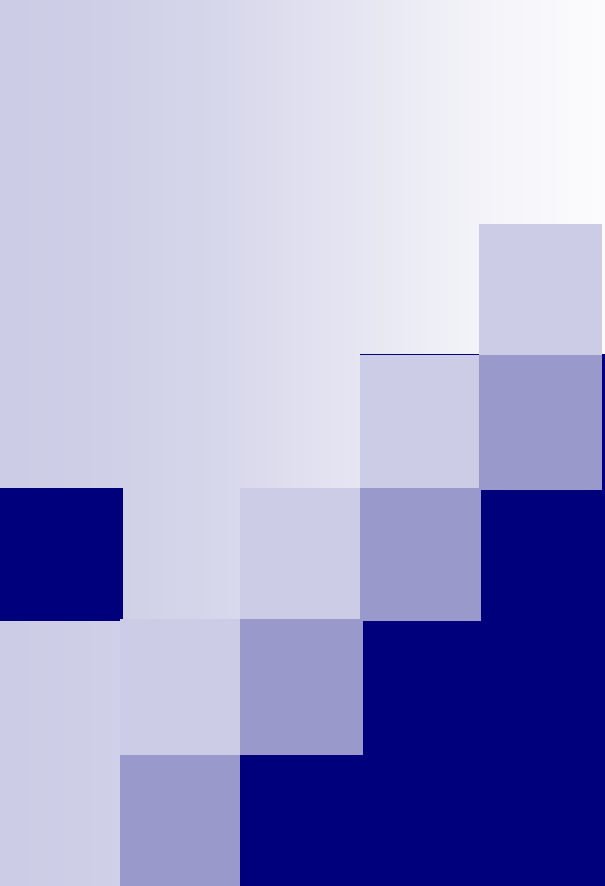
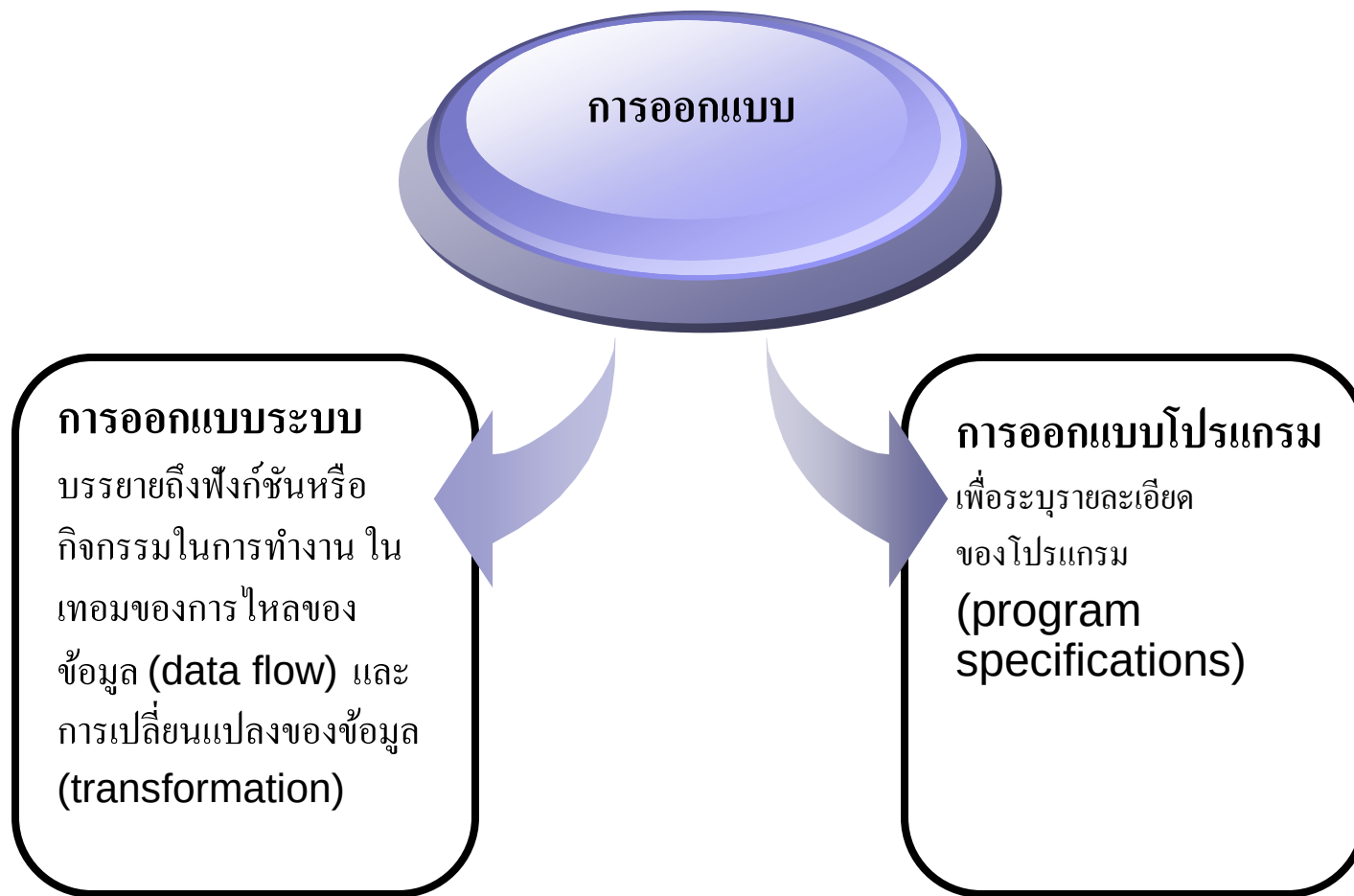




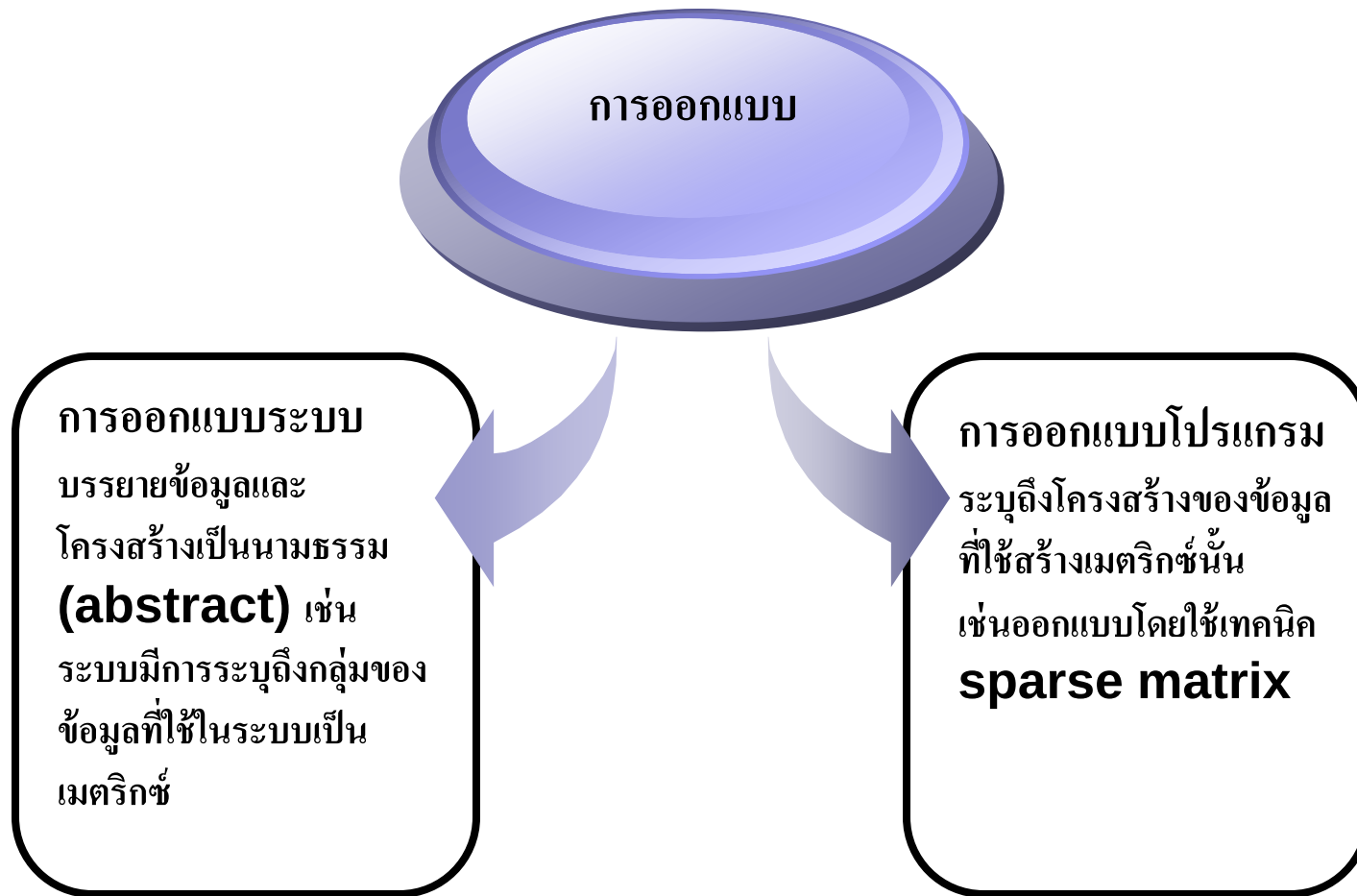
บทที่ 7

การออกแบบโปรแกรมและ
การพัฒนาโปรแกรม

การออกแบบ



ความแตกต่างระหว่างการออกแบบระบบกับการออกแบบโปรแกรม



การออกแบบโปรแกรมคืออะไร

- ระบุรายละเอียดของโปรแกรม(program specifications)
 - การกำหนดรายละเอียดของโมดูลต่างๆ รวมทั้งการเชื่อมโยงระหว่างโมดูล
 - อัลกอริทึมที่ใช้ในโมดูล ทำให้ทราบถึงรูปแบบของข้อมูลเข้า, รูปแบบข้อมูลที่เป็นผลลัพธ์ และการประมวลผลในโมดูล

