



Software Engineering

Computer Engineering Department

•การกำหนดคะแนน

• สอบ Midterm	25%
• สอบ Final	25%
• Homework	25%
• Quiz	15%
• เข้าเรียน	10%



•Course Outline

- บทนำ-วิศวกรรมซอฟต์แวร์ (Introduction to Software Engineering)
- ภาพรวมของกระบวนการ (A Generic View of Process)
- แบบจำลองกระบวนการพัฒนาระบบ (Process Model)
- การประมาณการซอฟต์แวร์ (Software Estimation)
- วิศวกรรมความต้องการ (Requirement Engineering)
- การสร้างแบบจำลองการวิเคราะห์ (Analysis Model)
- การวิเคราะห์เชิงวัตถุ (Object Oriented Analysis)
- วิศวกรรมการออกแบบ (Design Engineering)
- การออกแบบส่วนต่อประสานผู้ใช้ (User Interface Design)
- การทดสอบซอฟต์แวร์ (Testing)



•Software Engineer

- Software Engineer เกี่ยวข้องอย่างไรกับกระบวนการผลิตซอฟต์แวร์

ตอบ. เป็นบุคคลในสายงานอาชีพทางด้านคอมพิวเตอร์ที่ทำหน้าที่วิเคราะห์และตรวจสอบซอฟต์แวร์ที่พัฒนาขึ้นอย่างมีแผน โดยอาศัยหลักการทางด้านวิศวกรรมศาสตร์เข้ามาช่วย เช่น วัดค่าความซับซ้อนของซอฟต์แวร์ที่ผลิตขึ้นมา ตรวจสอบว่าการเขียนโปรแกรมนั้นถูกต้องตามหลักหรือไม่

เป็นกลุ่มคนที่มีทักษะและความเข้าใจในการพัฒนาซอฟต์แวร์เป็นอย่างดี สามารถเขียนโปรแกรมได้หลายๆ ภาษา บุคคลเหล่านี้สามารถผลิตซอฟต์แวร์ขนาดใหญ่ได้เป็นอย่างดี



ได้มีการ จัดอันดับอาชีพที่น่าทำที่สุดในปี 2012 ซึ่งอาชีพที่น่าทำที่สุดได้แก่ วิศวกรซอฟต์แวร์ (**Software Engineer**) โดยวัดจากปัจจัยความต้องการ ในตลาดแรงงานที่สูงสุดเป็นประวัติการณ์ รายได้ที่ดี บรรยากาศในการทำงานที่ สบายๆ ต่างจากอาชีพอื่น และใช้ความต้องการทางกายภาพน้อยมาก

อุปสรรคสำหรับคนที่ จะเข้ามาทำงานวิศวกรซอฟต์แวร์ คือต้องใช้ความสามารถ เฉพาะทาง สิ่งที่เคยเรียนมาอาจจะใช้ไม่ได้ตลอดไป ต้องเรียนรู้ตลอดเวลา พร้อมต่อ การเปลี่ยนแปลง ส่วนอาชีพอื่นๆ ที่ได้อันดับรองลงมาคือ นักสถิติ, นักบริหาร ทรัพยากรบุคคล (HR), ทันตแพทย์และนักวางแผนการเงิน



• ลักษณะของวิศวกรรมซอฟต์แวร์

- เกี่ยวข้องกับการสร้างโปรแกรมขนาดใหญ่
- สามารถจัดการเกี่ยวกับความซับซ้อนได้
- เน้นการทำงานร่วมกันของบุคลากร
- สามารถเปลี่ยนแปลงได้ง่ายเมื่อจำเป็น
- เน้นการพัฒนาให้มีประสิทธิภาพ
- สนองความต้องการของผู้ใช้

