรแกรมมีรูปแบบการจัดการเป็นลำดับชั้นอย่างไร
สามารถแสดงได้ในลักษณะของ Structure Chart

การวิเคราะห์และออกแบบเชิง โครงสร้างสมัยใหม่





การพัฒนาาระบบเชิงวัตถุ

1. การวิเคราะห์ระบบเชิงวัตถุ

(Object-Oriented Analysis : OOA)

เป็นการกำหนดวัตถุหรือ object ต่าง ๆ ที่ต้องมีในระบบ และแสดงถึง object แต่ละตัวจะโต้ตอบกันอย่างไรเพื่อให้งานใดงานหนึ่งเสร็จสมบูรณ์

2. การออกแบบระบบเชิงวัตถุ

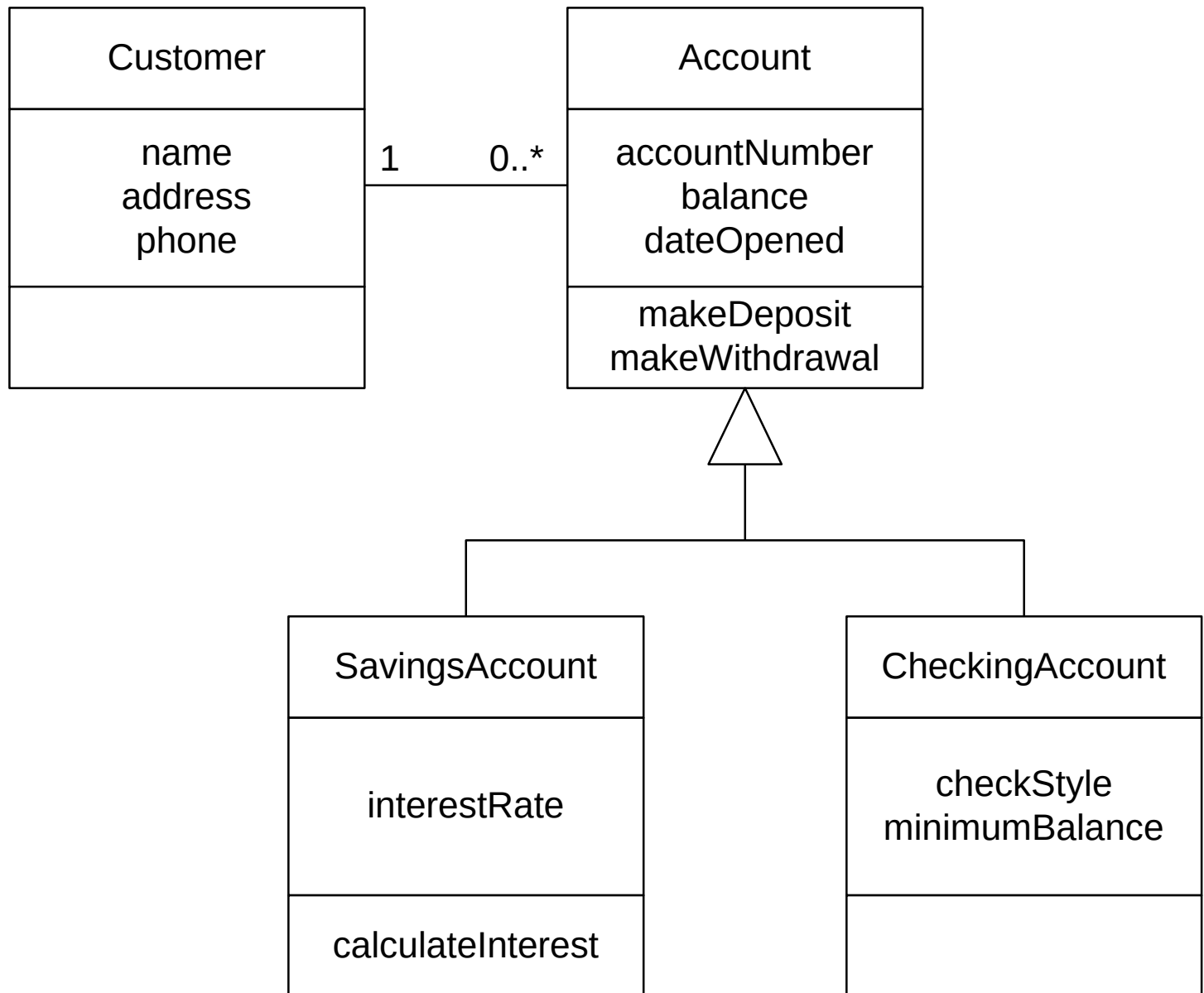
(Object-Oriented Design : OOD)

หลังจากที่ได้ทำการวิเคราะห์และได้โมเดลเชิงวัตถุที่สามารถนำมาใช้เป็นหลักในการออกแบบ จากนั้นก็ดำเนินการกำหนดชนิดของ object เพิ่มเติมที่มีส่วนสำคัญในการสื่อสารกับมนุษย์และอุปกรณ์ในระบบ เพื่อการกำหนดรายละเอียดของภาษาการเขียนโปรแกรมเชิงวัตถุ

3. การเขียนโปรแกรมเชิงวัตถุ

(Object-Oriented Programming : OOP)

เป็นการเขียนชุดคำสั่ง โดยโปรแกรมที่เขียนขึ้นจะถูกจัดให้มีโครงสร้างการทำงานร่วมกันของวัตถุต่าง ๆ





