



Software Development Methodology

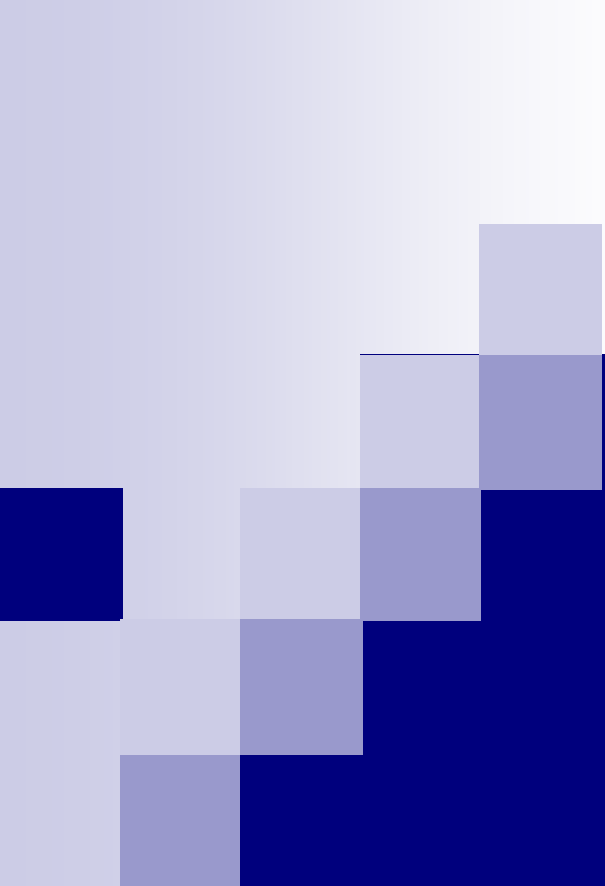
โดย ผศ.ดร.อุไร ทองหัวไผ่

เนื้อหา

- วิศวกรรมซอฟต์แวร์
- การวางแผนโครงการ
- การวิเคราะห์ความต้องการ
- การออกแบบระบบ
- การออกแบบโปรแกรม

เนื้อหา

- การพัฒนาโปรแกรม
- การทดสอบโปรแกรม
- การทดสอบระบบ
- การส่งมอบระบบ
- การบำรุงรักษาระบบ
- ความผิดพลาดในการพัฒนาระบบ



ครั้งที่ 1

วิศวกรรมซอฟต์แวร์คืออะไร?

วิศวกรรมซอฟต์แวร์คืออะไร

- การพัฒนาระบบงานที่ประสบผลสำเร็จ โดยใช้ความรู้ทางด้านคอมพิวเตอร์และการประมวลผล ซึ่งต้องเข้าใจถึงธรรมชาติของปัญหา และเลือกใช้เทคโนโลยี เช่น เครื่องมือ (tools) ที่เหมาะสม รวมทั้งเทคนิคต่างๆ เพื่อช่วยในการแก้ปัญหา

Software Engineering Definition

วิศวกรรมซอฟต์แวร์ คือกระบวนการสร้างสรรค์
โปรแกรม โดยใช้หลักทางวิศวกรรมเข้ามาช่วย
ในการดำเนินการสร้าง

อ.สมหมาย สุขคำ

Software Engineering Definition

“Software Engineering is systematic approach to the development operation , maintenance , retirement of software”

IEEE 83

“การนำเอาหลักการทำงานวิศวกรรมมาใช้ใน
การพัฒนาซอฟต์แวร์ให้เป็นไปอย่างมีระบบ”

Software Engineering Definition

การพัฒนาซอฟต์แวร์ให้ได้ผลลัพธ์ใกล้เคียงเป้าหมาย หรือ
บรรลุเป้าหมายของการพัฒนาอันได้แก่

@ ซอฟต์แวร์ที่มีคุณภาพ

@ สามารถส่งมอบได้ตรงเวลา

@ อยู่ภายใต้งบประมาณที่คาดการณ์

@ มีคุณสมบัติตรงตามความต้องการของผู้ใช้

Software Engineering Definition

“วิชาการว่าด้วยการออกแบบโปรแกรมคอมพิวเตอร์
ตลอดจนการบริหารงานการพัฒนาเพื่อที่จะได้มาซึ่ง
ผลิตภัณฑ์ซอฟต์แวร์ที่มีคุณภาพสูง ราคาถูก และภายใน
เวลาที่กำหนดให้”

สุชาย ธนวเสถียร

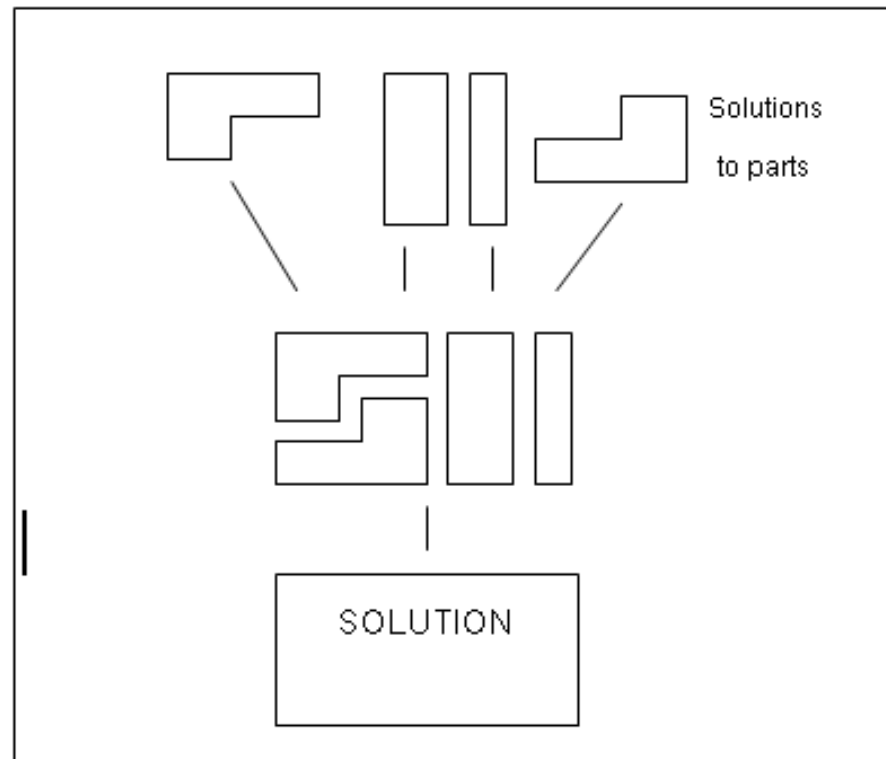
จากหนังสือ Software Project Development

ลักษณะของวิศวกรรมซอฟต์แวร์

1. เกี่ยวข้องกับการสร้างโปรแกรมขนาดใหญ่
2. สามารถจัดการเกี่ยวกับความซ้ำซ้อนได้
3. เน้นการทำงานร่วมกันของบุคลากร
4. สามารถเปลี่ยนแปลงได้ง่ายเมื่อจำเป็น
5. เน้นการพัฒนาให้มีประสิทธิภาพ
6. สนองความต้องการของผู้ใช้