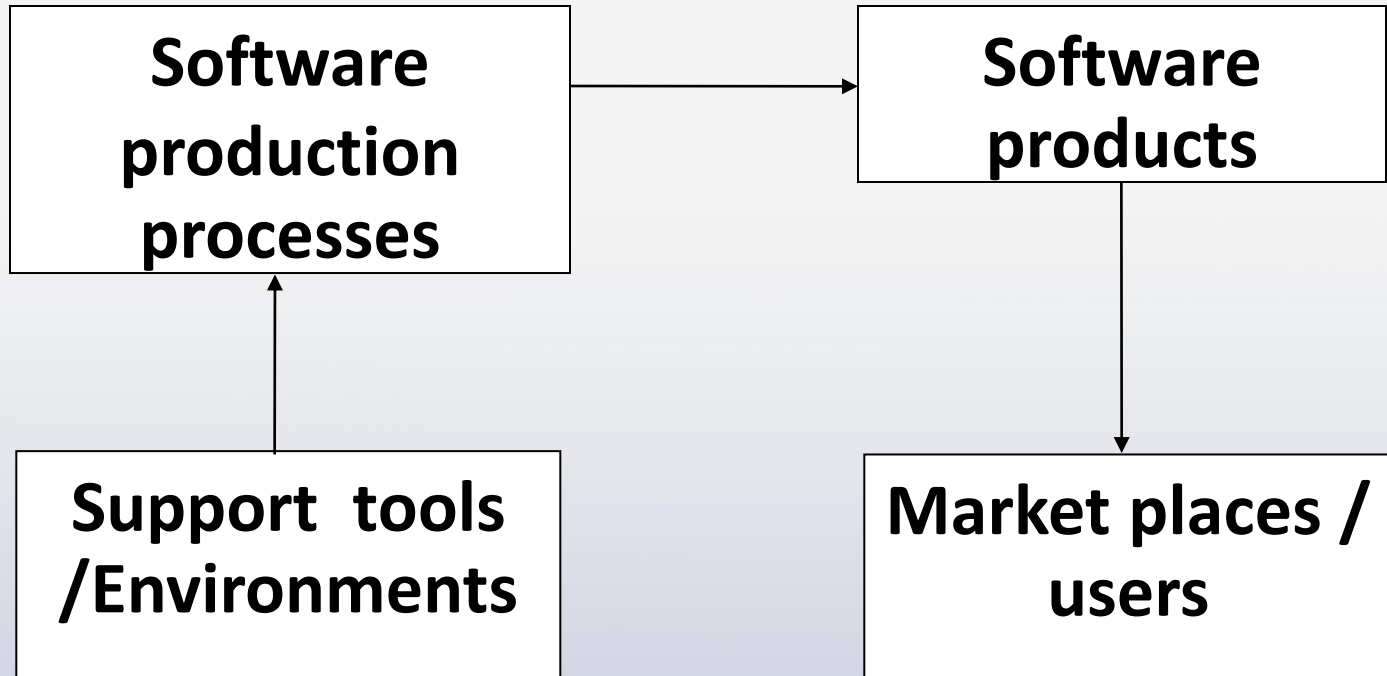


• องค์ประกอบของการวิศวกรรมซอฟต์แวร์

- การวิศวกรรมซอฟต์แวร์เป็นกระบวนการผลิต (production) ที่ประกอบด้วยกิจกรรมช่วงต่างๆ เพื่อสร้างผลิตภัณฑ์ซอฟต์แวร์ (software products) การทำกิจกรรมในแต่ละช่วงอาศัยเทคนิคและเครื่องมือช่วยต่างๆ (support tools) ที่นักวิชาการคอมพิวเตอร์และนักวิจัยได้เสนอไว้



• องค์ประกอบของการวิศวกรรมซอฟต์แวร์



- **คุณลักษณะของกระบวนการวิศวกรรมซอฟต์แวร์**

- **Understandability** : มีการนิยามขอบเขตของกระบวนการที่ชัดเจนและง่ายต่อการเข้าใจ
- **Visibility** : ทำให้กิจกรรมกระบวนการชัดเจนที่สุดเพื่อสามารถมองเห็นจากภายนอกได้ชัดเจน
- **Supportability** : เครื่องมือช่วยการวิศวกรรมซอฟต์แวร์ (CASE) สามารถช่วยสนับสนุนกิจกรรมกระบวนการในขอบเขตใด