โมดูลการออกแบบโปรแกรม

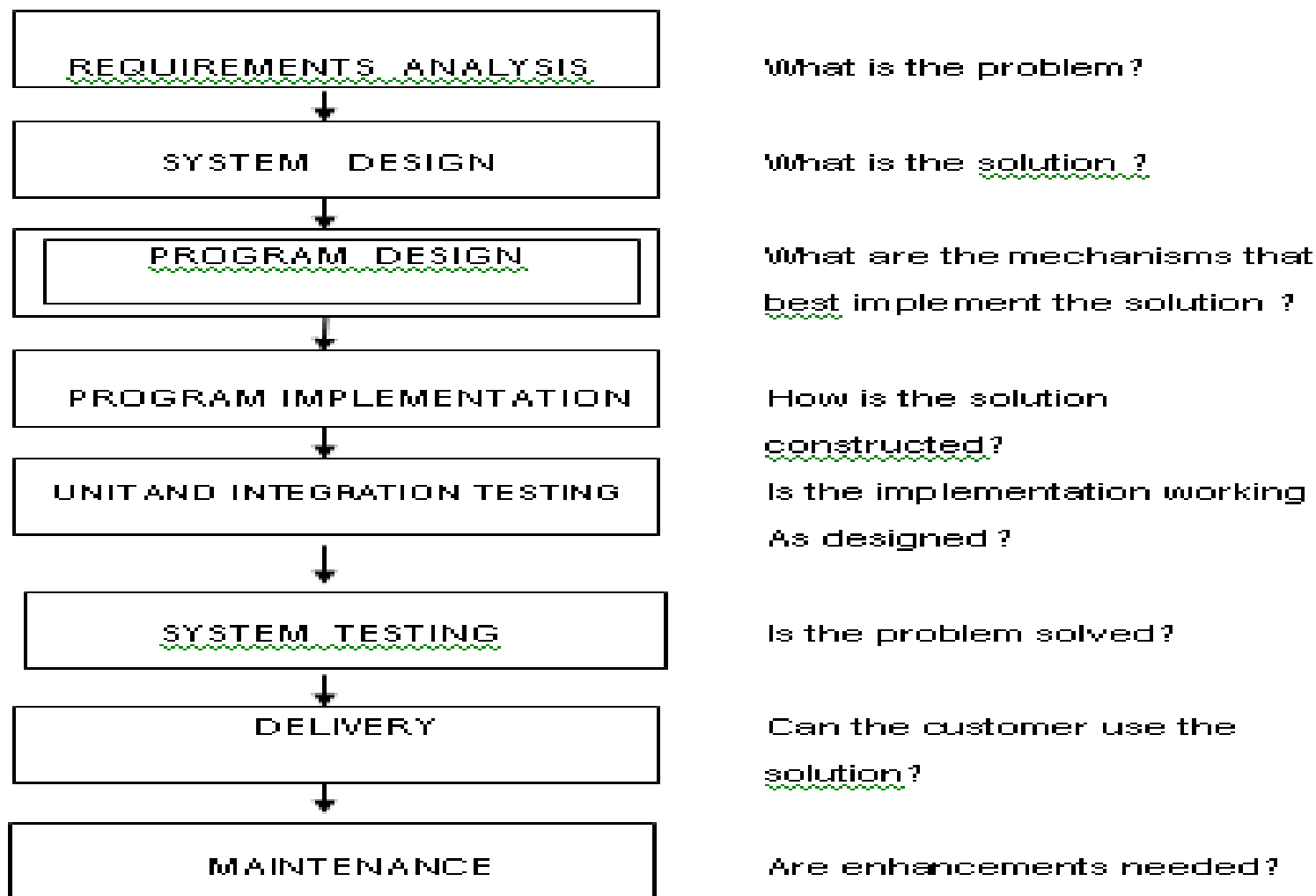
อัลกอริทึมอย่างละเอียด

การแทนค่าข้อมูลและโครงสร้างของข้อมูลต่างๆ

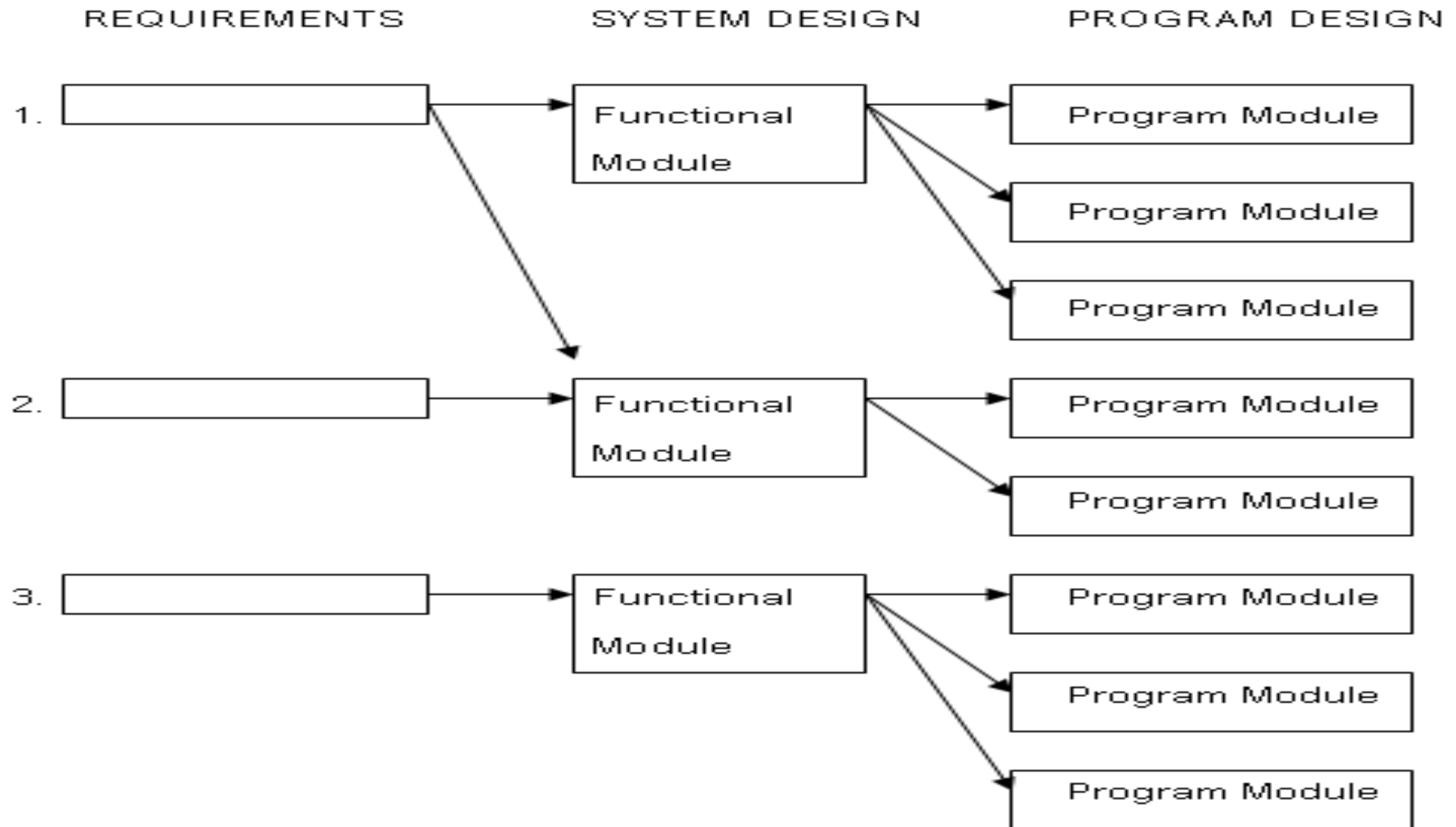
ความสัมพันธ์ระหว่างฟังก์ชันที่กระทำ
และข้อมูลต่างๆ ที่ใช้

รายละเอียดของ
การประมวลผลทางตรรก

รูปภาพที่ 5.1 แสดงขั้นตอนในการพัฒนาระบบ



Relationship among Requirements; System Design and Program Design



Top-down Approach

- การระบุรายละเอียดของโมดูลโปรแกรมจะกำหนดเป็นลำดับจากระดับบนก่อนและกำหนดในระดับต่ำลงมา
- กำหนดถึงข้อมูล, สารสนเทศของการควบคุม และโครงสร้างที่จำเป็นของโมดูล ซึ่งจะไม่เห็นถึงสารสนเทศในระดับล่าง



■ Modularity

- ทำการแยกการทำงานของระบบอย่างชัดเจน
- ทำให้สามารถเคลื่อนย้ายหรือลบหรือทดสอบและบำรุงรักษาระบบได้ง่ายขึ้น

Examination of Algorithms

- ทุกอัลกอริทึมต้องมีการตรวจสอบถึงความถูกต้องและประสิทธิภาพ
- ตรวจสอบถึงความเหมาะสมของอัลกอริทึมที่ใช้กับฮาร์ดแวร์หรือซอฟต์แวร์ที่ระบุไว้ในการออกแบบระบบว่าสามารถกระทำได้หรือไม่

Examination of the data type

- การออกแบบโปรแกรมนั้นผู้ออกแบบต้องออกแบบชนิดของข้อมูลให้ชัดเจน เพื่อที่ข้อมูลต่างๆในระบบจะมีความคงที่และมีความเชื่อถือได้สูง