การแก้ปัญหา

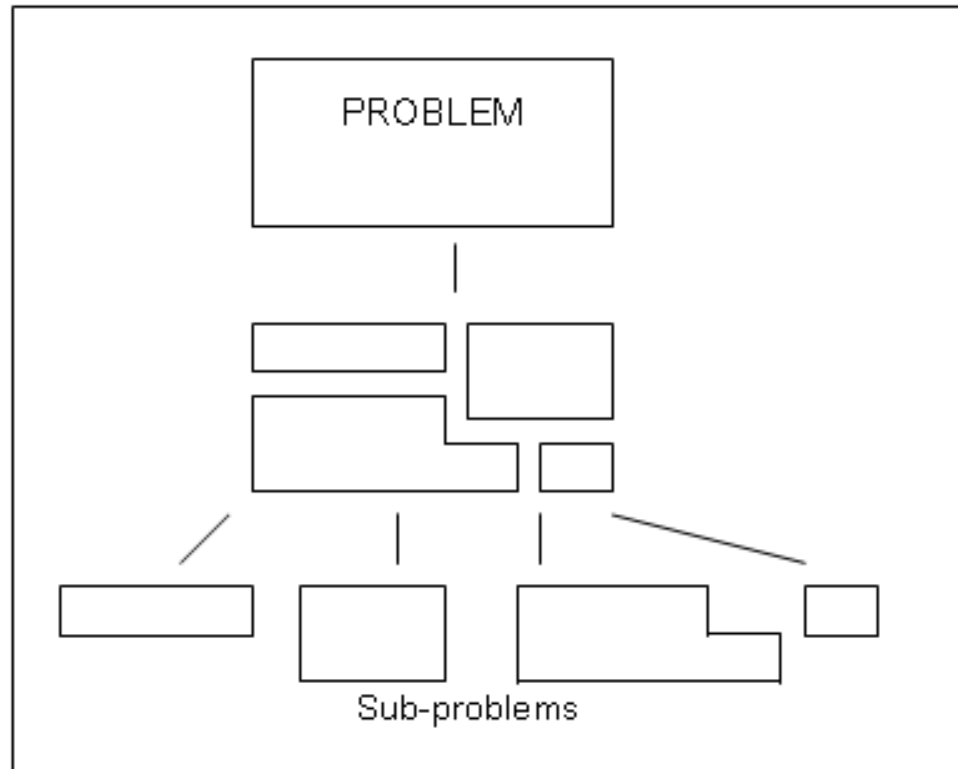
โปรเซสที่มีการแบ่งโครงสร้างขนาดใหญ่ให้เป็นส่วนย่อยๆ



รูปภาพที่ 1.1 The Process of Analysis

การแก้ปัญหา

- การนำปัญหาย่อยๆมารวมเข้าด้วยกันเป็น โครงสร้างที่ใหญ่



รูปภาพที่ 1.2 The Process of Synthesis

วิธีการแก้ปัญหา

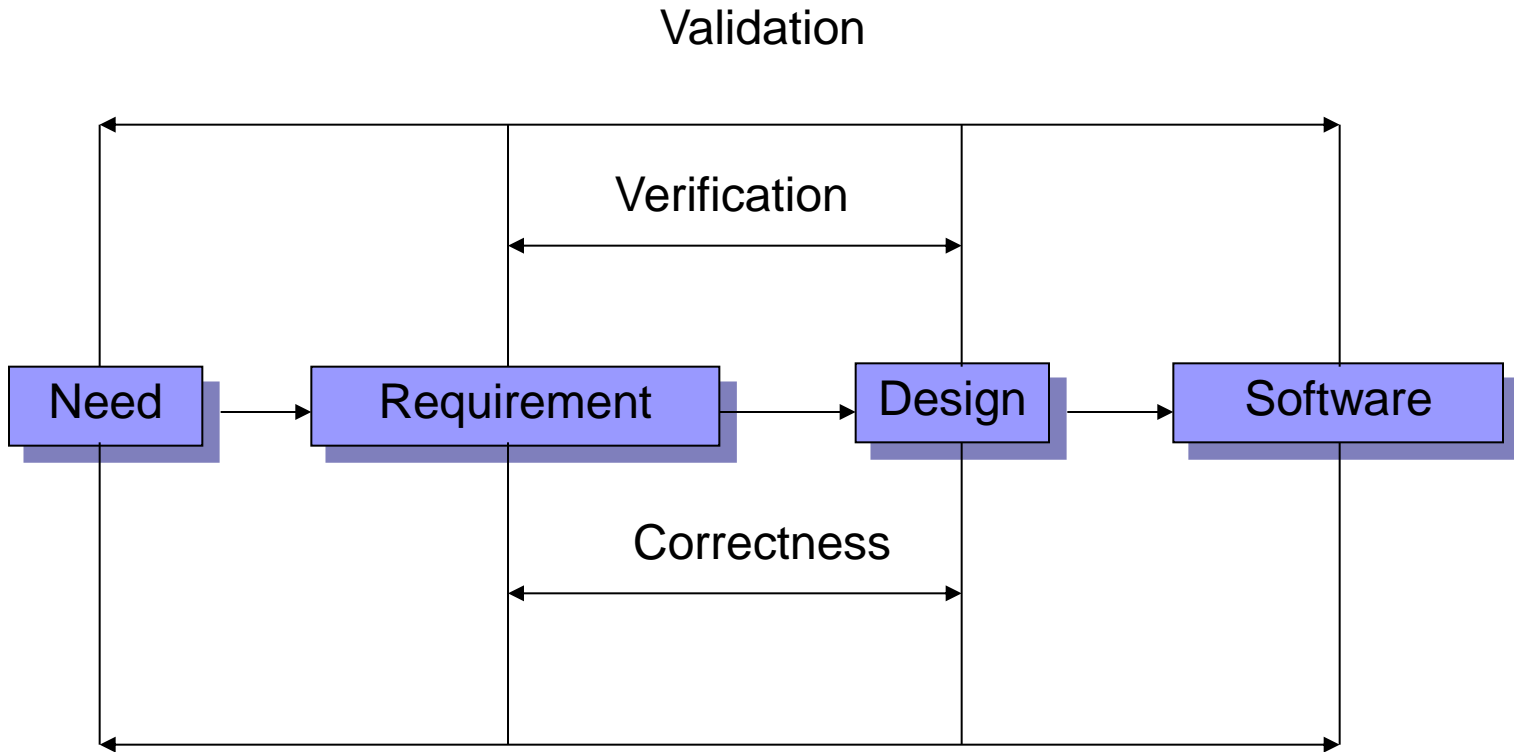
เครื่องมือ(**tools**)

- เครื่องพิมพ์ดีด
- เครื่องคอมพิวเตอร์
- แก้ปัญหาซึ่งให้ความสะดวก ความรวดเร็ว มีคุณภาพสูง และเพิ่มผลผลิตได้มากขึ้น

■ เทคนิค(**techniques**)