- **คุณลักษณะของกระบวนการวิศวกรรมซอฟต์แวร์**
- **Acceptability** : กระบวนการที่กำหนดสามารถยอมรับและใช้โดยวิศวกรซอฟต์แวร์ในการผลิตผลิตภัณฑ์ซอฟต์แวร์
- **Reliability** : กระบวนการถูกออกแบบในแนวทางซึ่งความผิดพลาดของกระบวนการถูกหลีกเลี่ยงก่อนที่จะส่งผลกระทบต่อความผิดพลาดของผลิตภัณฑ์ซอฟต์แวร์
- **Robustness** : กระบวนการสามารถทำงานต่อได้แม้ว่ามีปัญหาที่ไม่คาดการณ์เกิดขึ้น

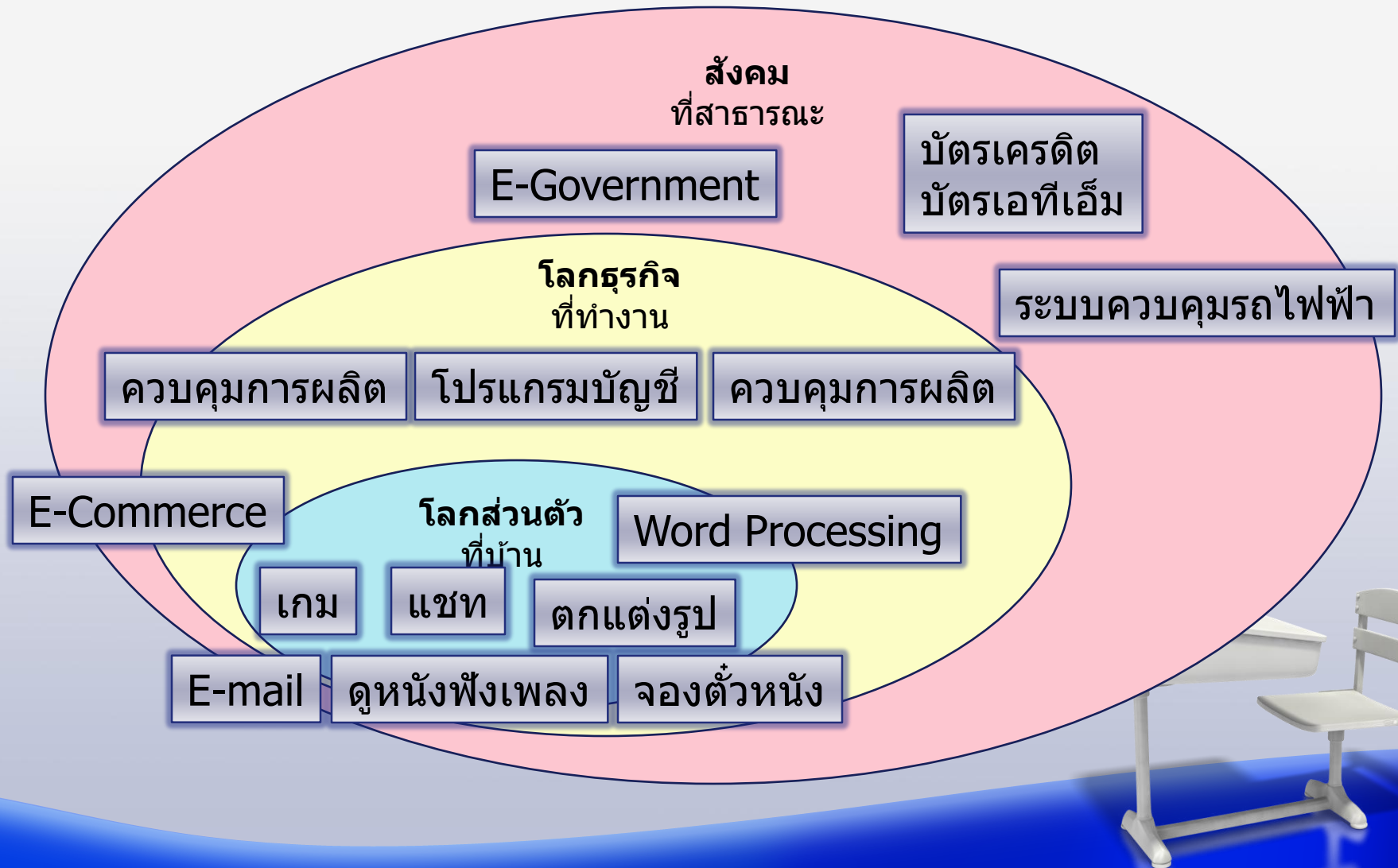


- คุณลักษณะของกระบวนการวิศวกรรมซอฟต์แวร์
- **Maintainability** : กระบวนการสามารถวิวัฒนาการเพื่อตอบสนองการเปลี่ยนแปลงความต้องการขององค์กร
- **Rapidity** : กระบวนการสามารถทำให้ส่งมอบผลิตภัณฑ์ได้เร็วขึ้นจากที่รูปแบบคุณลักษณะของซอฟต์แวร์(Software specifications) ถูกกำหนด



•ซอฟต์แวร์คืออะไร

ซอฟต์แวร์อยู่รอบตัวเรา



•การประยุกต์ใช้ซอฟต์แวร์ในอุปกรณ์ต่าง ๆ

ซอฟต์แวร์ควบคุม

ระบบควบคุมโรงงาน
ระบบควบคุมสัญญาณจราจร

คอมพิวเตอร์ทำงาน
ร่วมกับอุปกรณ์อื่น

ซอฟต์แวร์ที่ถูกฝังในอุปกรณ์ต่าง ๆ

โทรศัพท์มือถือ เครื่องเล่นวีซีดี ดีวีดี
แอร์ เต้าไมโครเวฟ เครื่องยนต์ในรถยนต์

อุปกรณ์อื่น