โมเดลการพัฒนาซอฟต์แวร์

(Software Development Process Models)

จัดเป็นกรรมวิธีการพัฒนาซอฟต์แวร์
(Methodology) ที่สามารถนำมาประยุกต์
ใช้เพื่อเป็นแนวทางการพัฒนาซอฟต์แวร์
ตั้งแต่เริ่มต้นจนกระทั่งสำเร็จ



โมเดลการพัฒนาซอฟต์แวร์

1. Build-and-Fix Model
2. Water Fall Model
3. Incremental Model
4. Spiral Model
5. Rapid Application Development (RAD)
6. Joint Application Development (JAD)
7. Rational Unified Process (RUP)

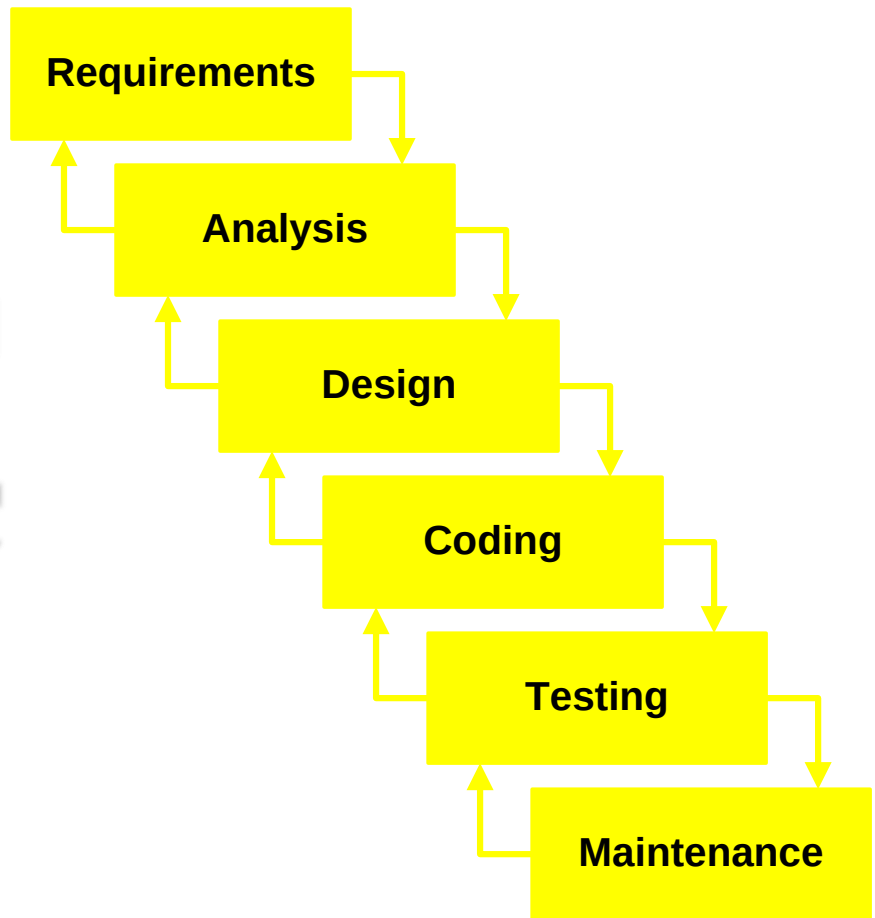
Build and Fix Model

พัฒนาแบบลองผิด
ลองถูกไปเรื่อย ๆ
จนกระทั่งคิดว่าพอใจ
หรือว่าคิดว่าตรงกับ
ความต้องการแล้ว