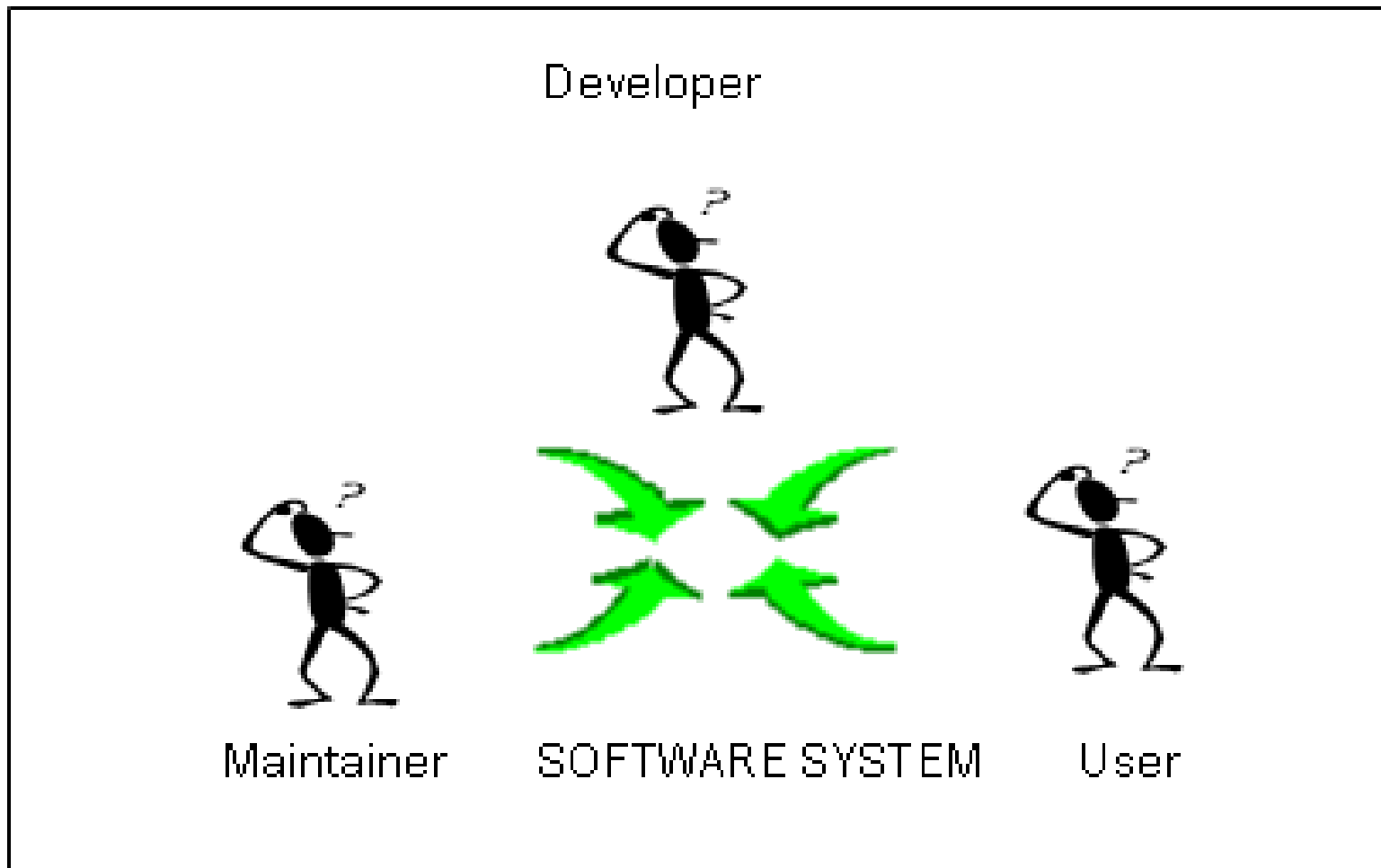
- ประสบการณ์ในการทำงานหรือเป็นพรสวรรค์ของแต่ละบุคคล
- แม่ครัว
- **procedure or method**

ขั้นตอนการพัฒนาซอฟต์แวร์ที่มีคุณภาพ

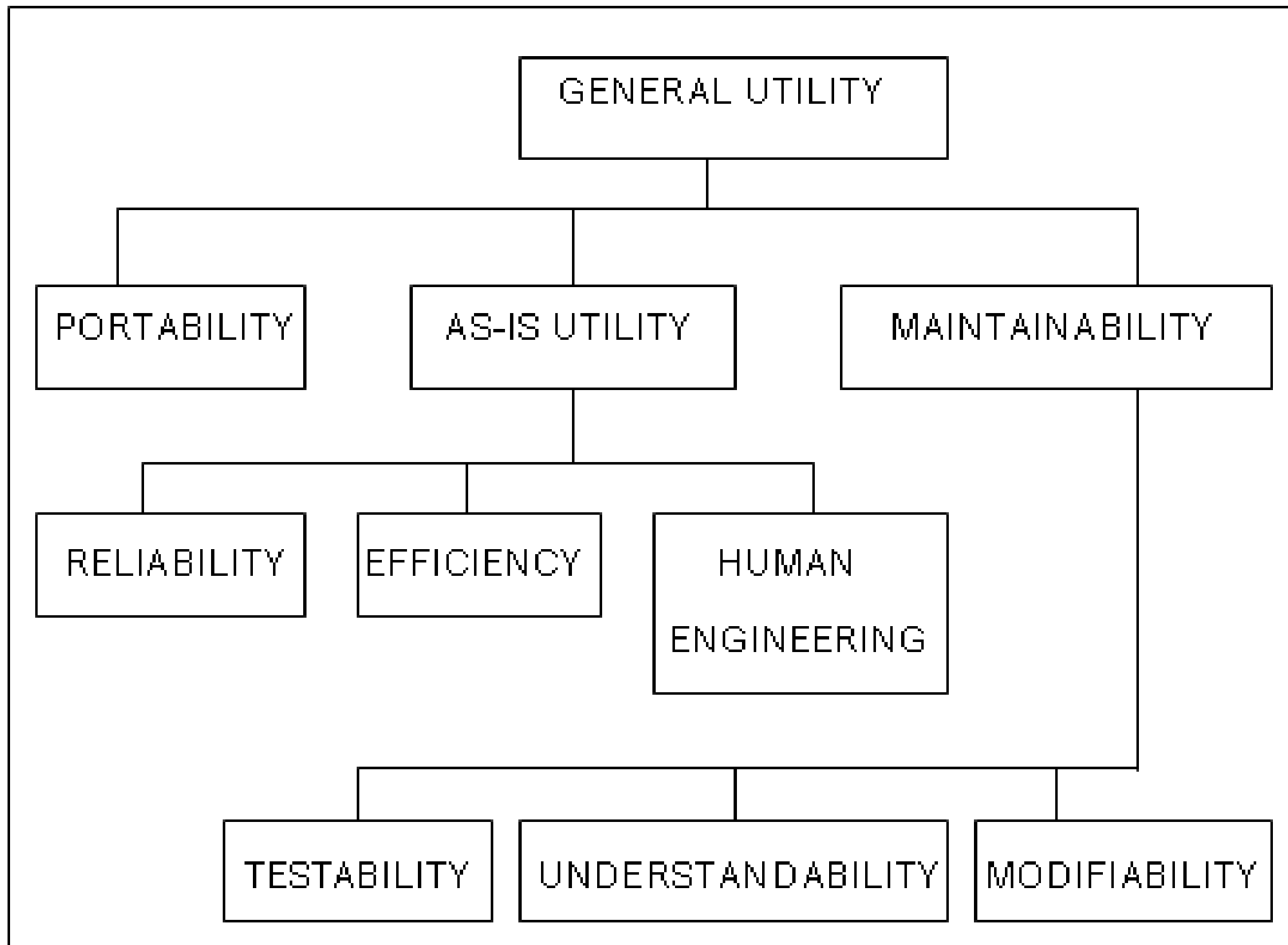


Verification คือ การตรวจสอบความถูกต้องหลังจากการยอมรับในรายละเอียด (Specification)