ที่ไม่ใช่คอมพิวเตอร์

แอปพลิเคชัน (ใช้ในงาน)

ระบบบริหารการขาย ระบบบัญชี
ระบบออนไลน์ของธนาคาร ควบคุมการผลิต

แอปพลิเคชัน (ใช้ส่วนตัว)

เกม อินเทอร์เน็ต โปรแกรมตกแต่งรูป

ซอฟต์แวร์พื้นฐาน

ระบบปฏิบัติการ ภาษาที่ใช้พัฒนาโปรแกรม

คอมพิวเตอร์



•ซอฟต์แวร์ที่ดีเป็นอย่างไร

1. คุณภาพดี
2. ค่าใช้จ่ายต่ำ
3. ใช้งานง่าย
4. มีความปลอดภัยสูง
5. ใช้เวลาพัฒนาไม่นาน
6. ใช้ทรัพยากรฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ
7. ต้องเป็นทรัพยากรที่นำมาใช้งานได้โดยง่าย

มุมมองของ
ผู้พัฒนาซอฟต์แวร์และผู้ใช้

มุมมองของ
ผู้พัฒนาซอฟต์แวร์



•Introduction

- ซอฟต์แวร์ (software) หมายถึงชุดคำสั่งหรือโปรแกรมที่ใช้สั่งงานให้คอมพิวเตอร์ทำงาน ซอฟต์แวร์จึงหมายถึงลำดับขั้นตอนการทำงานที่เขียนขึ้นด้วยคำสั่งของคอมพิวเตอร์ คำสั่งเหล่านี้เรียงกันเป็นโปรแกรมคอมพิวเตอร์
- ซอฟต์แวร์ หมายถึงโปรแกรมคอมพิวเตอร์ทุกประเภทที่ทำให้คอมพิวเตอร์ทำงานได้



•Introduction

- คุณสมบัติของซอฟต์แวร์
 - ซอฟต์แวร์ถูกพัฒนาหรือจัดการให้เกิดขึ้น
 - ซอฟต์แวร์ไม่สึกหรอ
 - ถูกสร้างตามแบบที่ลูกค้าต้องการ (Custom build)



- ซอฟต์แวร์ (software)