Module Characteristics

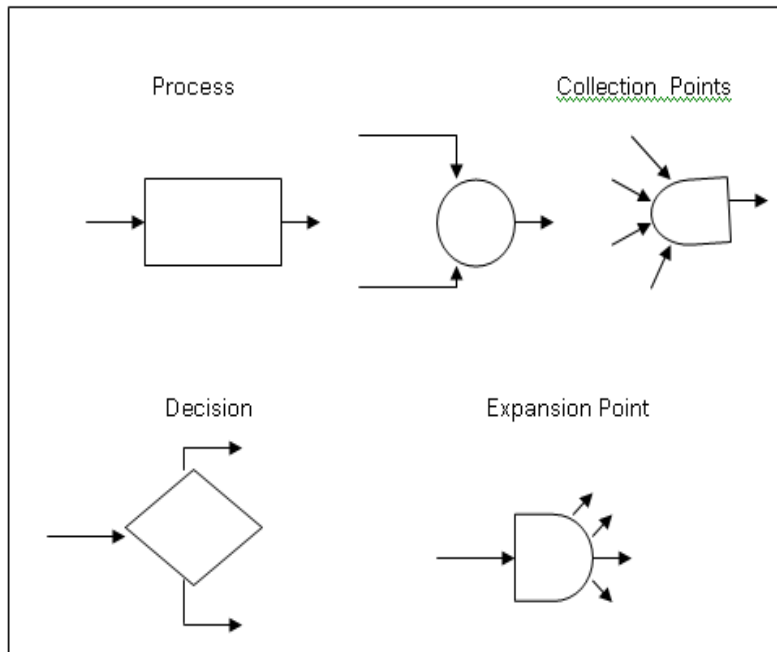
- โมดูลที่ออกแบบทุกโมดูลต้องมีทางเข้าและทางออก **1** ทางเท่านั้น
- เป็นประโยชน์ในการตรวจสอบการทำงาน, การติดตาม, การค้นหาถึงตำแหน่งที่ผิดพลาด รวมถึงแก้ไขเปลี่ยนแปลงได้ง่ายขึ้นด้วย

Packaging

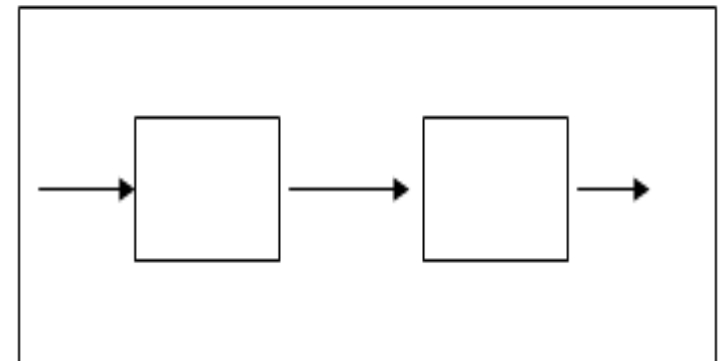
- กำหนดสภาพแวดล้อมของระบบที่สามารถทำงานได้ อาทิเช่น
กำหนดถึงฮาร์ดแวร์ที่ใช้ได้ในระบบ , ระบบควบคุมปฏิบัติการ
ที่ใช้ได้, ภาษาโปรแกรม , ข้อกำหนดของเวลาในการ
ประมวลผล , ขนาดของหน่วยความจำที่ต้องการ และอื่นๆ

เครื่องมือที่ใช้ในการออกแบบโปรแกรม

- **Flow Diagrams** บรรยายถึงตรรกของโปรแกรมโดยใช้แผนภาพทางเรขาคณิตผสมผสานกับลูกศรซึ่งกำกับการไหลของข้อมูลเพื่อแทนถึงการทำงานต่างๆในระบบ