Validation คือ การตรวจสอบความถูกต้องโดยการพิจารณาจากความต้องการของผู้ใช้งาน



รูปภาพที่ 1.4 Judges of Software Quality



รูปภาพที่ 1.5 Characteristics of software

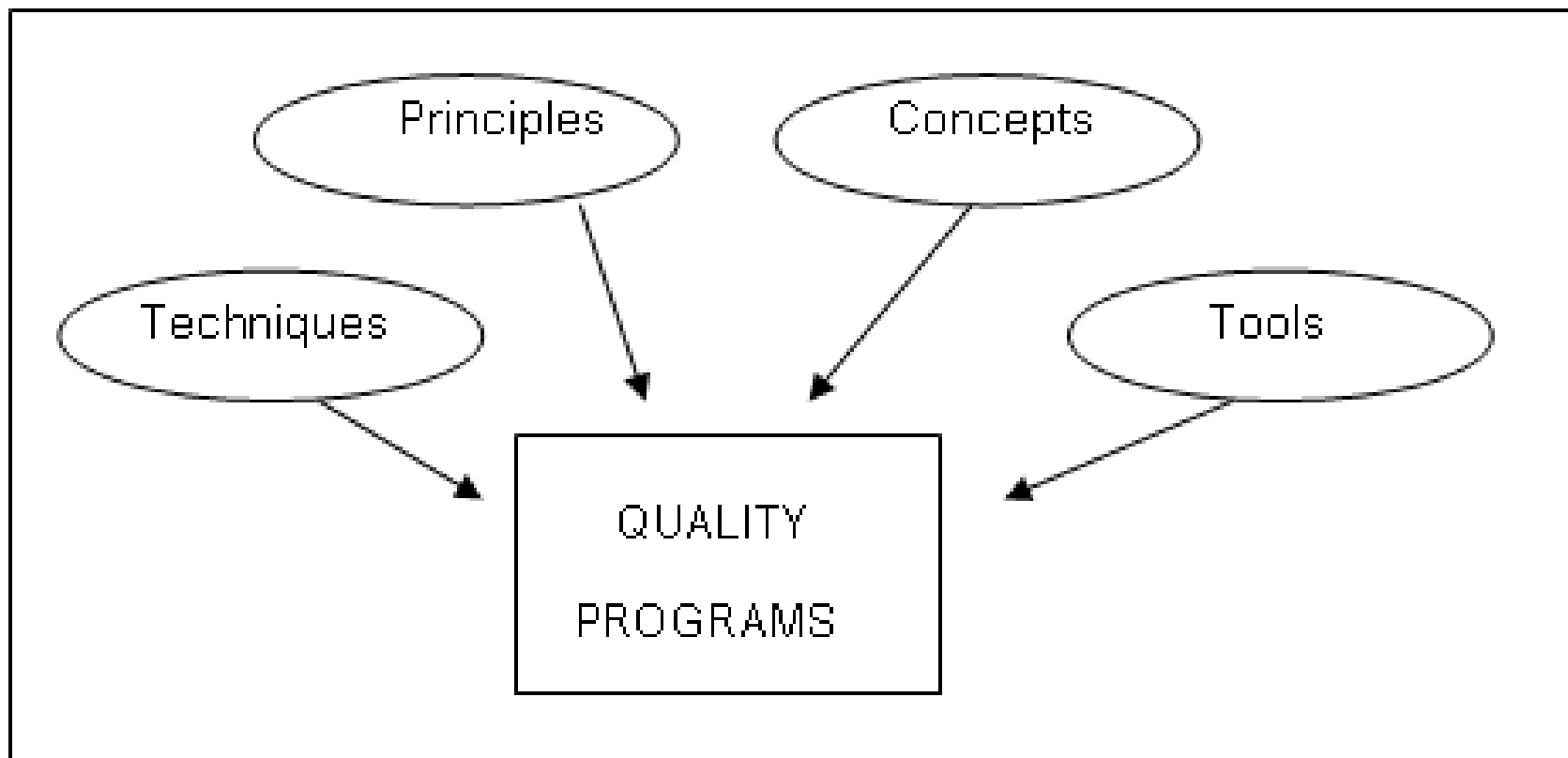
คุณสมบัติของซอฟต์แวร์ที่มีคุณภาพ

- มีความถูกต้อง (Correctness)
- มีความน่าเชื่อถือ (Reliability)
- ใช้งานง่าย (User Friendliness)
- มีความง่ายต่อการปรับเปลี่ยน (Adaptability)
- สามารถนำกลับมาใช้งานใหม่ได้ (Reusability)
- มีความเข้ากันได้กับระบบที่แตกต่าง (Interoperability)
- มีประสิทธิภาพ (Efficiency)
- มีความสะดวกในการเคลื่อนย้าย (Portability)
- มีความปลอดภัย (Security)

ซอฟต์แวร์ที่มีคุณภาพ

- สามารถทำงานตามที่ใช้ต้องการให้กระทำได้อย่างถูกต้อง
- ใช้ทรัพยากรคอมพิวเตอร์อย่างถูกต้องและมีประสิทธิภาพ
- ง่ายต่อการเรียนรู้ และใช้งาน
- ผู้พัฒนาซอฟต์แวร์ สามารถออกแบบ ทดสอบ ทดสอบ และ บำรุงรักษาระบบได้ง่าย

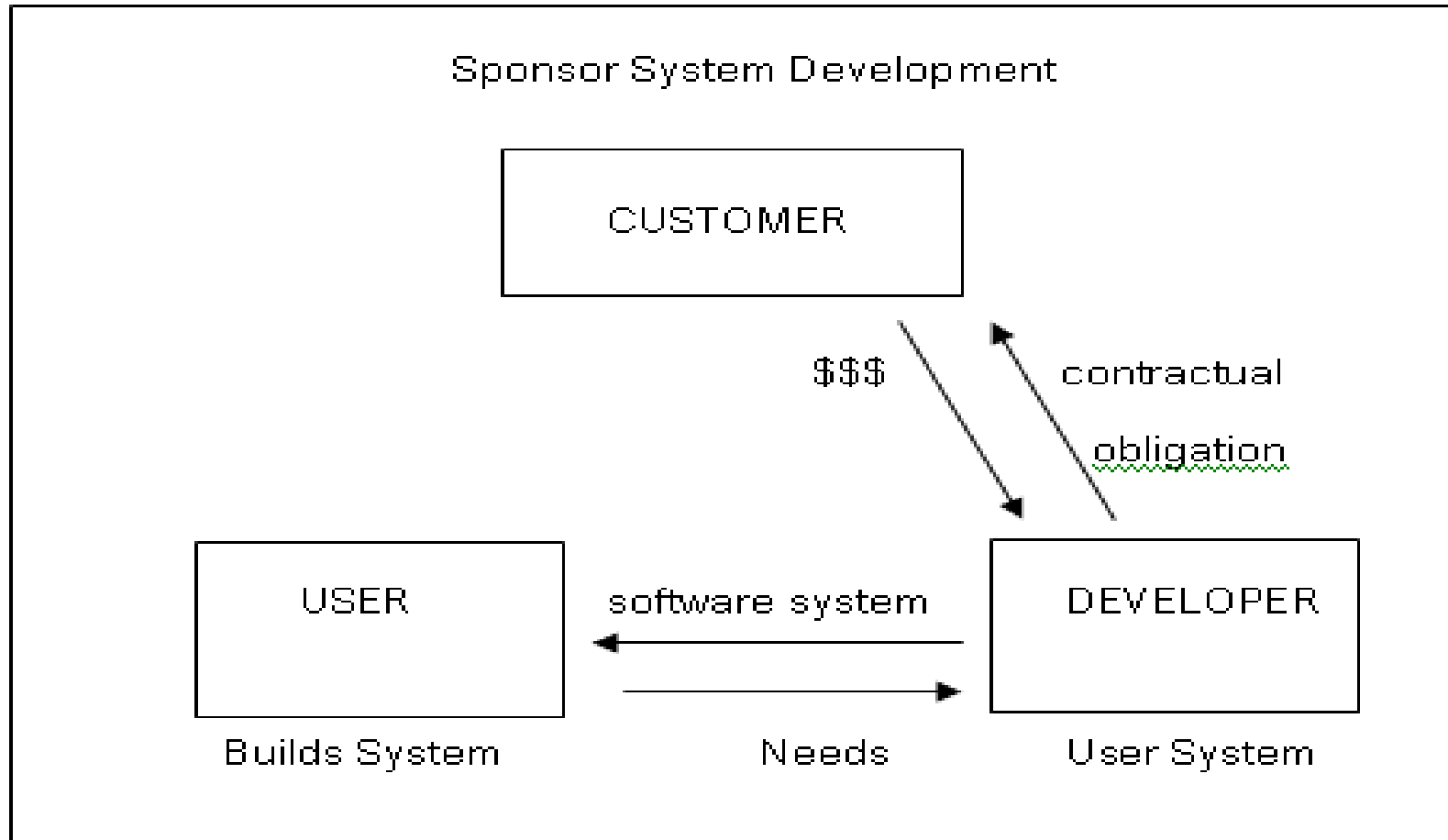
การผลิตซอฟต์แวร์ที่มีคุณภาพ



รูปภาพที่ 1.6 How to Build Quality Programs

งานและความรับผิดชอบ

การพัฒนาโครงการซอฟต์แวร์



รูปภาพที่ 1.7 Relationships among Customer, User, and Developer

ระบบประกอบด้วย

- กิจกรรม(**activities**) หมายถึง เหตุการณ์ต่างๆที่เกิดขึ้นในระบบ
- อาจเป็นกิจกรรมการเคลื่อนย้ายข้อมูล (**move**) จากที่หนึ่งไปอีกที่หนึ่ง