- ชนิดของซอฟต์แวร์

หากแบ่งแยกชนิดของซอฟต์แวร์ตามสภาพการทำงาน แบ่งแยกซอฟต์แวร์ได้เป็น 2 ประเภท คือ

- ซอฟต์แวร์ระบบ (system software)
- ซอฟต์แวร์ประยุกต์ (application software)



- ซอฟต์แวร์ (software)

- ซอฟต์แวร์ระบบ คือ ซอฟต์แวร์ที่บริษัทผู้ผลิตสร้างขึ้นมาเพื่อใช้จัดการกับระบบ หน้าที่การทำงานของซอฟต์แวร์ระบบคือดำเนินงานพื้นฐานต่าง ๆ ของระบบคอมพิวเตอร์ เช่น รับข้อมูลจากแผงแป้นอักขระแล้วแปลความหมายให้คอมพิวเตอร์เข้าใจ นำข้อมูลไปแสดงผลบนจอภาพหรือนำออกไปยังเครื่องพิมพ์ จัดการข้อมูลในระบบแฟ้มข้อมูลบนหน่วยความจำรอง

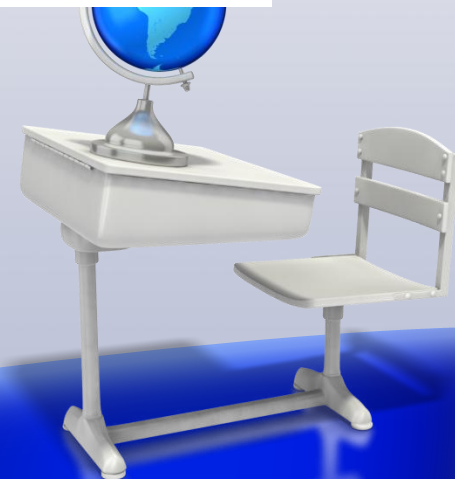


- ซอฟต์แวร์ (software)

- ซอฟต์แวร์ประยุกต์ เป็นซอฟต์แวร์ที่ใช้กับงานด้านต่าง ๆ ตามความต้องการของผู้ใช้ ที่สามารถนำมาใช้ประโยชน์ได้โดยตรง ปัจจุบันมีผู้พัฒนาซอฟต์แวร์ใช้งานทางด้านต่าง ๆ ออกจำหน่ายมาก การประยุกต์งานคอมพิวเตอร์จึงกว้างขวางและแพร่หลาย เราอาจแบ่งซอฟต์แวร์ประยุกต์ออกเป็นสองกลุ่มคือ ซอฟต์แวร์สำเร็จ และซอฟต์แวร์ที่พัฒนาขึ้นใช้งานเฉพาะ



- ซอฟต์แวร์ (software)



•ซอฟต์แวร์ (software)

• ชนิดของซอฟต์แวร์

ในปัจจุบัน สามารถแบ่งลักษณะของซอฟต์แวร์ออกเป็น 7 ประเภท ดังนี้

1. ซอฟต์แวร์ระบบ (System Software)
2. ซอฟต์แวร์ประยุกต์ (Application Software)
3. ซอฟต์แวร์เชิงวิศวกรรม/วิทยาศาสตร์ (Engineering/Scientific Software)
4. ซอฟต์แวร์ฝังตัว (Embedded Software)



•ซอฟต์แวร์ (software)

5. ซอฟต์แวร์สายการผลิต (Product-line Software)
6. เว็บแอปพลิเคชัน (Web-application)
7. ซอฟต์แวร์ปัญญาประดิษฐ์ (Artificial Intelligence Softwear)



•วิวัฒนาการของซอฟต์แวร์ (Software Evolution)

- กฎแห่งการเปลี่ยนแปลงอย่างต่อเนื่อง (Law of Continuing Change)
- กฎแห่งความซับซ้อนที่เพิ่มขึ้น (Law of Increasing Complexity)
- กฎแห่งการวางระเบียบตัวเอง (Law of Self-regulation)
- กฎแห่งอนุรักษ์สภาพเสถียรการจัดระเบียบ (Law of Conservation of Organizational Stability)
- กฎการคงไว้ซึ่งความคุ้นเคย (Law of Conservation of Familiarity)