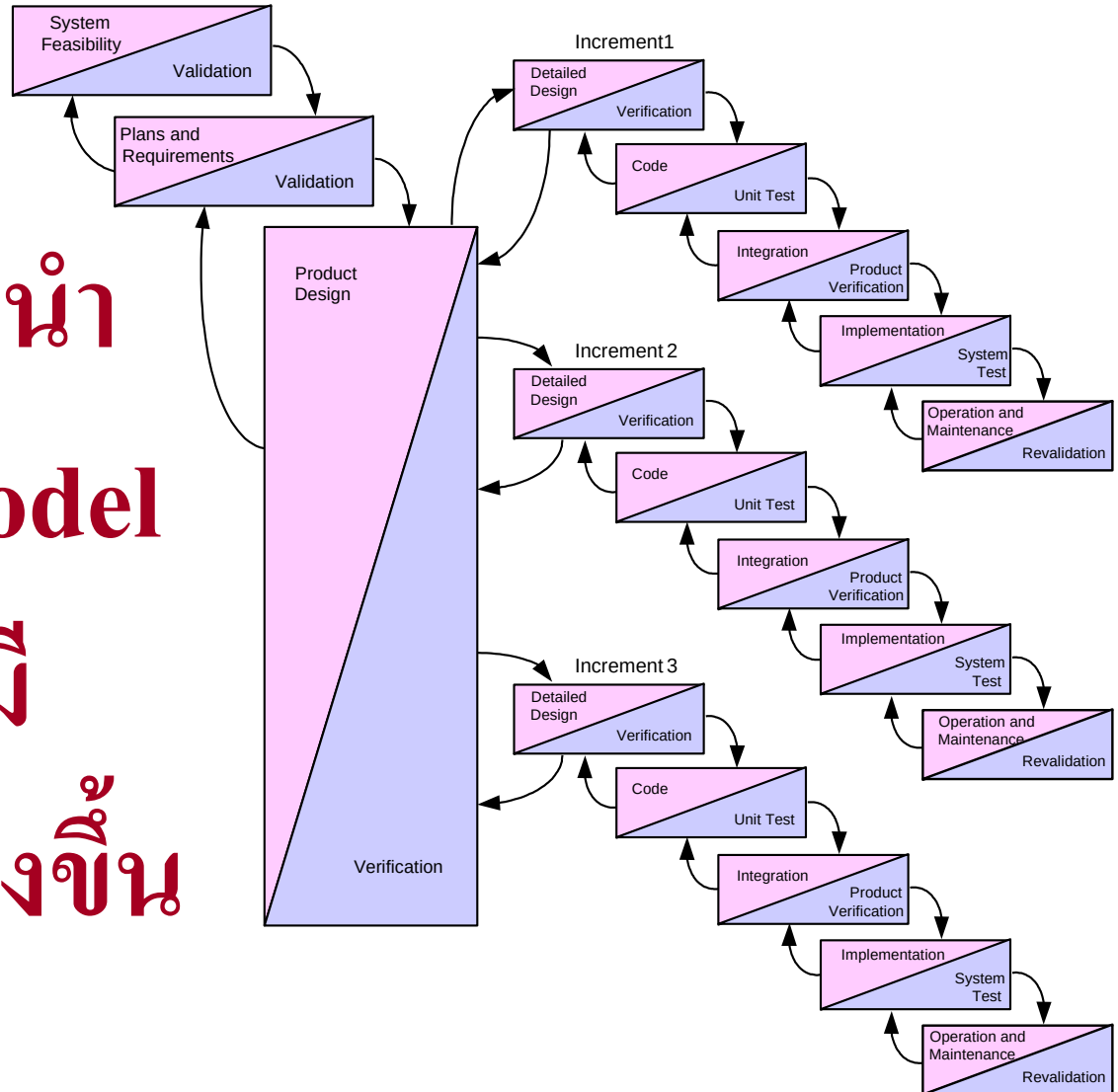
Water Fall Model

หรือเรียกว่า
โมเดลน้ำตก มีความ
คล้ายคลึงกับ SDLC
แต่เพิ่มคุณสมบัติ
แบบ Iteration



Incremental Model

เป็นโมเดลที่นำ
Water Fall Model
มาปรับปรุงให้มี
ประสิทธิภาพยิ่งขึ้น



Incremental Model (ต่อ)

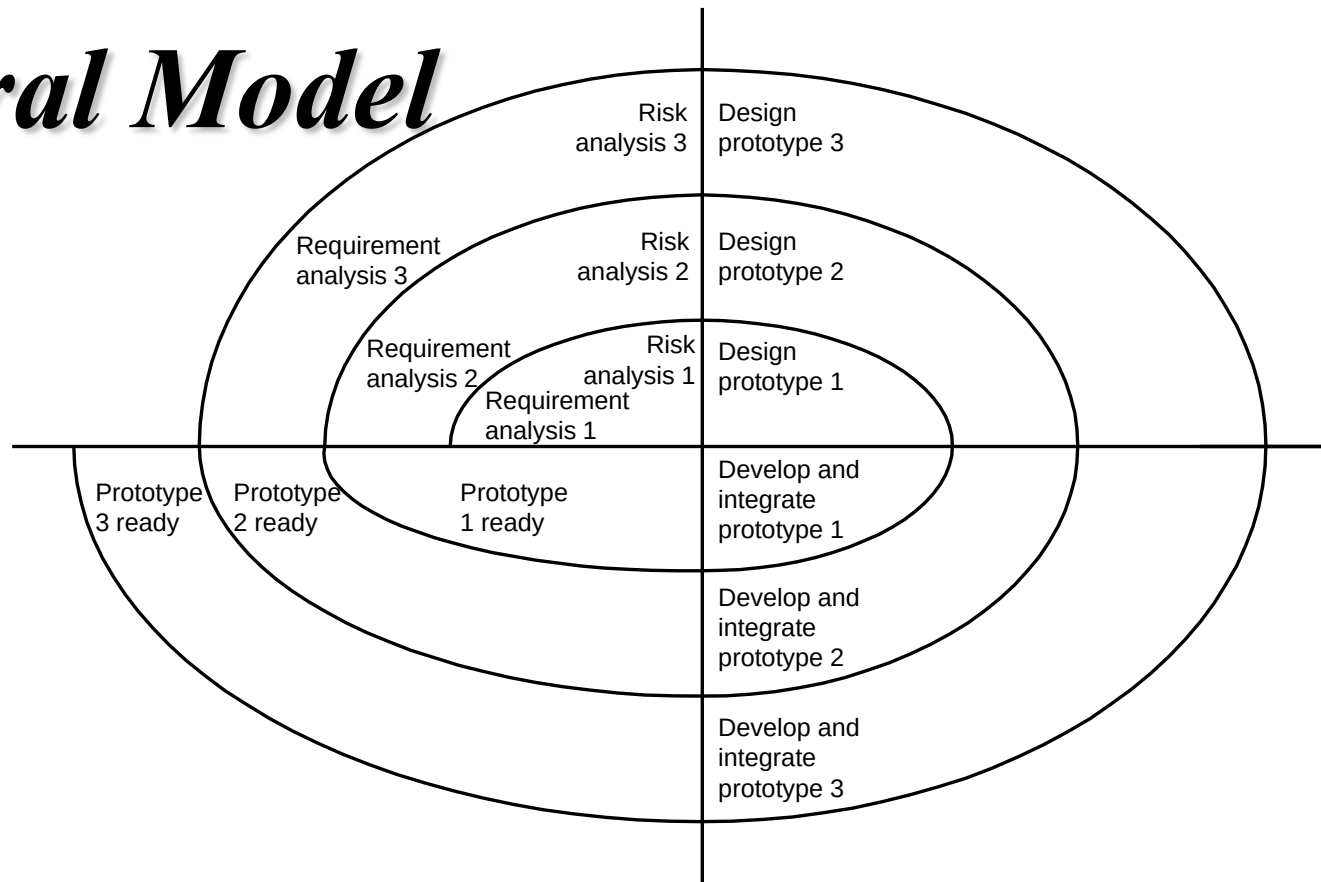
เพิ่มคุณสมบัติแบบ Incremental
เข้าไป โดยแต่ละรอบจะมีการทวนซ้ำ
พร้อมระบบการตรวจสอบ