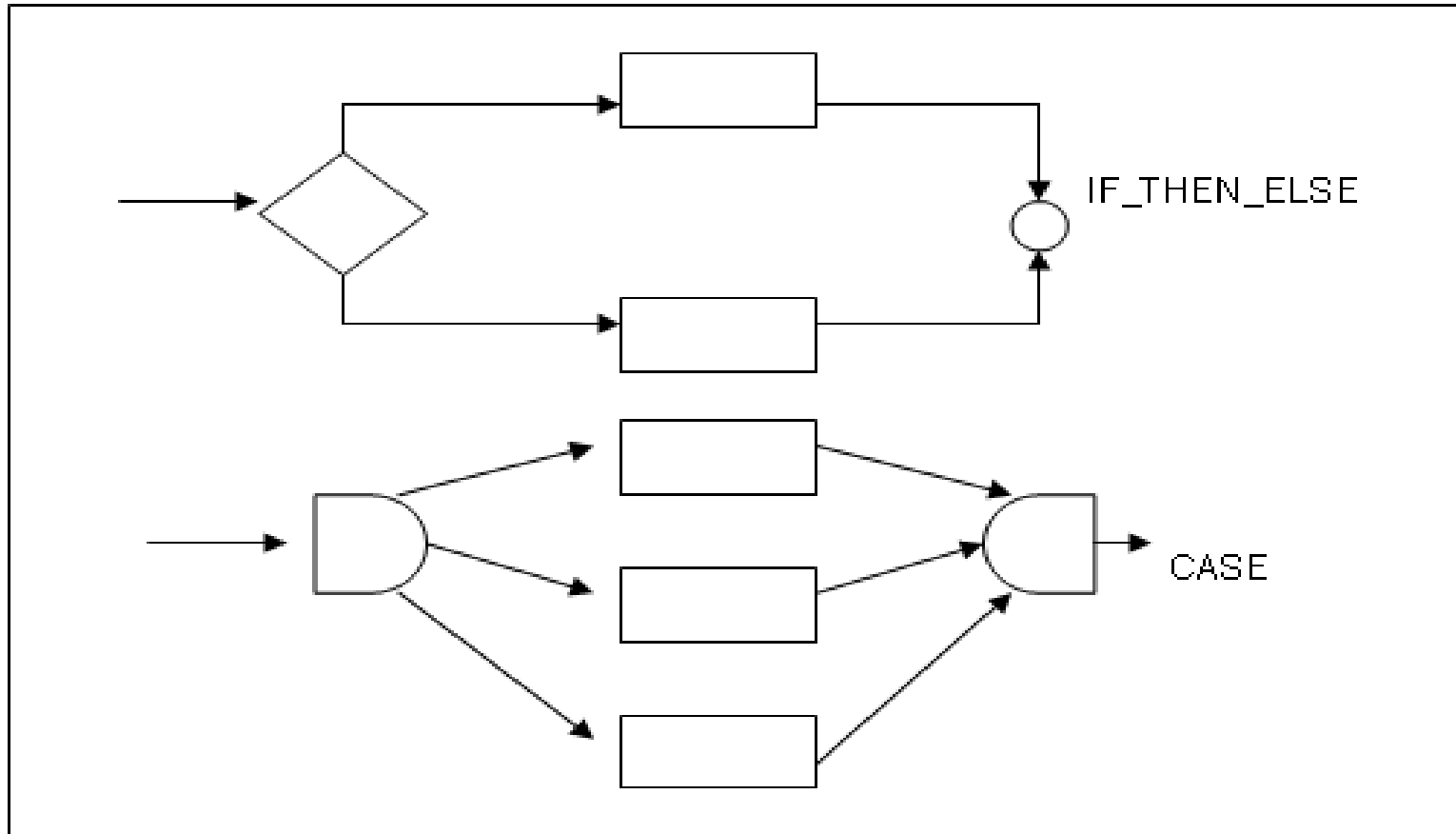


รูปภาพที่ 5.7 A Sequence Structure



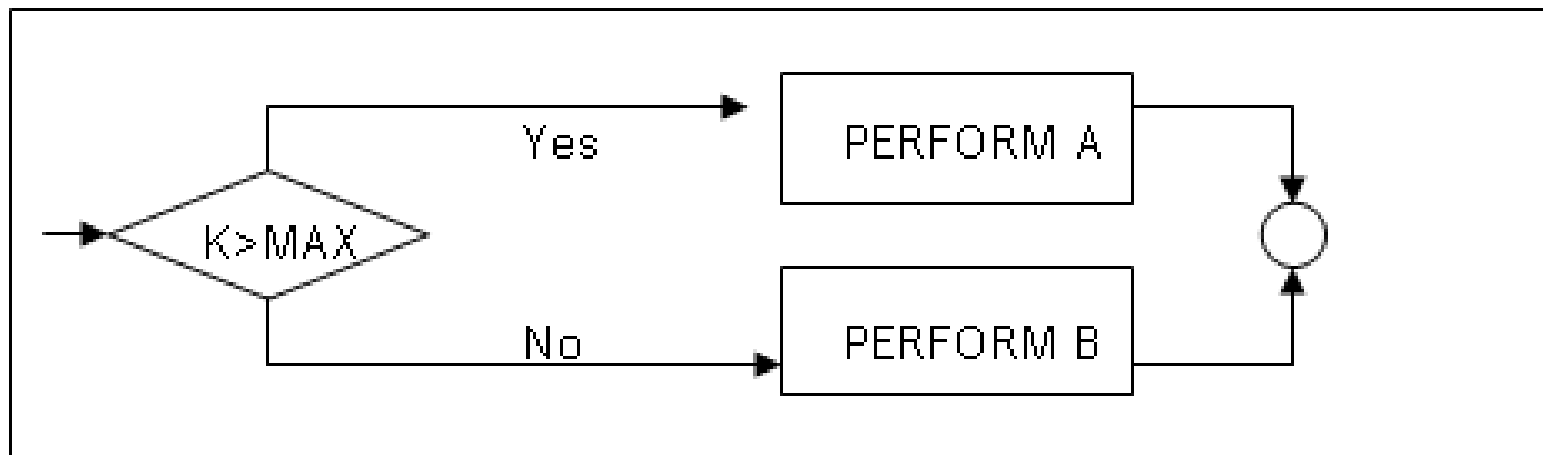
Flow Diagrams

รูปภาพที่ 5.8 Selection Structures



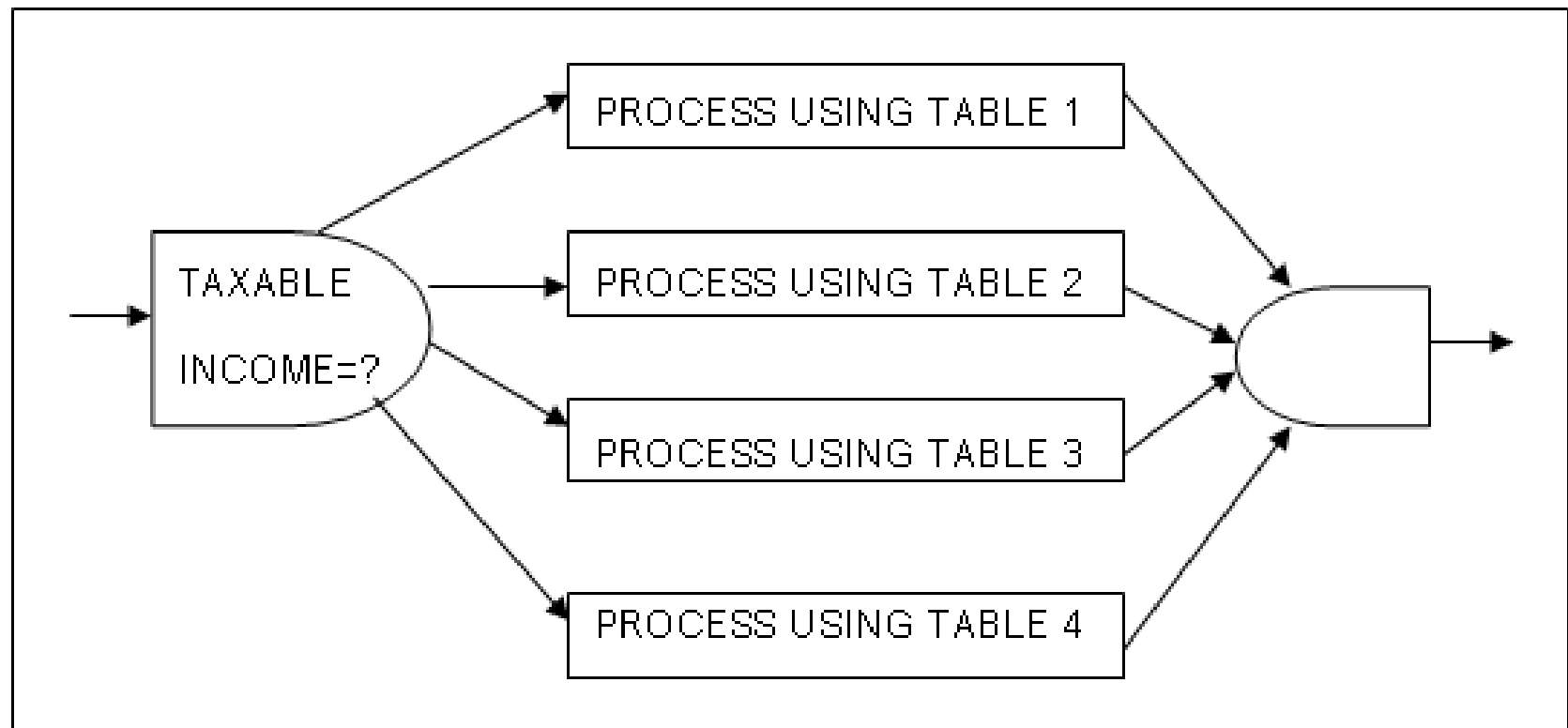
Flow Diagrams

รูปภาพที่ 5.9 IF-THEN-ELSE Selection



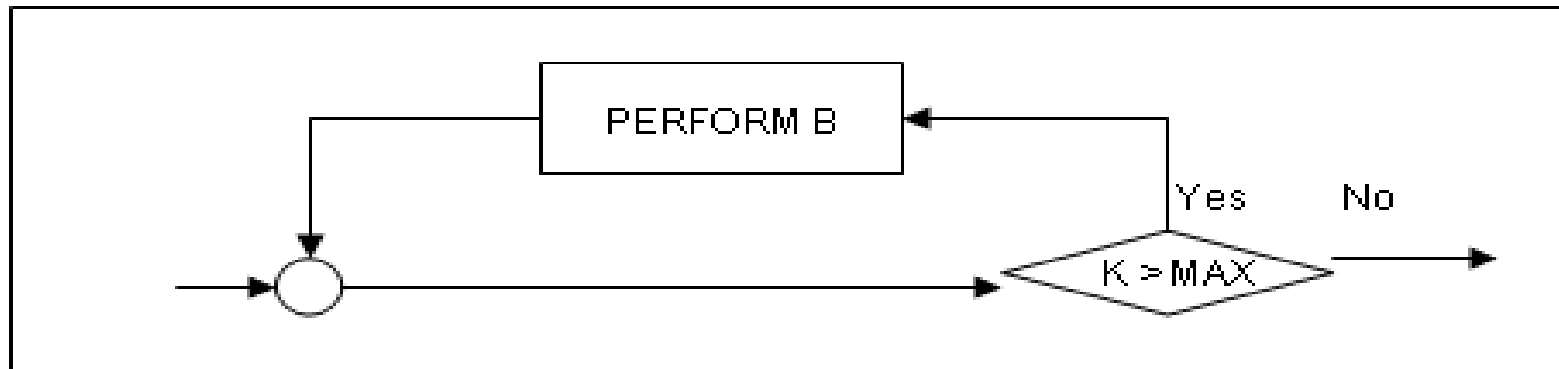
Flow Diagrams

รูปภาพที่ 5.10 CASE Selection

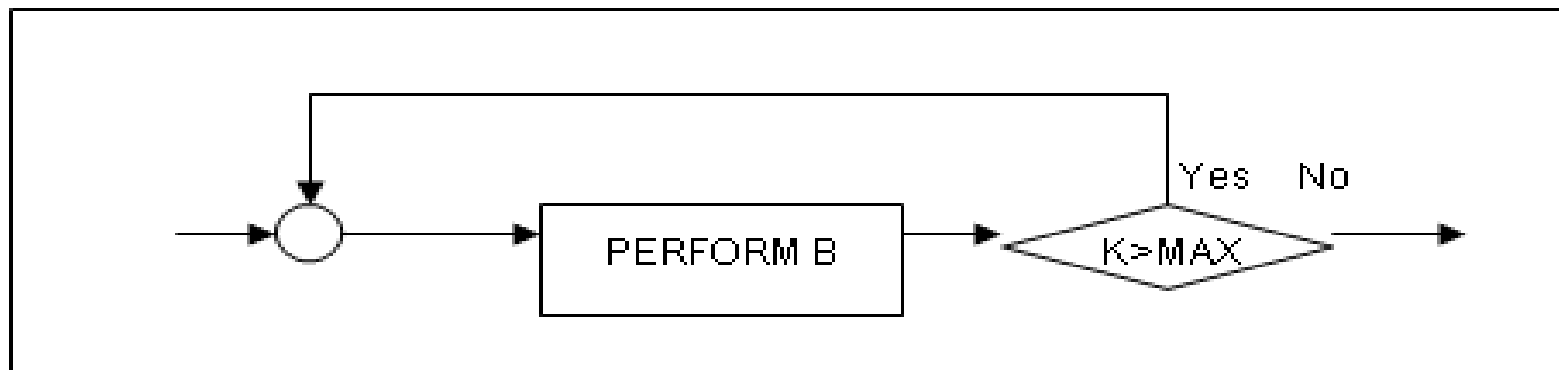


Flow Diagrams

รูปภาพที่ 5.12 DO-WHILE Structure



รูปภาพที่ 5.13 REPEAT-UNTIL Structure



Pseudo Code

- เป็นการออกแบบกระบวนการทำงานเป็นภาษาเขียน โดยไม่อิงถึงไวยากรณ์ หรือวากยสัมพันธ์ของภาษาโปรแกรมใดๆ
- การบรรยายจะแสดงถึงลำดับขั้นตอนในการทำงาน การเปรียบเทียบทางตรรก ทางเลือกของการทำงาน รวมทั้งการกระทำซ้ำต่างๆ
- ผู้ออกแบบสามารถปรับเปลี่ยนแก้ไขได้ง่ายและเป็นอิสระในการประยุกต์ใช้กับภาษาโปรแกรมต่างๆ

Pseudo Code

```
BEGIN FLOW_STRENGTH
```

```
    Get coordinates of measuring device
```

```
    Signal device to up – load file
```

```
    Record measurement
```

```
    Check measurement for within range
```

```
    Signal device to end transmission
```

```
END FLOW_STRENGTH
```

Pseudo Code

IF measurement in acceptable range

THEN update data tape

ELSE ask for retransmission

ENDIF

CASE of STREAM_HEIGHT

WHEN STREAM_HEIGHT < N1 call RESERVOIR_LOW

WHEN STREAM_HEIGHT = N1 call ADEQUATE_SUPPLY

WHEN STREAM_HEIGHT > N1 call FLOOD_ALERT

ENDCASE

Pseudo Code

```
WHILE hours worked this week > 0
```

```
    DO tax computations
```

```
ENDDO
```

```
    REPEAT data translation
```

```
        UNTIL end – of – file is reached
```

```
    ENDREPEAT
```

Automated Design Tools

GRAPHIC		
F Data Flow Diagram	Modify	Add
S Structure Chart		Delete
L Logical Data Model	Copy	Rename
R Ent. Rel Data model		List
D Struture Daigram		
P Presentation Graph		
Exit	NAME ASSMBLE AND DISTRIBUTE	

การเลือกเครื่องมือในการออกแบบโปรแกรม

- ขนาดของระบบ
- ชนิดของงานประยุกต์ที่เราพัฒนา
- มาตรฐานและกระบวนการของลูกค้า
- บุคลากรที่ทำงาน

คุณภาพของการออกแบบโปรแกรม

■ Structured Design

- ออกแบบให้เป็นโครงสร้างเพื่อให้ง่ายต่อการทำความเข้าใจ, ง่ายต่อการสร้างและง่ายต่อการบำรุงรักษา
- การออกแบบโครงสร้างที่ดีนั้นต้องมีคุณลักษณะดังนี้
 - ภายในโมดูลต้องมีความเกี่ยวข้องของฟังก์ชันต่างๆสูง รวมทั้งเกี่ยวข้องกับโมดูลอื่นๆในระดับต่ำ
 - แต่ละโมดูลควรมีทางเข้าและทางออกเพียง **1** ทางเท่านั้น

คุณภาพของการออกแบบโปรแกรม

■ Error Prevention and Correct

- การป้องกันและพยายามไม่ให้เกิดข้อผิดพลาดเท่าที่จะเป็นไปได้
- ถ้ามีข้อผิดพลาดก็ควรหาวิธีการในการแก้ไขเพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพ

Error detection

- เป็นการสืบหาข้อผิดพลาดในระบบ
- การสืบค้นความผิดพลาดได้ในขณะที่ระบบกำลังประมวลผลอยู่ เรียกว่า passive error detection และถ้าระบบสามารถตรวจสอบความผิดพลาดได้ด้วยตัวเองเราเรียกการสืบค้นนี้ว่า active error detection

Defensive Designing

- เป็นการออกแบบเพื่อป้องกันตัวเองจากความผิดพลาด เช่น
 - การตรวจสอบข้อมูลที่เข้าสู่ระบบหรือ โมดูลว่ามีค่าอยู่ในช่วงที่กำหนดหรือมีคุณลักษณะที่ถูกต้องหรือไม่
 - รวมทั้งความเป็นไปได้ของการคำนวณต่างๆ เช่นการหารด้วยค่าศูนย์ การหาค่ารากที่สองของเลขติดลบ เป็นต้น

Error Correction

- ระบบสามารถแก้ไขหรือตัดสินใจเพื่อทำการใดๆเพื่อแก้ปัญหาได้ เช่น
 - ☐ อาจให้หยุดกิจกรรมของระบบ
 - ☐ เก็บประวัติของผิดพลาดที่เกิดขึ้นแล้วย้อนกลับไปทำงานตามปกติโดยไม่มี ความเสียหายใดๆเกิดขึ้น

Error Tolerance

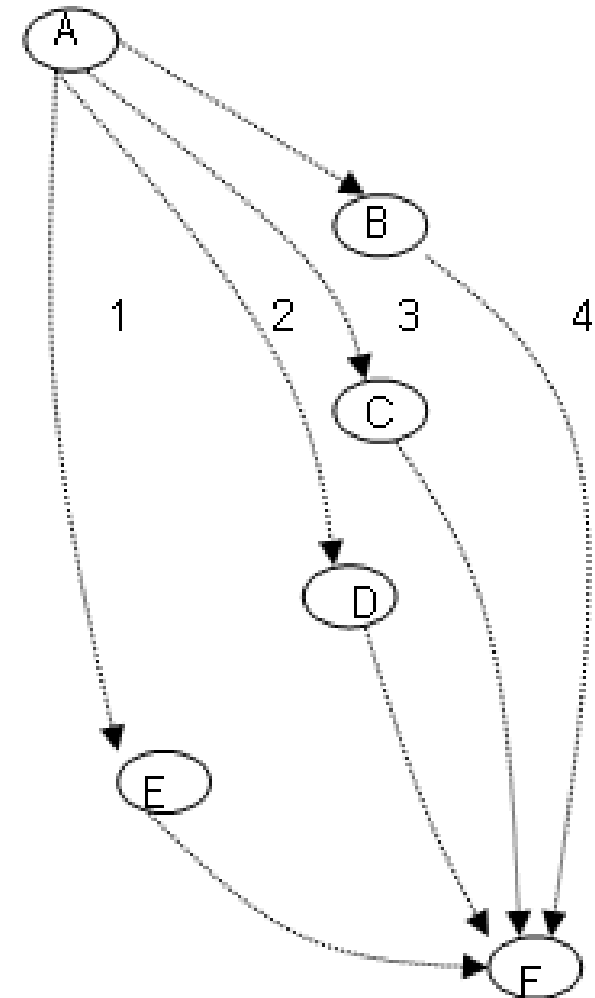
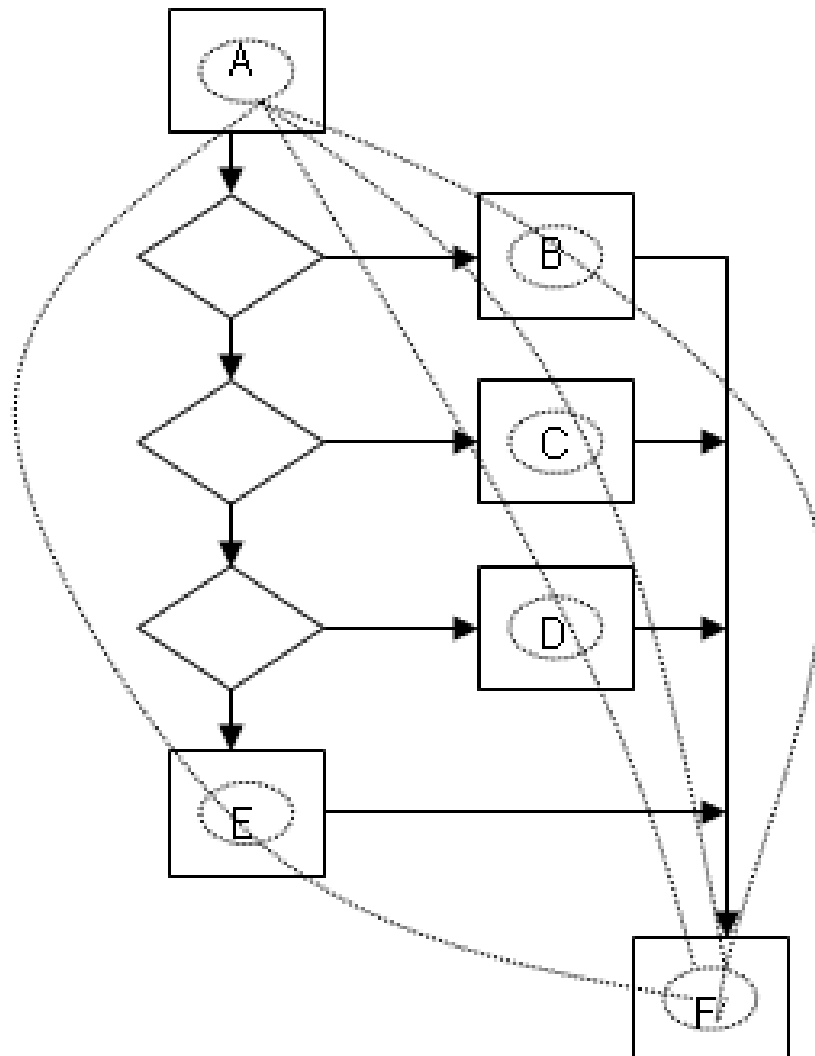
- เพิกเฉยกับความผิดพลาดที่เกิดขึ้น โดยแยกต้นเหตุของความเสียหายออกจากความผิดพลาดที่เกิดขึ้น โดยพยายามให้ส่งผลกระทบต่อระบบให้น้อยที่สุดเพื่อให้ระบบยังคงสามารถทำงานต่อไปได้

คุณภาพของการออกแบบโปรแกรม

- **Module Complexity**

- คุณลักษณะ **modularity** สามารถช่วยจัดการกับระบบที่มีความซับซ้อน โดยแยกระบบใหญ่ให้เป็นส่วนย่อยๆ เป็นการแยกปัญหาออกจากกัน

Flow Diagram for Example Module



คุณภาพของการออกแบบโปรแกรม

■ Unconditional Branching

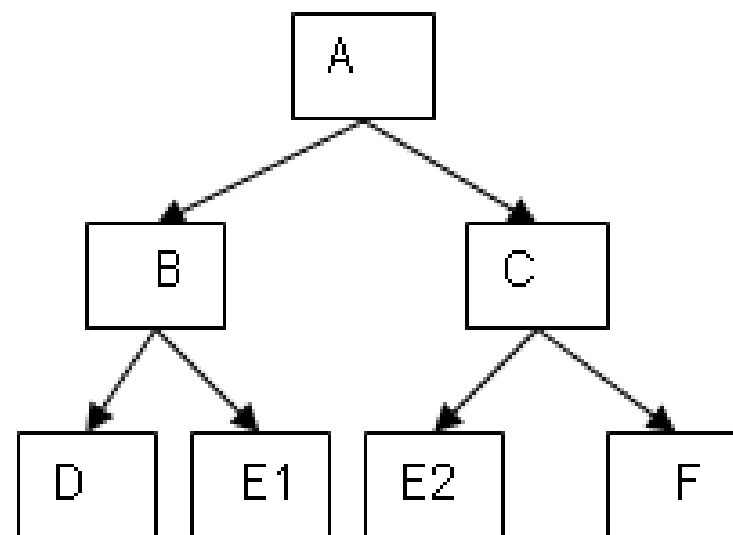
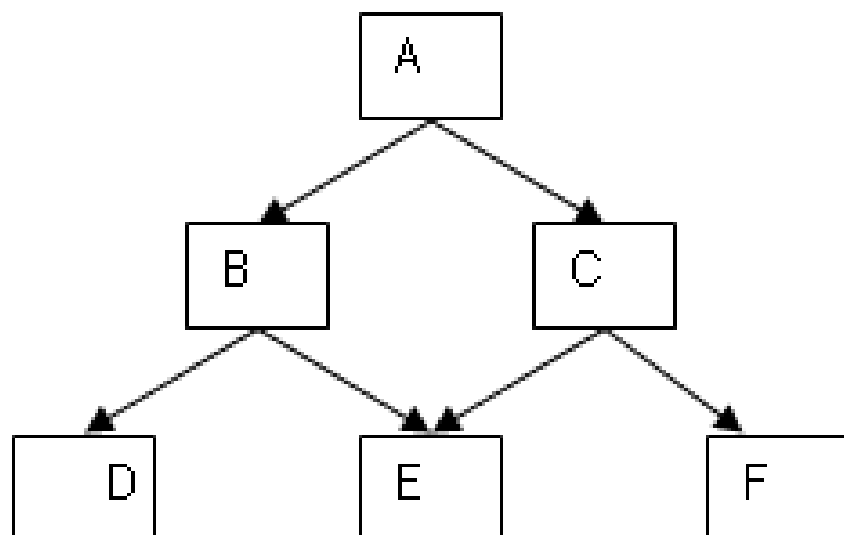
- ไม่จำเป็นต้องเขียนคำสั่งที่มีการกระโดดข้ามแบบไม่มีเงื่อนไข
 - ☐ การกระโดดข้ามไปมาทำให้โปรแกรมที่ออกแบบไม่เป็นโครงสร้างและยากต่อการทำความเข้าใจ หลีกเลี่ยงการใช้คำสั่ง **goto**

คุณภาพของการออกแบบโปรแกรม

■ Increasing the Structure

- ในกรณีเมื่อตรวจสอบแล้วมีทางเข้าหลายทางต้องมีการออกแบบโปรแกรมใหม่โดยทำการ คัดลอกการทำงานนั้นออกมา