- กิจกรรมหนึ่งๆนั้นประกอบด้วยสมาชิกที่เกี่ยวข้องกัน เรียกว่า **ออฟเจกต์ (objects)** หรือ **เอนตีตี้ (entities)** เช่นกิจกรรมการพิมพ์ ระเบียบนักศึกษาที่เกรดเฉลี่ยน้อยกว่า 2.0

- ขอบเขต(**boundary**)ของระบบ

ENTITIES:

Particular matter

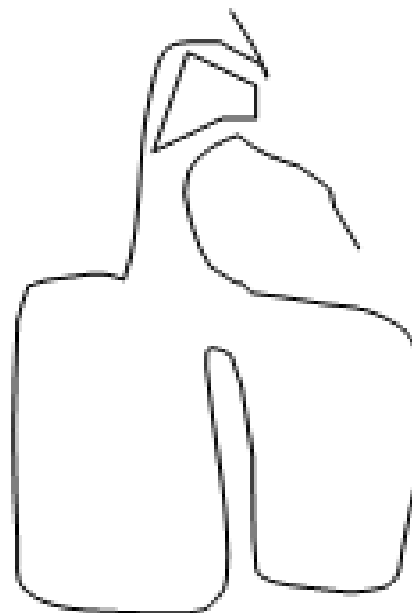
Oxygen, Nitrogen

Carbon dioxide

Carbon monoxide

Lungs, Alveoli

Nose, Mouth, Trachea



← Boundary

ACTIVITIES:

Inhale gases

Filtering gases

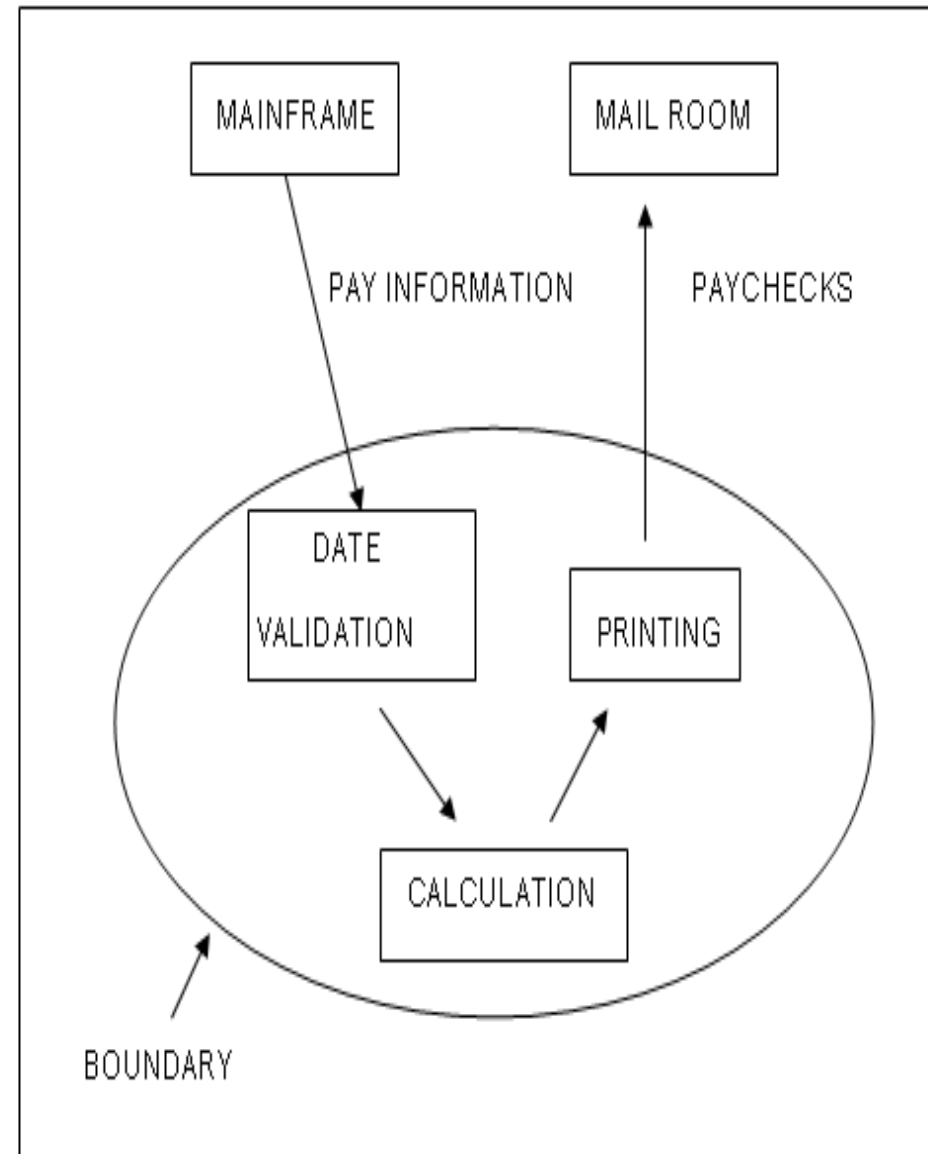
Transfer molecules to /
from blood

Exhale gases

รูปภาพที่ 1.8 Respiratory

ระบบ Paycheck

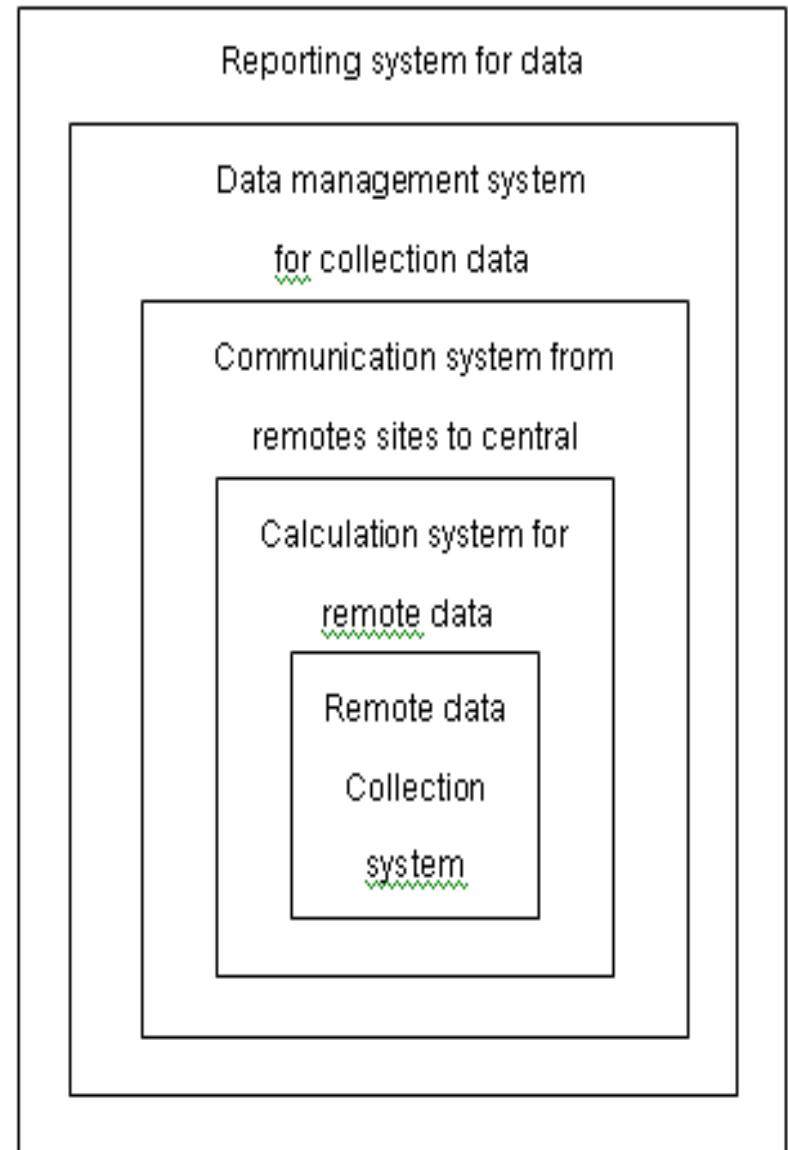
- ระบบนี้เริ่มต้นด้วยการรับอินพุตซึ่งเป็นเอ็นติตี้ **PAY INFORMATION** เข้าสู่ระบบต่อจากนั้นมีกิจกรรมการตรวจสอบวันที่ ต่อด้วย กิจกรรมการคำนวณ และกิจกรรมการพิมพ์ ซึ่งความสัมพันธ์ของแต่ละเอ็นติตี้และกิจกรรมเป็นการทำงานตามลำดับ จุดจบหรือผลลัพธ์ของระบบคือ **PAYCHECKS** ที่ส่งไปยัง **mail room**