Incremental Model (ต่อ)

Verification คือตรวจสอบความ
ถูกต้องตามข้อกำหนด (Specification)

Validation คือการตรวจสอบผลิตภัณฑ์
ว่าตรงความต้องการของผู้ใช้งานหรือไม่

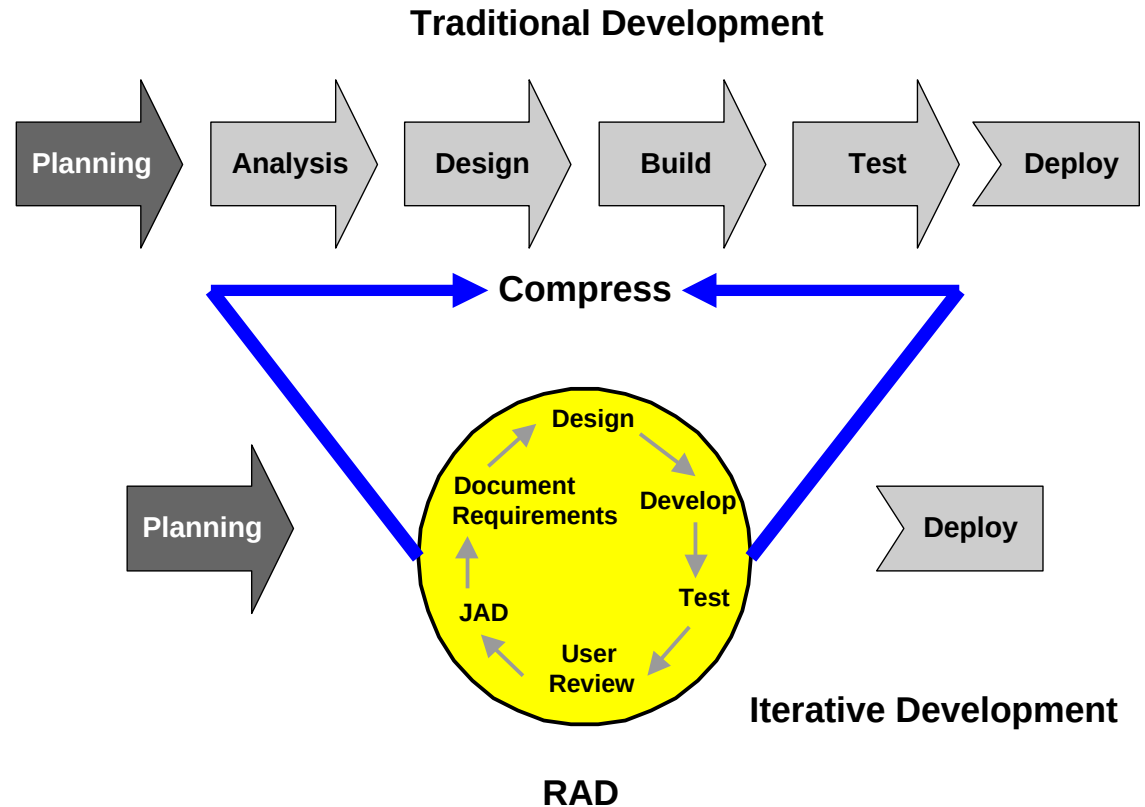
Spiral Model



มีหลักการทำงานในลักษณะรอบวงกลม

โดยวนจากวงในสู่วงนอก

Rapid Application Development (RAD)