•ซอฟต์แวร์ประเภทต่าง ๆ

- แบ่งตามขนาด
- แบ่งเป็นเลเยอร์(ชั้น)
- แบ่งตามการนำไปใช้งาน
- แบ่งตามที่มา
- แบ่งตามความจำเพาะเจาะจงในการใช้งาน
- แบ่งตามลักษณะข้อมูล



• การแบ่งประเภทซอฟต์แวร์ตามขนาด

ขนาดขึ้นอยู่กับจำนวนผู้ใช้ และความแพร่หลายในการใช้งาน

- ขนาดใหญ่ : เช่น ระบบสำรวจอวกาศ

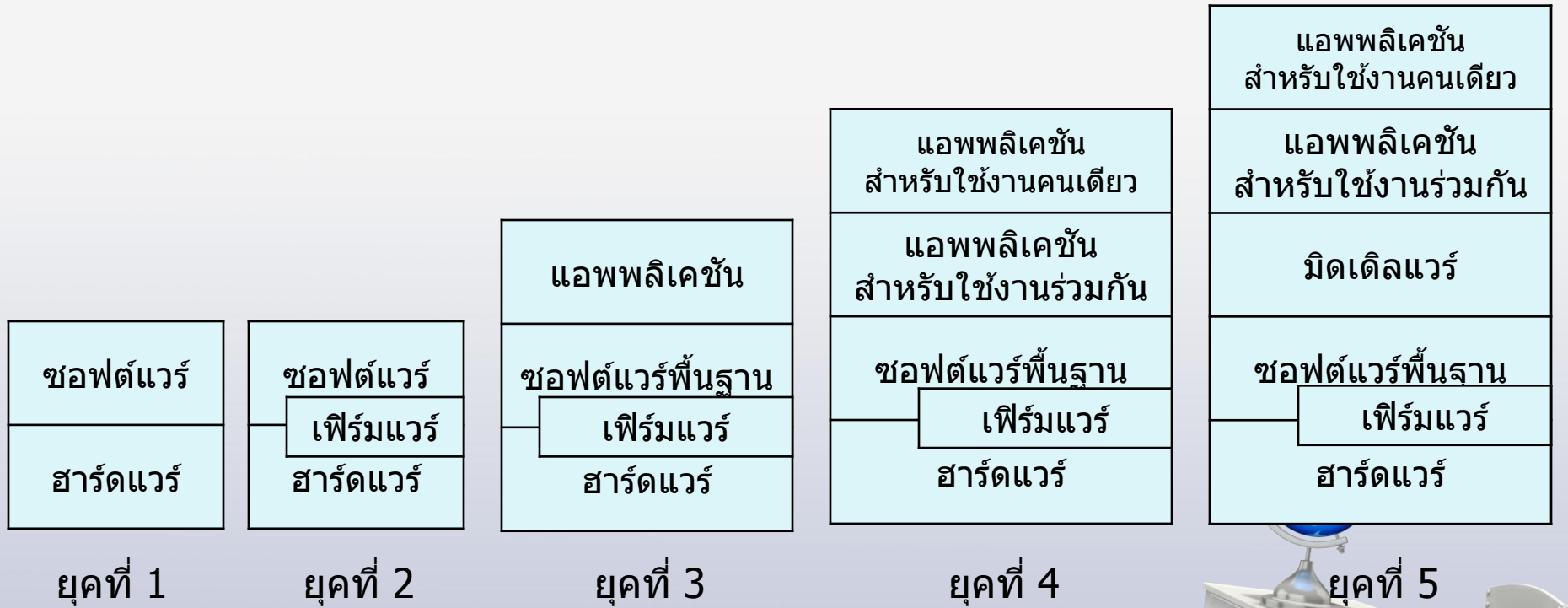
ควบคุมการบิน ควบคุมจรวด

- ขนาดกลาง : เช่น ระบบบริหารการผลิต

- ขนาดเล็ก : เช่น ระบบคำนวณเงินเดือนพนักงาน



•การแบ่งซอฟต์แวร์เป็นเลเยอร์ (ชั้น)