รูปภาพที่ 1.9 Paycheck System

Water Monitoring System

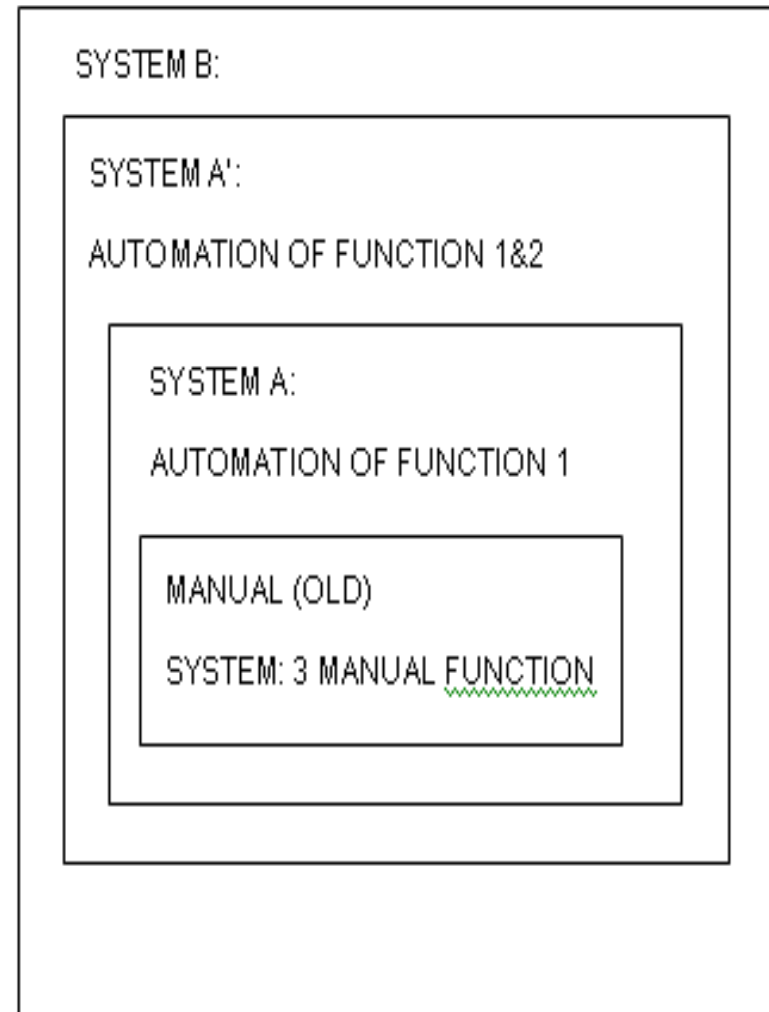
- แสดงถึงการแบ่งระดับของ **Water Monitoring System** ซึ่งในแต่ละระดับระบุหน้าที่การทำงานของระบบอย่างชัดเจน โดยที่เชื่อมโยงแต่ละอันจะแทนขอบเขตของการทำงานของระบบนั้น ในที่นี้ที่เชื่อมโยงที่อยู่ในสุดจะเป็นระบบที่เล็กที่สุด ออฟเจ็ทหรือกิจกรรมของส่วนนี้มีความสัมพันธ์กับที่เชื่อมโยงในระดับนอก และมีความสัมพันธ์กันเกี่ยวโยงกันเป็นลำดับ จนได้ผลลัพธ์ที่ต้องการ



รูปภาพที่ 1.10 Layers of Water Monitoring System

ผู้ใช้ที่ไม่มีความรู้ทางด้านคอมพิวเตอร์ต่อต้านไม่ยอมรับระบบใหม่

- การเปลี่ยนระบบ A ไปเป็นระบบ B ซึ่งการพัฒนาจะค่อยๆเปลี่ยนแปลงจาก A เป็น A' และ เป็น B ตามลำดับ