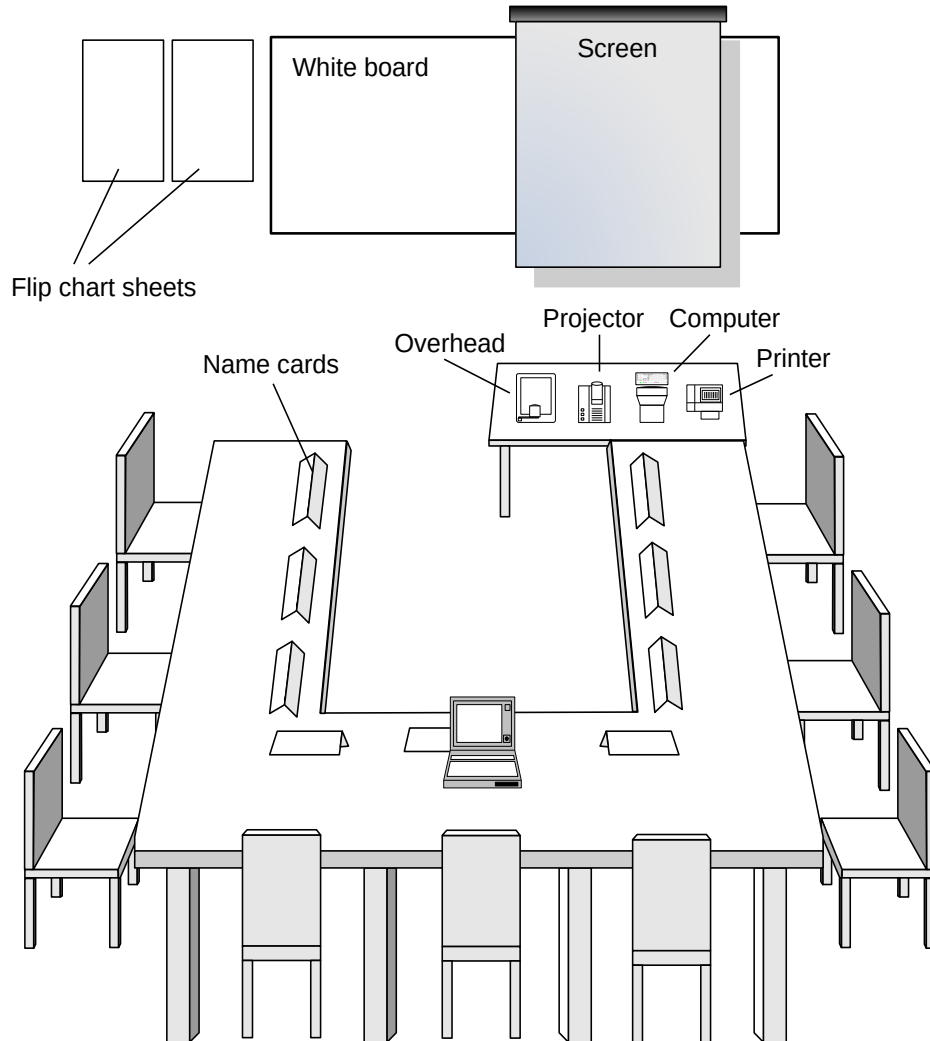


เป็นการพัฒนาแอปพลิเคชันแบบรวดเร็ว

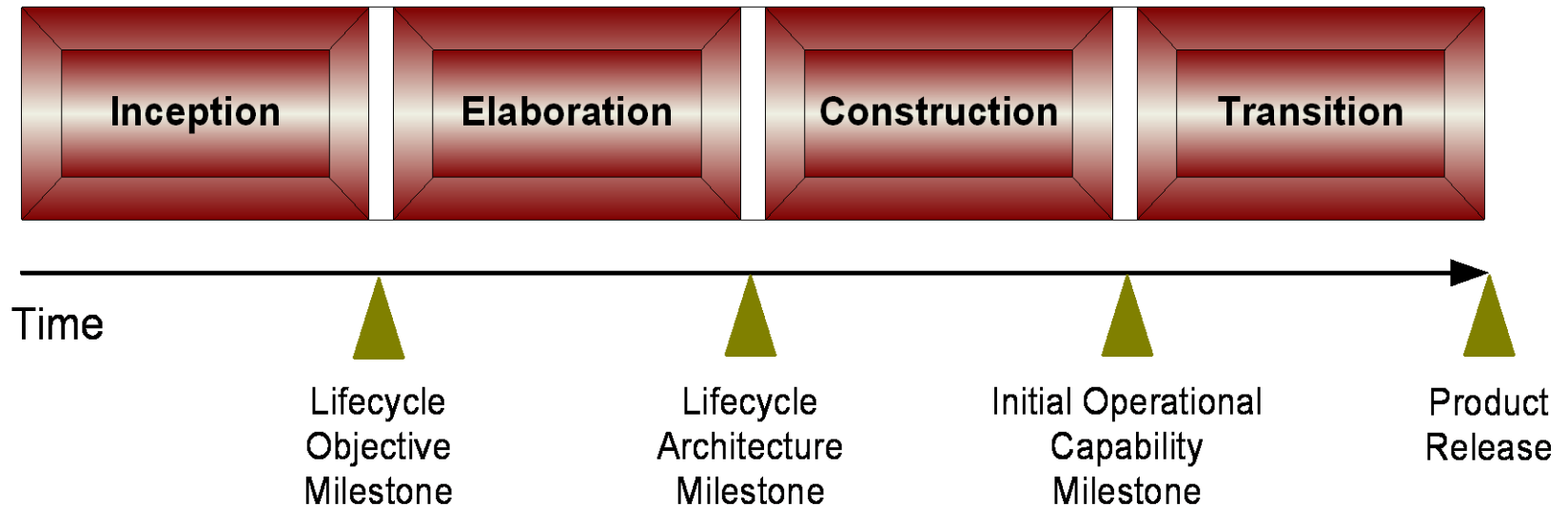
มีทีมงานขนาดเล็กที่มีความเชี่ยวชาญสูง

Joint Application Development (JAD)