คุณภาพของการออกแบบโปรแกรม



Testing the Modules for Structure

DESIGN REVIEWS

1

นักวิเคราะห์ระบบ,
นักออกแบบระบบ

2

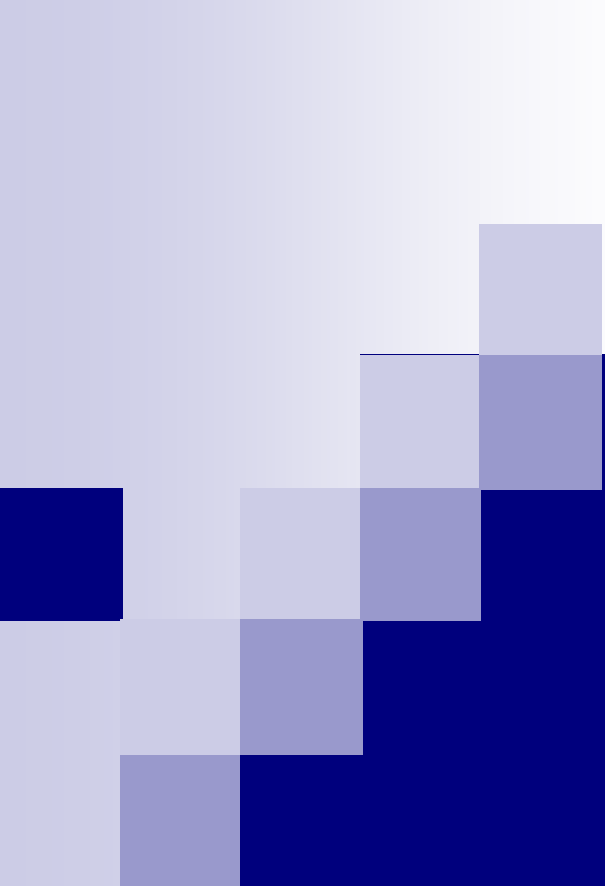
นักออกแบบโปรแกรม ,
ผู้ควบคุมการประชุม

3

เลขานุการจะทำหน้าที่จด
บันทึกและทำเอกสาร
ทางด้านเทคนิค

ในการตรวจสอบการออกแบบ โปรแกรม





การพัฒนาโปรแกรม



การจัดหา การติดตั้ง

- หลังจากการออกแบบทั้งระบบหรือบางส่วนจากระบบได้เสร็จเรียบร้อยแล้ว และผู้บริหารอนุมัติงบประมาณได้ดำเนินการต่อไป
- ที่ต้องดำเนินการ คือ การพิจารณาจัดหาฮาร์ดแวร์ และบริการที่เกี่ยวข้องกับส่วนของระบบซึ่งได้ออกแบบไปแล้ว

การจัดหาส่วนประกอบของระบบ

- การจัดหาส่วนประกอบของระบบจากผู้ขาย (Acquiring system components from vendors)
- ได้แก่
 - ☐ ฮาร์ดแวร์
 - ☐ ซอฟต์แวร์
- ซึ่งมีผู้ผลิตมากมายในปัจจุบัน และข้อมูลเกี่ยวกับผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ มีการเปลี่ยนแปลงที่ค่อนข้างเร็วมากเนื่องจากสภาพการแข่งขันในปัจจุบันมีสูงมาก

รายชื่อของบริษัทที่ประกอบธุรกิจด้านอุตสาหกรรมคอมพิวเตอร์

ลำดับที่จากโลก (International ranking)	ชื่อบริษัท (Company)	ลำดับที่จากโลก (International ranking)	ชื่อบริษัท (Company)
1	IBM	2	Fujitsu Ltd.
3	NEC	4	Digital Equipment Corp.
5	Hewlett-Packard Co.	6	Hitachi
7	AT&T	8	Unisys
9	Siemens/Nixdorf	10	Apple
11	Olivetti	12	Groupe Bull
13	Toshiba	14	Matsushita
15	Canon	16	EDS
17	Sun	18	ICL
19	Compaq	20	Xerox

แสดงรายชื่อผู้ประกอบการธุรกิจอุตสาหกรรมคอมพิวเตอร์ชั้นนำ 20 อันดับจากทั่วโลก

(The computer industry's global top 20 firms) (Parker and Case.1993:691)

รายชื่อบริษัทชั้นนำที่ผลิตซอฟต์แวร์มีอยู่ในส่วนการตลาด ซอฟต์แวร์ทั่วโลก

ลำดับที่ (Ranking)	ชื่อบริษัท (Company)	ลำดับที่ (Ranking)	ชื่อบริษัท (Company)
1	IBM	2	Fujitsu
3	Microsoft	4	Nec
5	Computer Associates	6	Oracle
7	Siemens/Nixdorf	8	Hitachi
9	Lotus Deveopment	10	Digital
11	Sema Group	12	Novell
13	Olivetti	14	ICL
15	Finsiel		

Some of the leading worldwide firms in the software segment of the computer industry)
(Parker and Case.1993:693



การขอให้ส่งข้อเสนอ

[Request For Proposal (RFP)]

- ในกรณีนี้เกิดขึ้นเมื่อผู้ซื้อมีความต้องการระบบ
- ผู้ซื้อจะทำจดหมายถึงผู้ขายเพื่อให้ส่งข้อเสนอมาให้ซึ่งเรียกว่า “การขอให้ส่งข้อเสนอ” (Request For Proposal (REP))
- รายละเอียดของเอกสารที่ส่งไปคือสิ่งที่บริษัทต้องการและขอร้องให้ผู้ขายที่สนใจทำข้อเสนอที่เป็นทางเลือกโดยที่รายละเอียดที่ผู้ขายเหล่านั้นมีความพอใจที่จะทำตามที่บริษัทต้องการอย่างไร การขอให้ส่งข้อมูลเสนอ (REP) อาจทำโดยมีความยาว 2-3 หน้า หรืออาจเป็นร้อย ๆ หน้าก็ได้ขึ้นอยู่กับขนาดความซับซ้อนของโครงการ

ตัวอย่าง ข้อมูลที่ส่งไปยังผู้ขายในคำร้องสำหรับใบแจ้งราคาและเงื่อนไขที่จะขายสินค้า

ข้อเสนอเกี่ยวกับส่วนตัวเครื่อง (Hardware requirements)

- ❖ 100 IBM PS/2 แต่ละหน่วยมีหน่วยความจำ 4 เมกกะไบต์ หน่วยขับเคลื่อนบันทึก 3 1/2 นิ้ว จานบันทึกแข็ง (Harddisk) 120 เมกกะไบต์ และแป้นพิมพ์
- ❖ 90 IBM Proprinter พร้อมสายขนาดใหญ่
- ❖ 3 IBM เครื่องพิมพ์เลเซอร์พร้อมตัวประสานที่จำเป็น
- ❖ 100 SVGA จอสี

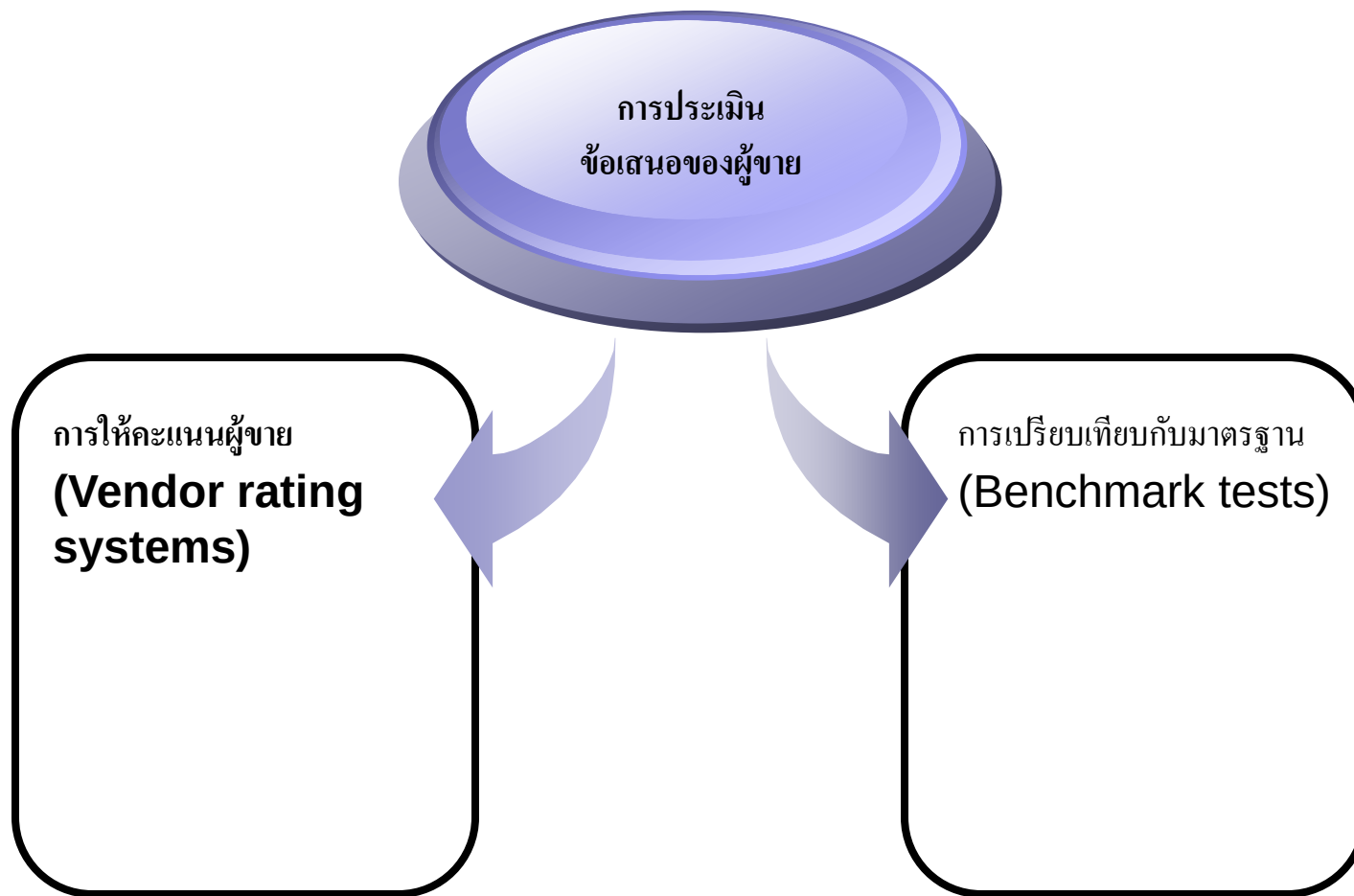
ข้อเสนอเกี่ยวกับซอฟต์แวร์ (Software requirements)