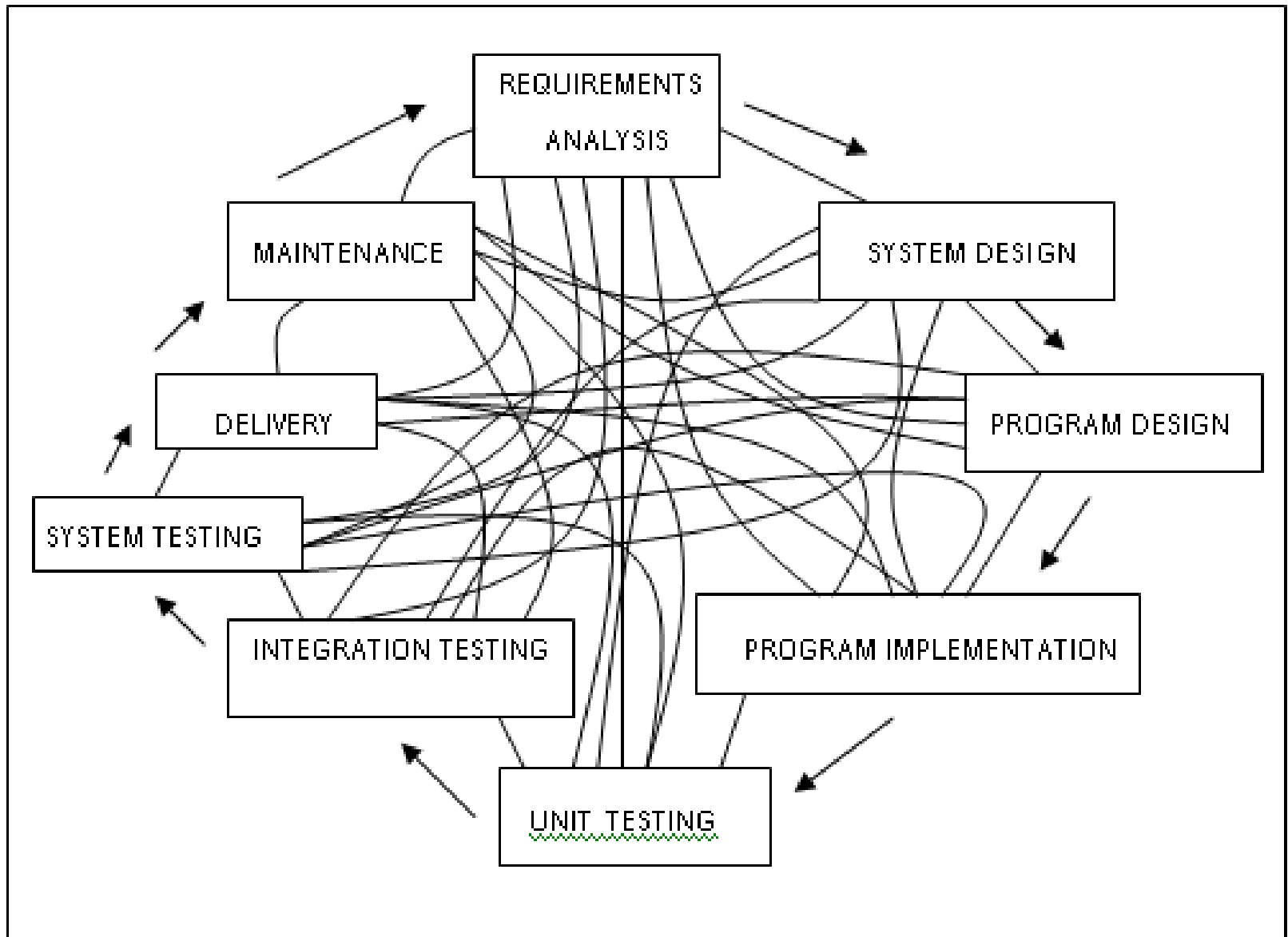


รูปภาพที่ 1.11 Incremental Steps from Old System to New System

ขั้นตอนการพัฒนาซอฟต์แวร์ประกอบด้วย

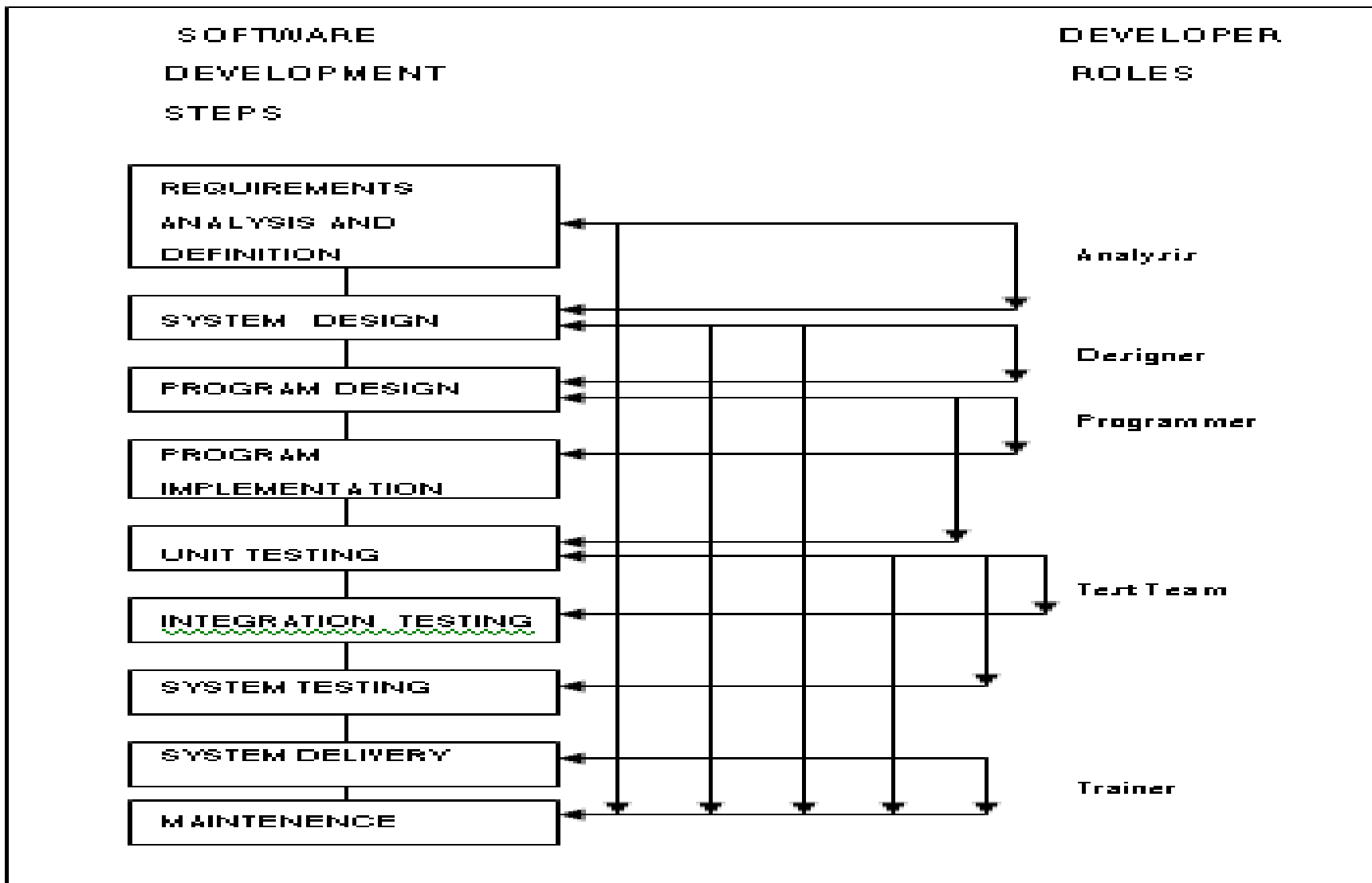
- การกำหนดและวิเคราะห์ความต้องการ
- การออกแบบระบบ
- การออกแบบโปรแกรม
- การเขียนโปรแกรม