• การแบ่งตามความจำเพาะเจาะจงในการ ใช้งาน

แบ่งเป็น 3 ประเภท

- ใช้งานเฉพาะอย่าง
 - ทำงานได้ตามที่ลูกค้ากำหนด ตอบสนองความต้องการของลูกค้า
- ใช้งานกึ่งเฉพาะอย่าง(ใช้งานกึ่งทั่วไป)
 - ซอฟต์แวร์แพ็คเกจ ต้องมีการปรับแก้ให้เหมาะสมกับลูกค้าแต่ละราย
- ใช้งานทั่วไป
 - สามารถใช้งานได้ทุกที่ หรือ ซอฟต์แวร์พื้นฐาน



- **วิวัฒนาการของซอฟต์แวร์ (Software Evolution)**
 - กฎแห่งการเติบโตอย่างต่อเนื่อง (Law of Continuing Growth)
 - กฎแห่งการลดลงซึ่งคุณภาพ (Law of Declining Quality)
 - กฎระบบย้อนกลับ (Feedback System Law)



•การพัฒนาซอฟต์แวร์

วงจรชีวิตของซอฟต์แวร์



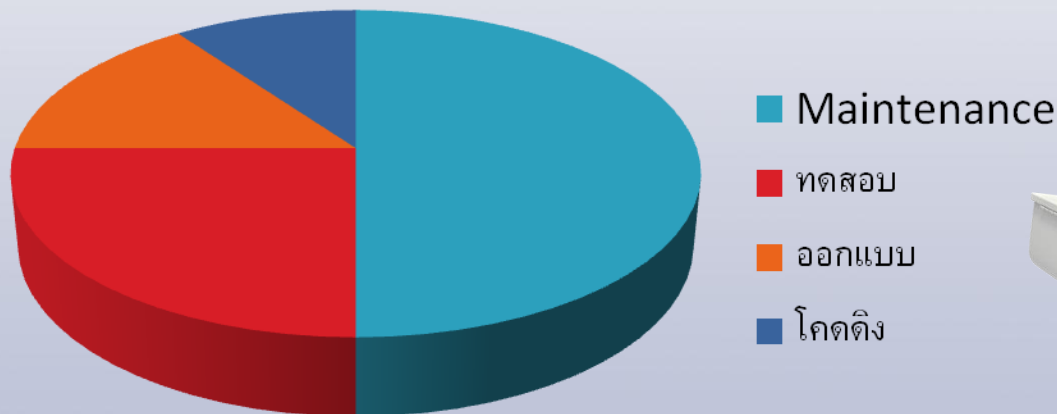
1. วางแผน	สรุปความต้องการ
	ออกแบบภายนอก
2. พัฒนา	ออกแบบภายใน
	โคดดิ้ง
	ทดสอบ
3. ใช้งานและเมนเทน (บำรุงรักษา)	ใช้งาน
	เมนเทน



• การเมนเทนนั้นสำคัญอย่างไร

การเมนเทน หมายถึง งานปรับแก้ซอฟต์แวร์เพื่อแก้ไขข้อบกพร่องที่พบระหว่างการใช้งาน เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้น และให้ตอบสนองความต้องการใหม่

สัดส่วนต้นทุนของซอฟต์แวร์



•ซอฟต์แวร์เอ็นจินีเยริงกับการพัฒนาซอฟต์แวร์

วิกฤตก่อให้เกิดซอฟต์แวร์เอ็นจินีเยริง

ซอฟต์แวร์เอ็นจินีเยริง
ถูกเสนอในปี **1968**



•ความท้าทายของซอฟต์แวร์เอ็นจีเนียริง



•การกำหนดสเปก

ความต้องการที่แสดงออกมา
ภายนอกเปรียบเสมือนยอดของ
ภูเขาน้ำแข็ง ส่วนที่เรามองเห็น
จะลอยอยู่เหนือน้ำ