- ❖ MS-DOS, OS/2 และ Microsoft Windows พร้อมด้วยใบอนุญาตสำหรับ 100 ระบบ (รุ่นล่าสุดเท่าที่จะหาได้)

รายละเอียดอื่น ๆ ที่ควรต้องรวมอยู่ในใบสั่ง (Other details to be included in the bid)

- ❖ ราคาซื้อสำหรับแต่ละรายการถ้าซื้อแยกเดี่ยว ๆ หรือปริมาณเฉพาะ
- ❖ ระยะเวลาที่รับประกัน
- ❖ สิ่งอื่น ๆ ที่เสนอเพิ่มมาให้
- ❖ ค่าธรรมเนียมด้านการส่งของและติดตั้ง
- ❖ ระยะเวลาการจัดส่ง

การประเมินข้อเสนอของผู้ขาย (Evaluating vendor proposals)



การให้คะแนนผู้ขาย (Vendor rating systems)

- เป็นเครื่องมือที่ใช้กันอย่างแพร่หลายในการประเมินผู้ขาย

คือระบบซึ่งผู้ขายจะถูกให้คะแนนตามเกณฑ์ที่กำหนด

ปัจจัยในการเลือก (Selection criterion)	ค่าน้ำหนัก (1-10) (Weight)	คะแนนบริษัทผู้ขาย A. (Vendor score A.)	คะแนนบริษัทผู้ขาย B. (Vendor score B.)
ความง่ายในการใช้งาน (Ease of use)	10	$5 \times 10 = 50$	$10 \times 10 = 50$
ความเร็วของการประมวลผล (Speed)	7	$8 \times 7 = 56$	$6 \times 7 = 42$
ขีดความสามารถ (Capacities)	7	$8 \times 7 = 56$	$5 \times 7 = 35$
ราคา (Cost)	4	$4 \times 4 = 16$	$7 \times 4 = 28$
การบริการ (Support)	5	$3 \times 5 = 15$	$7 \times 5 = 35$
เอกสารประกอบ (Documentation)	5	$5 \times 5 = 25$	$7 \times 5 = 35$
จำนวนงานที่ทำได้ (Functionality)	10	$7 \times 10 = 70$	$8 \times 10 = 80$
คะแนนรวม (Total score)		288	355

การเปรียบเทียบกับมาตรฐาน (Benchmark tests)

- การทดสอบเพื่อวัดความสามารถในการประมวลหรือการทำงานของโปรแกรมใดโปรแกรมหนึ่ง หรืออุปกรณ์คอมพิวเตอร์ชนิดใดชนิดหนึ่ง โดยเทียบเคียงกับเกณฑ์มาตรฐาน



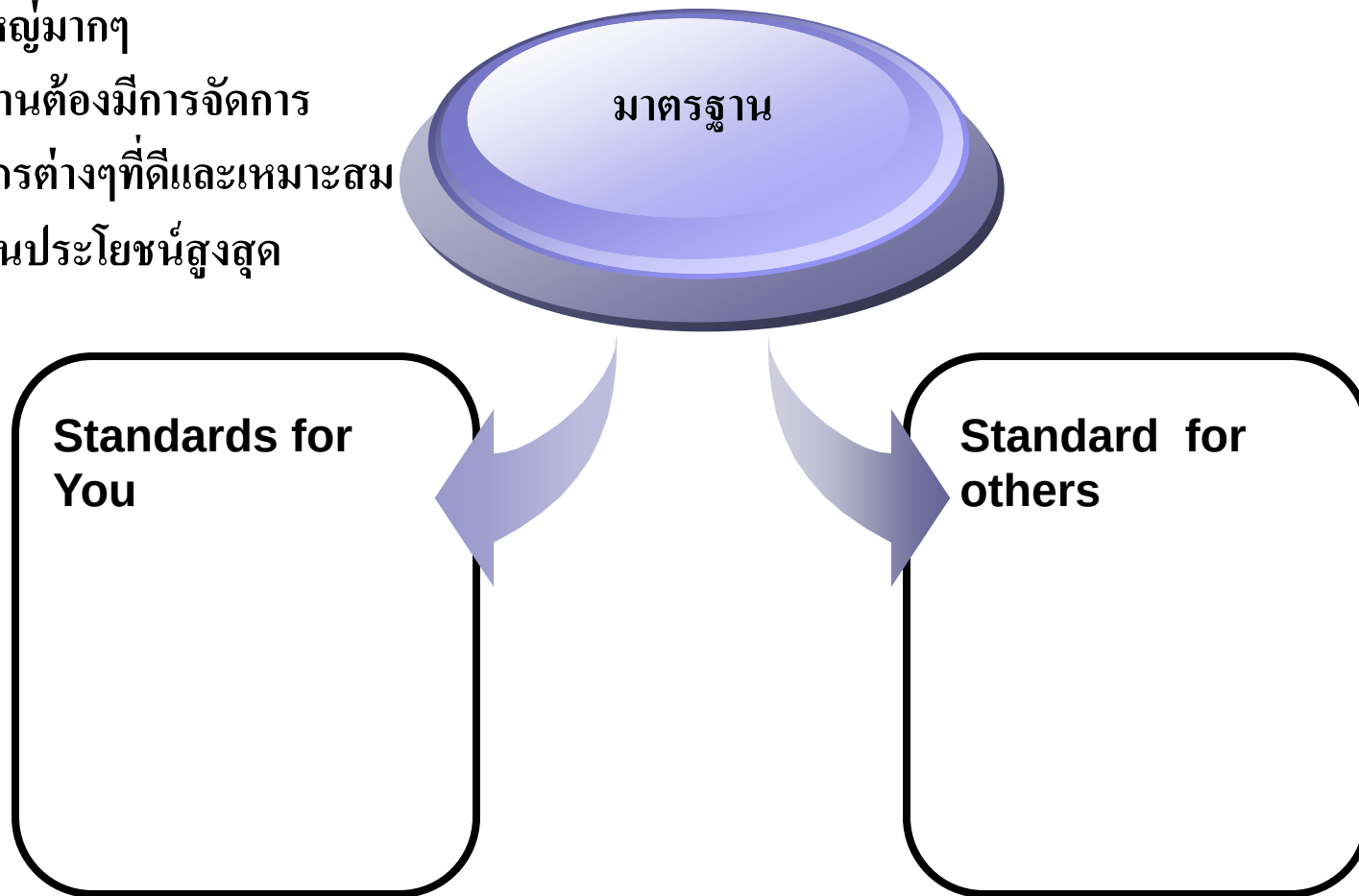
การทำสัญญากับผู้ซื้อ

(Contracting with vendors)

- หลังจากเลือกผู้ขายได้
- การทำสัญญาสัญญาคือเอกสารสำคัญตามกฎหมาย ซึ่งแสดงรายการว่ามีอะไรบ้างที่เป็นข้อตกลงขั้นต้น คณะผู้ทำการตกลงคือใคร สินค้าอะไร บริการอะไร และสกุลเงินตราที่แลกเปลี่ยน การกระทำต่อเนื่องที่คาดหวังว่าจะได้รับคืออะไร และถ้าไม่เป็นไปตามที่คาดหวังโดยฝ่ายใดฝ่ายหนึ่งจะต้องทำอย่างไร สัญญาที่ดีจะต้องทำในแบบที่ชัดเจนต่อทุกฝ่าย

มาตรฐานและกระบวนการในการเขียนโปรแกรม

การทำงานเป็นทีมงานที่บางครั้งมี
ขนาดใหญ่หลายๆ
การทำงานต้องมีการจัดการ
ทรัพยากรต่างๆที่ดีและเหมาะสม
จึงจะเป็นประโยชน์สูงสุด



Standards for You

- มาตรฐานและขั้นตอนวิธีในการเขียนโปรแกรมสามารถช่วยจัดการความคิด และหลีกเลี่ยงข้อผิดพลาด
 - ☐ เช่นการกำหนดถึงระเบียบและวิธีการในการเขียนเอกสารโปรแกรม
 - ☐ ผลดีคือเอกสารมีความชัดเจนและช่วยให้สามารถเข้าใจและตรวจสอบคำสั่งโปรแกรมเพื่อค้นหาตำแหน่งของความผิดพลาดและแก้ไขคำสั่งโปรแกรมนั้นได้โดยง่าย
- มาตรฐานและกระบวนการในการเขียนโปรแกรมสามารถช่วยในการเปลี่ยนจากการออกแบบไปเป็นคำสั่งโปรแกรมได้ง่ายขึ้น
 - ☐ โปรแกรมเมอร์สามารถกระทำได้โดยง่ายและลดความผิดพลาดที่อาจเกิดขึ้นได้

ระเบียบวิธีการในการเขียนคำสั่งโปรแกรม

- การกำหนดให้ทุก ๆ โมดูลที่พัฒนาต้องเริ่มต้นจากส่วนของ**การอธิบาย**หน้าที่การทำงานของโมดูลและการติดต่อสื่อสารกับโมดูลอื่น ๆ