มีห้องปฏิบัติการที่ใช้
เป็นศูนย์การทำงาน เน้น
การประชุม Workshop
ทีมงานตระหนักใน
หน้าที่ และพร้อมที่จะ
ทำงานหนัก



Ration Unified Process (RUP)



เป็นกรรมวิธีการพัฒนา S/W เชิงวัตถุประสงค์
พื้นฐานสำคัญคือการสร้างโมเดล และการ
จัดการด้วยภาษา UML

Extreme Programming (XP)

- เป็นแนวความคิดโดย Kent Beck [1999] โดยเน้นกระบวนการผลิตซอฟต์แวร์ที่ใช้แนวทางเชิงวัตถุเป็นหลัก แบ่งการทำงานออกเป็น 4 ขั้นตอน ดังต่อไปนี้

User Story
Iteration Plan

Simple Design
Spike Solution: Prototype

วางแผน
(Planning)

ออกแบบ (Design)

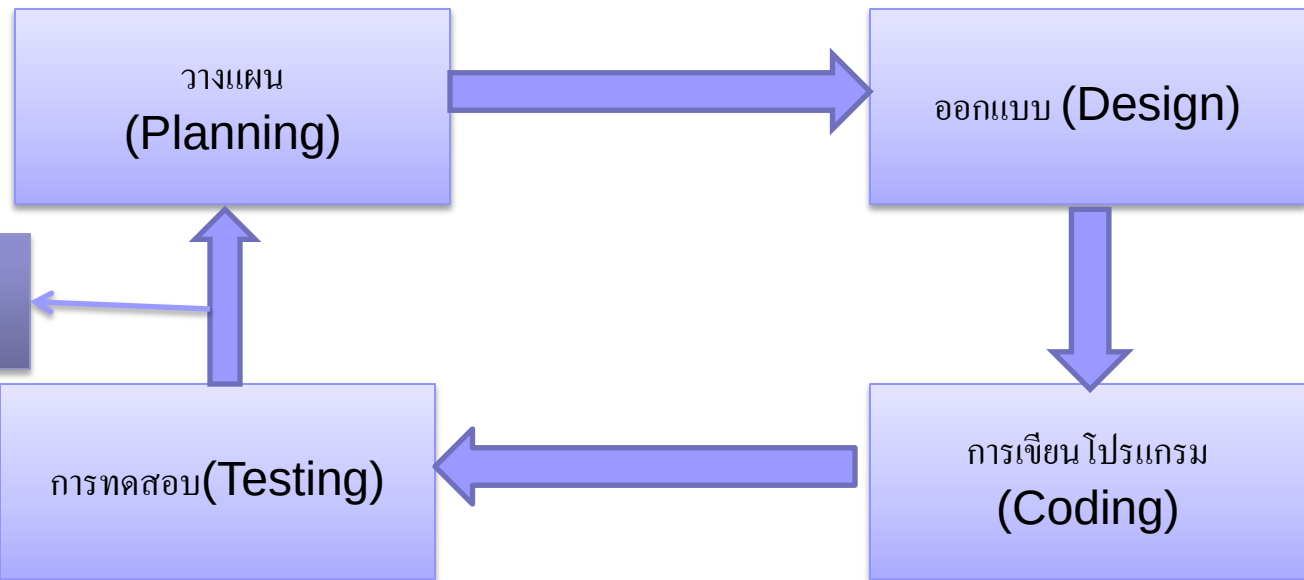
Release
Software Increment

การทดสอบ (Testing)

การเขียนโปรแกรม
(Coding)

Unit test ,Continuous Integration
Acceptance Test

Pair Programming
Unit test ,Continuous Integration



Extreme Programming (XP)

■ การวางแผน Planning

- ☐ เก็บรวบรวมข้อมูลความต้องการและวิเคราะห์ความต้องการของผู้ใช้เอง
- ☐ และจัดทำเป็น “User Story” และนำมาพิจารณาแต่ละ story ว่าต้องใช้ระยะเวลาและต้นทุนเท่าใด
- ☐ User เลือก Story ที่ต้องการให้พัฒนามากที่สุด
- ☐ จัดแบ่งกลุ่มกิจกรรมต่างๆ ออกเป็นรอบตามจำนวนรอบที่เหมาะสม

Extreme Programming (XP)

■ การออกแบบ Design

- ☐ ยึดหลักการทำให้ง่ายที่สุดแม้ว่าระบบจะซับซ้อนมาก โดยจัดทำเป็นต้นแบบ
- ☐ สามารถออกแบบและกำหนดรายละเอียดเพิ่มฟังก์ชันที่คาดว่าผู้ใช้ต้องการไว้ให้ด้วย

Extreme Programming (XP)

■ การเขียนโปรแกรม Coding

- ทีมงานจะใช้วิธีจับคู่โปรแกรมเมอร์ 2 คนให้นั่งเขียนโปรแกรมในแต่ละ story ด้วยกัน เรียกว่า “Pair Programming” เพื่อการประกันคุณภาพในการเขียนโปรแกรม

Extreme Programming (XP)

■ การทดสอบโปรแกรม Testing

- จะทดสอบแบบ Unit testing โดยมีการสร้าง Unit Testing Case ไว้ก่อนการเขียนโปรแกรมภายใต้กรอบการสร้างงานทดสอบ จากนั้นจะนำไปให้ลูกค้าทดสอบ เรียกว่า Customer Test ซึ่งก็คือ การทดสอบยอมรับ (Acceptance Test)