- การทดสอบโมดูล
- การทดสอบรวม
- การทดสอบระบบ
- การส่งมอบระบบ
- การบำรุงรักษาระบบ



รูปภาพที่ 1.12 The Development Cycle

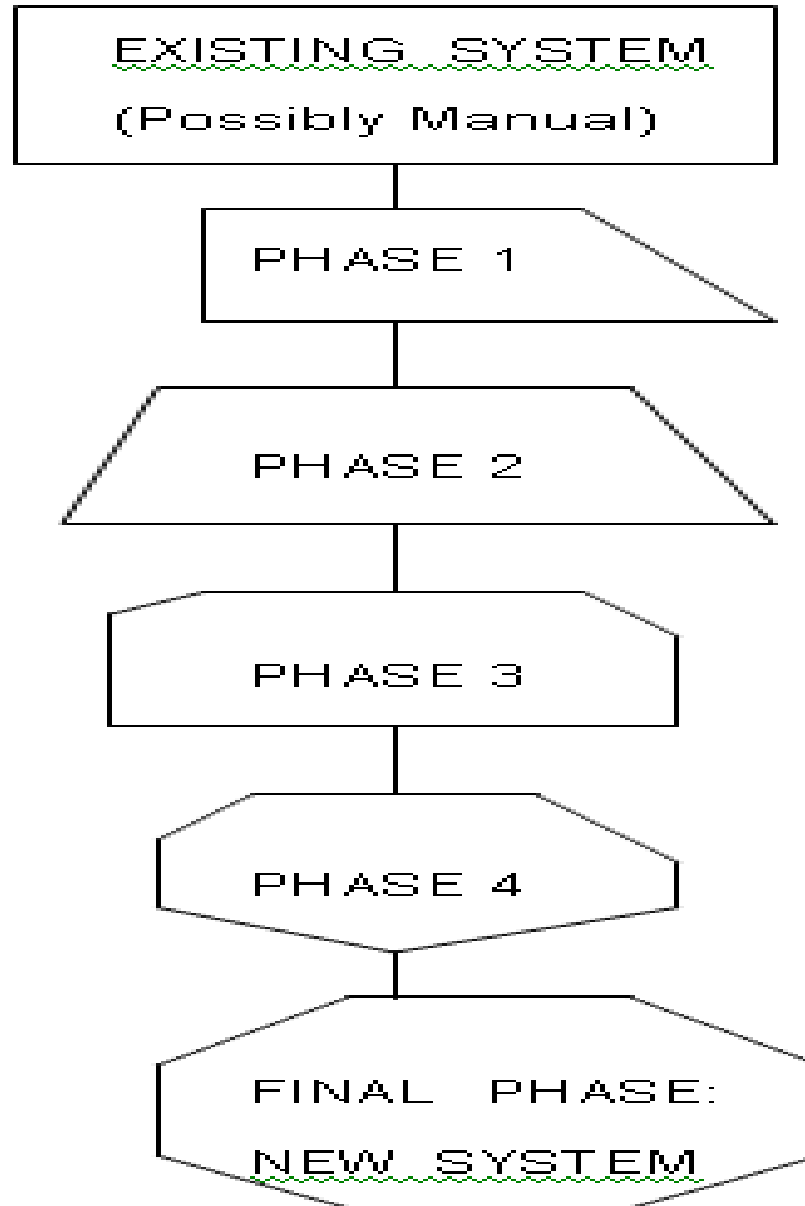
ทีมงานในการพัฒนาระบบ

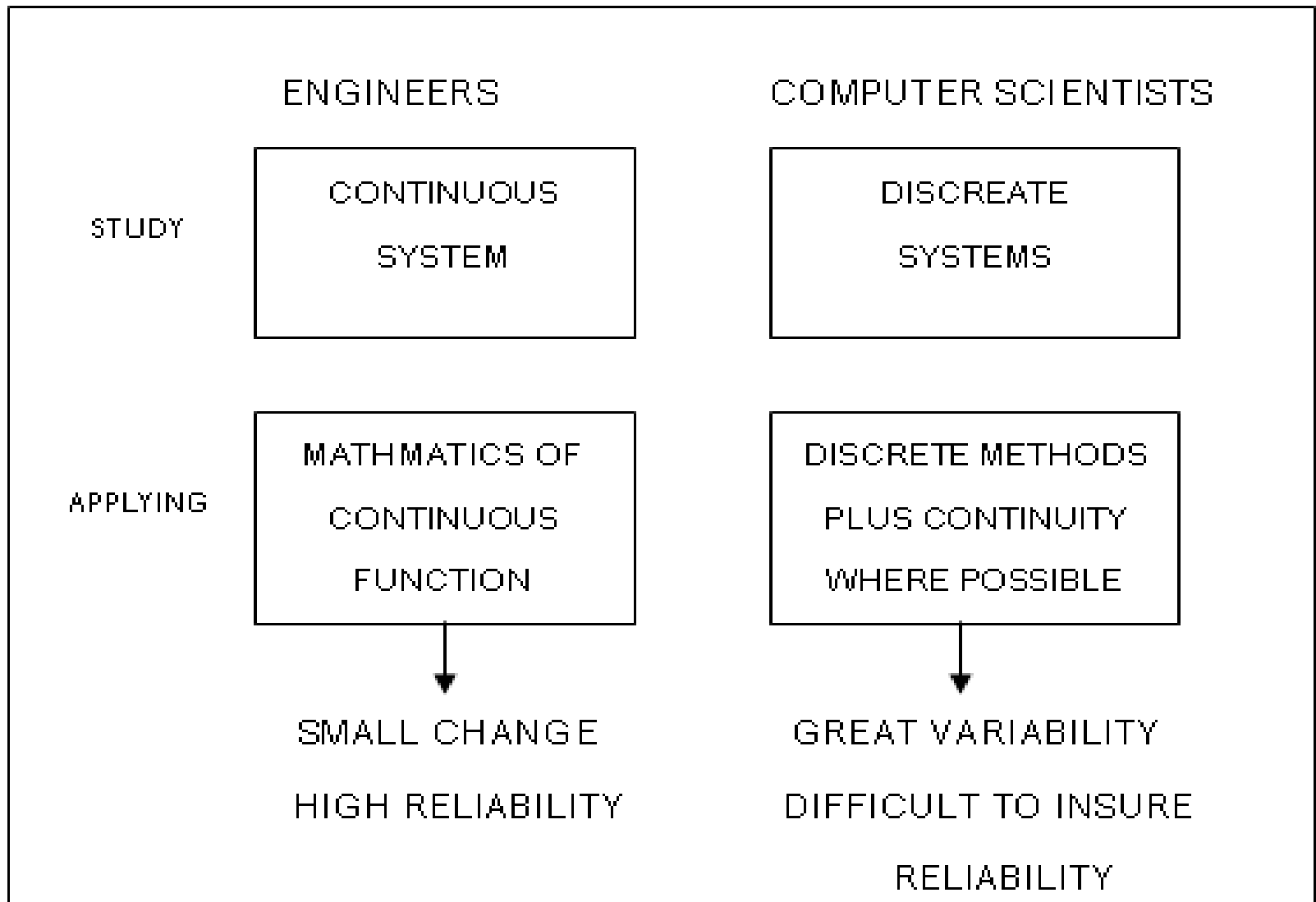


ปัญหาในการพัฒนาซอฟต์แวร์

- Changing Constraints and Requirement
- Phased Development System
- Interaction with Other Systems
- The Nature of Computer System Themselves

Phased Development System





รูปภาพที่ 1.15 Comparison of Engineering with Computer Science

ชนิดของโครงการซอฟต์แวร์

ตารางที่ 1.1 Degrees of difficulty for Project Development

Characteristic	Low	Moderate	High
<u>Number of functions performed</u>	Small	Medium	Large
Novelty of Function	Standard application	Similar to existing system but with a Few new functions	New theory or approach; never been built before
<u>Number of users requiring multi-user or concurrent</u>	1	Several	many

Characteristic	Low	Moderate	High
Multi-tasking	No	Some	Yes
<u>interactive VS.</u> Batch access	Batch or <u>interactive</u>	<u>Highly interactive</u>	highly interactive
Response-time <u>requirements</u>	Off-line; Non-critical	interactive; moderate response time <u>acceptable</u>	Real-time
Need for <u>distributed</u> <u>processing</u>	None	2 computer	3 or more computers

<u>Amount of data stored</u>	Will fit on single disk	Requires 2 or more disk	<u>Requires system to manage disk access</u>
<u>Structure of data</u>	Simple data relationships	<u>Moderately complex relationships</u>	Highly complex relationships
<u>Accuracy of data</u>	Low degree	Moderate degree	High degree
<u>Transaction size</u>	Small	Medium	Large
<u>Remote VS. Local</u>	Local only	Remote	Remote access
<u>Criticality;</u> <u>tolerance for</u> <u>downtime</u>	Can tolerate several hours of downtime	Can tolerate short periods of downtime	Can tolerate no downtime
<u>Security needs</u>	None	Moderate	High

ตารางที่ 1.1 Degrees of difficulty for Project Development (ต่อ)

Characteristic	Low	Moderate	High
Interaction with other systems	None	<u>Some</u> but well-defined	Much; possible parallel development
Number of phases of development	None	Few	Many
Need for manual Override	No	No	Yes
Dependence on hardware	Independent of hardware	Some	Tied to specific hardware constraints

Stability of <u>specification</u>	Fixed customer requirements	Some changes may occur	<u>Frequent changes</u> in specification
User Sophistication	Familiarity with <u>automated system</u>	<u>Some familiarity</u> with automated system	Naive
Developer <u>sophistication</u>	Has developed <u>similar system</u> with <u>similar tools</u>	Experience with tools, not with application	No experience

ระดับความยากง่ายของโครงการนี้ โดยเปรียบเทียบกับข้อมูล

ตารางที่ 1.2 Difficulty of Weaver Farm project Development

Characteristic	Probable Value for Farm	Difficulty
<u>Number of functions performed</u>	Large	H
<u>Novelty of Function</u>	Similar to <u>existing systems</u> but with a few new functions	M
<u>Number of users requiring multi-user or concurrent access</u>	Several	M
<u>Multi-tasking</u>	Some	M
<u>Interactive VS. batch access</u>	Highly interactive	H
<u>Response-time requirements</u>	Interactive; moderate response time acceptable	M

สรุป

- ค่าของความซับซ้อนอยู่ใน

- ระดับต่ำ(L) 5 คุณลักษณะ

- ระดับกลาง(M) 11 คุณลักษณะ

- ระดับสูง(H) 5 คุณลักษณะ

- สามารถสรุปได้ว่าโครงการนี้ระดับความยากง่าย

ในระดับปานกลาง