ตัวอย่าง

- * **MODULE TO FIND INTERSECTION OF TWO LINES**
- * **MODULE NAME : FINDPT**
- * **PROGRAMMER : K. MERTZ**
- * **VERSION : 1.0 (12 FEB 87)**
- * **PROCEDURE INVOCATION :**
 - * **CALL FINDPT (A1, B1, C1, A2, B2, C2, XS, YS, FLAG)**
- * **INPUT PARAMETERS :**
- * **INPUT LINES ARE OF THE FORM**
 - * **$A1 \cdot X + B1 \cdot Y + C1 = 0$ AND**
 - * **$A2 \cdot X + B2 \cdot Y + C2 = 0$**
 - * **SO INPUT IS COEFFICIENTS A1, B1, C1 AND**
 - * **A2, B2, C2**
- * **OUTPUT PARAMETERS :**
 - * **IF LINES ARE PARALLEL, FLAG SET TO 1**
 - * **ELSE FLAG = 0 AND POINT OF INTERSECTION**
 - * **IS (XS, YS)**
- * **PROCEDURE CALLED BY :**
 - * **DRAW1**
 - * **FINDNEXT**
- * **PROCEDURE CALLS :**
 - * **ERRMSG**

Control Structure

Using Fundamental Constructs

Top-Down Flow

Use Of Submodules

เขียนคำสั่งโปรแกรม
สอดคล้องกับการออกแบบ
โปรแกรมไปใน
แนวทางเดียวกัน

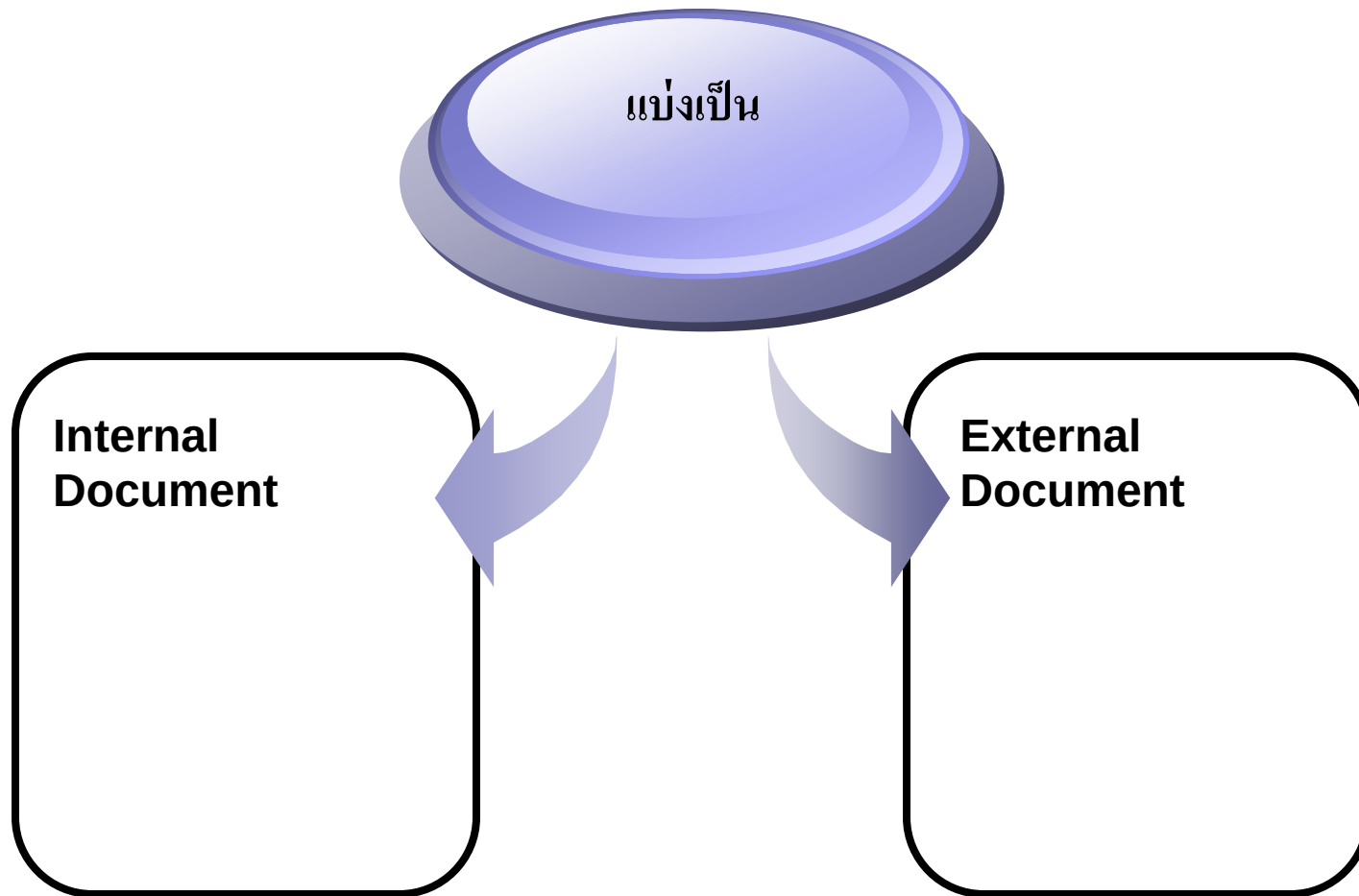
Algorithms

- พิจารณาถึงอัลกอริทึมที่เหมาะสมในการแก้ปัญหาอาจมีการนำอัลกอริทึมพิเศษมาใช้ในการเขียน โมดูล
- **Efficiency** ปรับปรุงประสิทธิภาพให้มีความรวดเร็วขึ้น

Data Structures

- การออกแบบรูปแบบของข้อมูลที่ดีจึงเป็นสิ่งสำคัญในการเขียนโปรแกรมคำสั่ง เพราะถ้ามีการจัดการ โครงสร้างของข้อมูลต่างๆ ให้อยู่ใน รูปแบบที่เหมาะสมจะช่วยให้การจัดการข้อมูลและการใช้งานได้ง่ายขึ้น

Documentation



Internal Document

■ เป็นเอกสารซึ่งเขียนภายในโมดูลโปรแกรม

เป็นรายละเอียดเพื่ออธิบายข้อต่อโค้ดของโมดูลโปรแกรม

- ☐ การนิยามโมดูลโปรแกรม
- ☐ คำอธิบายโครงสร้างของข้อมูล
- ☐ อัลกอริทึม
- ☐ การควบคุมการไหล(Control Flow)

Header Comment Block

PROGRAM SCAN – Program to scan a line of text for a given character

PROGRAMMER: Beatrice Clarman (718)345-6789

CALLING SEQUENCE: CALL SCAN(length, char)

Where 'LENGTH' is the length of the line to be scanned, "CHAR" is the character to be sought, line of text passed as array 'NTEXT'

VERSION 1: WRITTEN 1-12-86

REVISION 1.1: 2-3-86 improve searching algorithm.

PURPOSE: General- purpose scanning module to be used for each new line of text, no matter the length.

DATA STRUCTURES : Variable LENGTH – INTEGER

Variable CHAR – CHARACTER

ARRAY NTEXT – CHARACTER array of length
'LENGTH'

ALGORITHM : Reads array NTEXT one character at a time ; if CHAR is found, position in NTEXT returned in variable 'LENGTH' ; else variable 'LENGTH' set to 0

Internal Document

■ Other Program Comments

- คำอธิบายการทำงานของคำสั่งในโปรแกรมเป็นสิ่งที่จะทำให้เราเข้าใจโครงสร้างการทำงานของโปรแกรมโดยชัดเจน
 - ☐ แยกโปรแกรมเป็นระยะๆ เพื่อแสดงถึงกิจกรรมหลักๆ
 - ☐ ภายในแต่ละกิจกรรมจะมีการอธิบายออกเป็นขั้นตอนต่างๆ ตามหน้าที่ต่างๆที่เกี่ยวข้อง

Meaningful variable names and statement labels

- เป็นการเลือกชื่อตัวแปร และคำสั่ง ให้สื่อความหมายง่าย

ต่อการอ่านและเข้าใจ

weekwage = (Hrrate * Hour) + (1.5) * (Hrrate) * (Hours - 40)

Formatting to Enhance Understanding

- การจัดย่อหน้าและเว้นวรรคประโยคคำสั่งที่เหมาะสม สามารถตอบสนองกับโครงสร้างควบคุมพื้นฐานได้

```
if      xcoord < ycoord      then result = -1;
else    if xcoord = ycoord    then
        if slope1 > slope2    then result = 0;
                                else result = 1;
else    if slope1 > slope2    then result = 2;
else    if slope1 < slope2    then result = 3;
else                                         result = 4;
```

External Documentation

- สำหรับบุคลากรอื่นๆที่ไม่มีหน้าที่เกี่ยวข้องกับการเขียนคำสั่ง
เช่น นักออกแบบโปรแกรม
- เอกสารนี้มีประโยชน์ในการปรับปรุงหรือขยายขีดความสามารถของระบบเนื่องจากเอกสารนี้จะบ่งบอกถึงรายละเอียดต่างๆที่เกี่ยวข้องกันของโปรแกรมอย่างละเอียด

External Documentation

Describing the Problem

อธิบายถึง**ปัญหา**ต่างๆ และ **ตำแหน่งที่อยู่**ของปัญหาในโปรแกรม รวมทั้งอธิบายถึง**ทางเลือกในการแก้ปัญหา**และ**ข้อสรุป**ว่าทำไมถึงเลือกวิธีนี้ในการแก้ปัญหา

Describing algorithms

เป็นการบรรยายถึง**อัลกอริทึม**ที่ใช้ในโปรแกรมของแต่ละโมดูล สูตรที่ใช้ ขอบเขต ขั้นตอนวิธีการในการปฏิบัติงาน เงื่อนไขพิเศษต่างๆ รวมถึงการได้มาโดยอ้างอิงจาก textbook หรือ journal article หรือการออกแบบโปรแกรม

Describing the Data

สำหรับการ**บรรยายข้อมูล**ของเอกสารภายนอกนี้ ผู้ใช้หรือโปรแกรมเมอร์สามารถมองเห็นทิศทางการไหลของข้อมูลผ่านระบบเป็นชั้นของโมดูลโปรแกรม



สรุป

DOCUMENTATION:

Internal document:

- Header comment block

- Indentation to exhibit control structure

- Meaningful variable and data names

- Definition of interface requirements

- Description of program invocation

สรุป

PROCESSING:

- Error handling:

 - Brief error messages

CONTROL FLOW:

- Use structured techniques

- Keep coupling low

- Keep cohesion high

LANGUAGE:

- Avoid unconditional branching where possible

- Use language-specific structures to maximize productivity