Agile Process(อาไจล์)

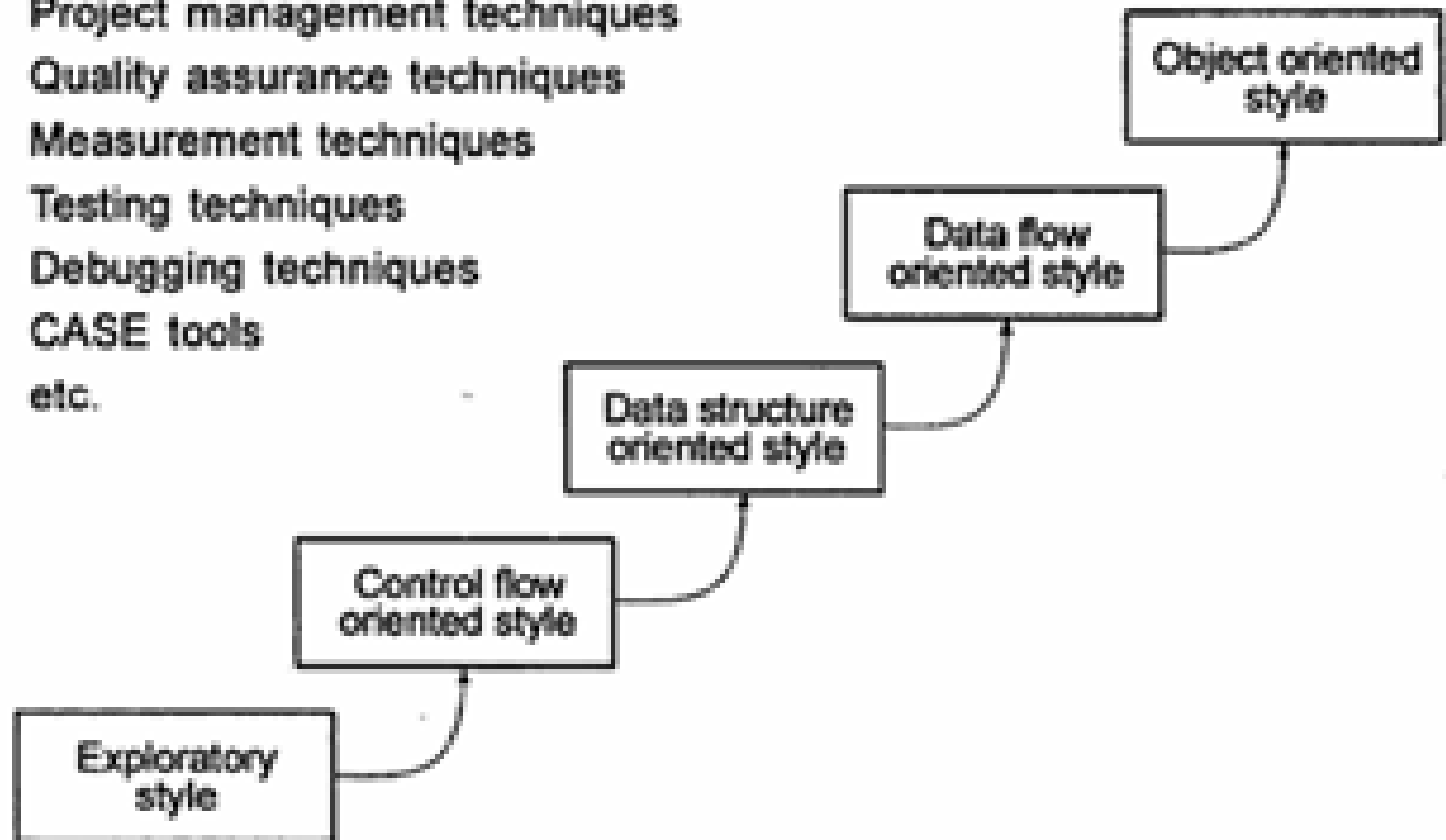
- ระเบียบวิธีปฏิบัติแบบ **Agile** แปลว่า คล่องแคล่ว ฉลาด นับไว ดังนั้นเป็นการคิดค้นวิธีการพัฒนาซอฟต์แวร์ใหม่ที่มีอิสระ และคล่องตัวสูงในการทำงาน

Agile Process มี 4 แนวทางปฏิบัติ ดังนี้

- ทีมงานทุกคนทำงานได้อย่างอิสระได้ สามารถติดต่อประสานงานด้วยกระบวนการและเครื่องมือแบบตัวต่อตัวไม่ต้องทำเอกสารเหมือนกับแบบจำลองอื่นๆ
- ทีมงานใช้เวลาส่วนใหญ่ในการเขียนโปรแกรมมากกว่าจัดทำเอกสาร
- ทีมงานมุ่งเน้นการทำงานร่วมกับลูกค้าโดยตรง มีการส่งมอบงานบ่อยครั้ง มีการเปลี่ยนแปลงเกิดขึ้นอยู่เสมอในระหว่างกระบวนการ
- ทีมงานให้ความสำคัญกับการแก้ไขงานทันทีที่มีการเปลี่ยนแปลง มากกว่าการวางแผนก่อนลงมือทำอื่นๆ

Evolution of the software development techniques

- Life cycle models
- Specification techniques
- Project management techniques
- Quality assurance techniques
- Measurement techniques
- Testing techniques
- Debugging techniques
- CASE tools
- etc.



เครื่องมือสนับสนุนการพัฒนาระบบ

CASE Tools เป็นเครื่องมือที่นำมาใช้
เพื่อสนับสนุนการพัฒนาระบบ ซึ่งจะช่วยให้
การพัฒนาระบบมีความรวดเร็ว
มีมาตรฐาน และคุณภาพดียิ่งขึ้น

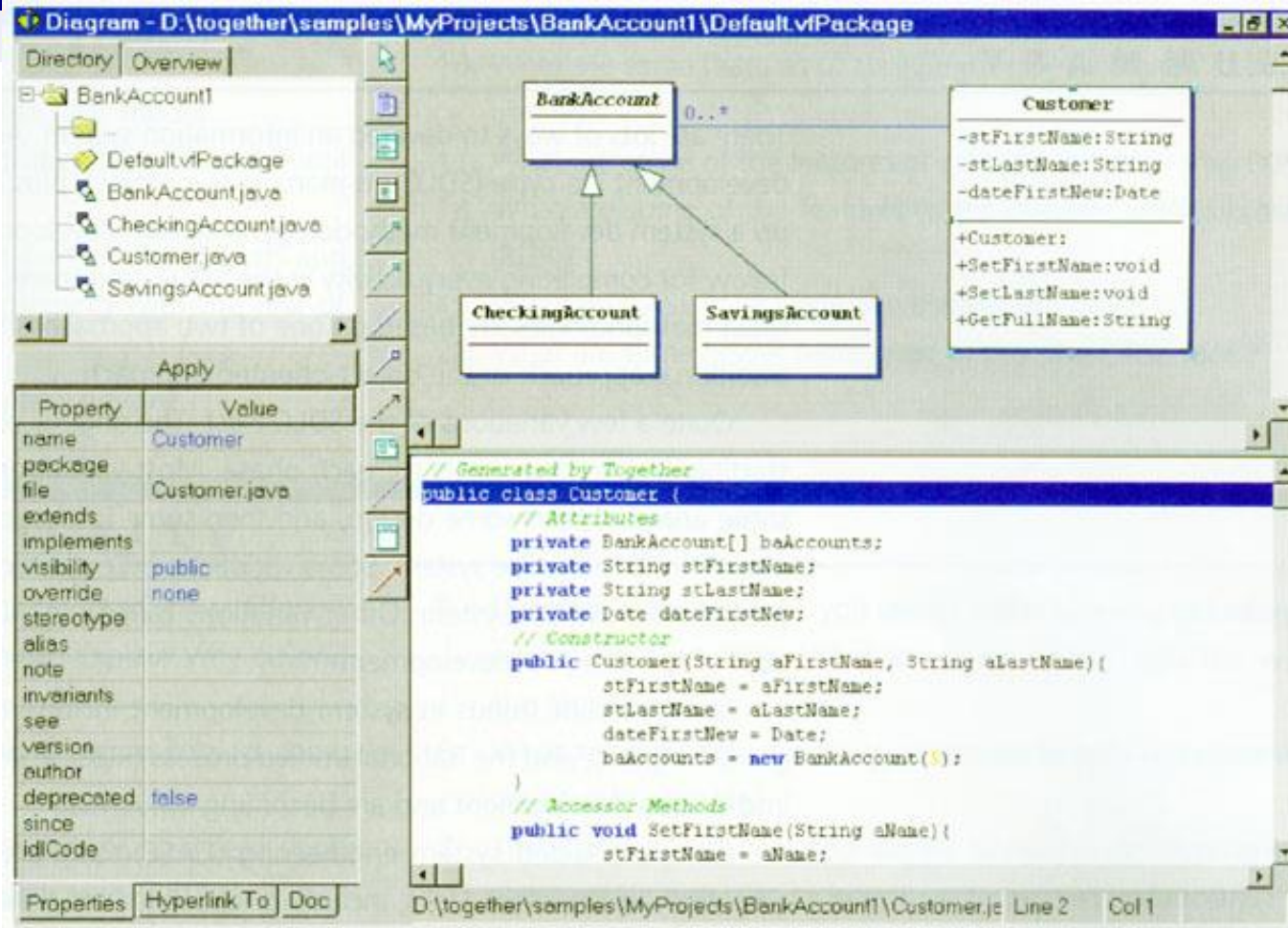
ประเภทของ CASE Tools

1. Upper CASE Tools

เป็นเครื่องมือสนับสนุนการวิเคราะห์และออกแบบจำลอง (Model)

2. Lower CASE Tools

เป็นเครื่องมือสนับสนุนการ Implement เช่น
การ Generate แบบจำลองที่สร้างขึ้นมาเป็นโค้ด



Lower CASE Tools

dfd1 - Microsoft Visio

File Edit View Insert Format Tools Shape Window Help

Type a question for help

Cordia New 12pt B I U

Normal Normal {No Layer}

Shapes

Search for Shapes:
Type your search

Entity Relationship
Data Flow Diagram Symbols
IDEF0 Diagram Shapes

Activity box Label
Title block Text block 8pt
Node Solid connector
1 legged connector IDEF0 connector

Basic Flowchart...

Process Decision
Document Data
Predefined process Stored data
Internal storage Sequential data
Direct data Manual input
Card Paper tape

Page 1

Page 1/1

start dfd1 - Microsoft Visio EN 20:10

Drawing Tools