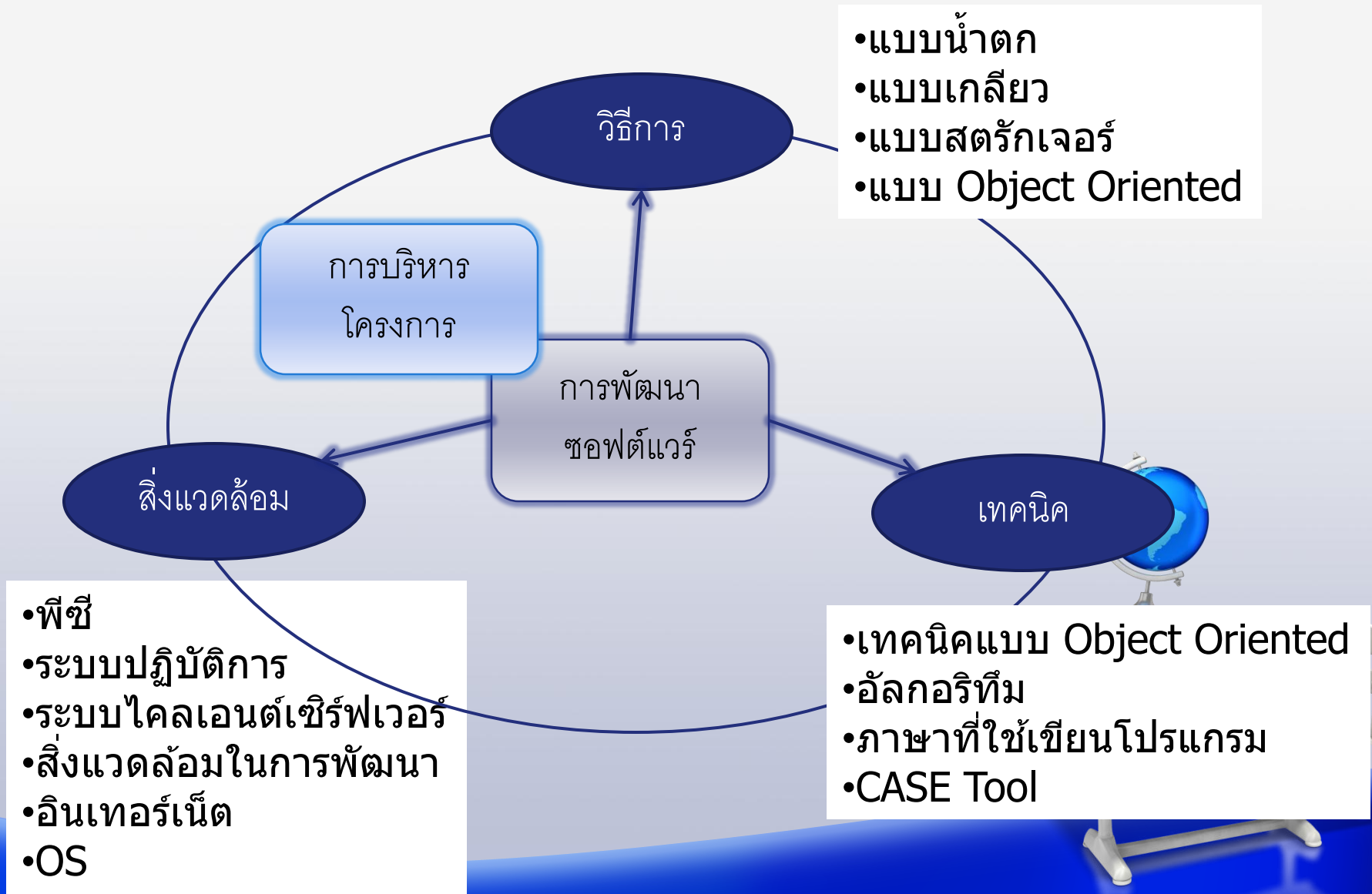
ความต้องการที่ซ่อนเร้นอยู่
ไม่ปรากฏออกมาให้เห็น

ตัวภูเขาน้ำแข็งส่วนใหญ่
จมอยู่ใต้น้ำ

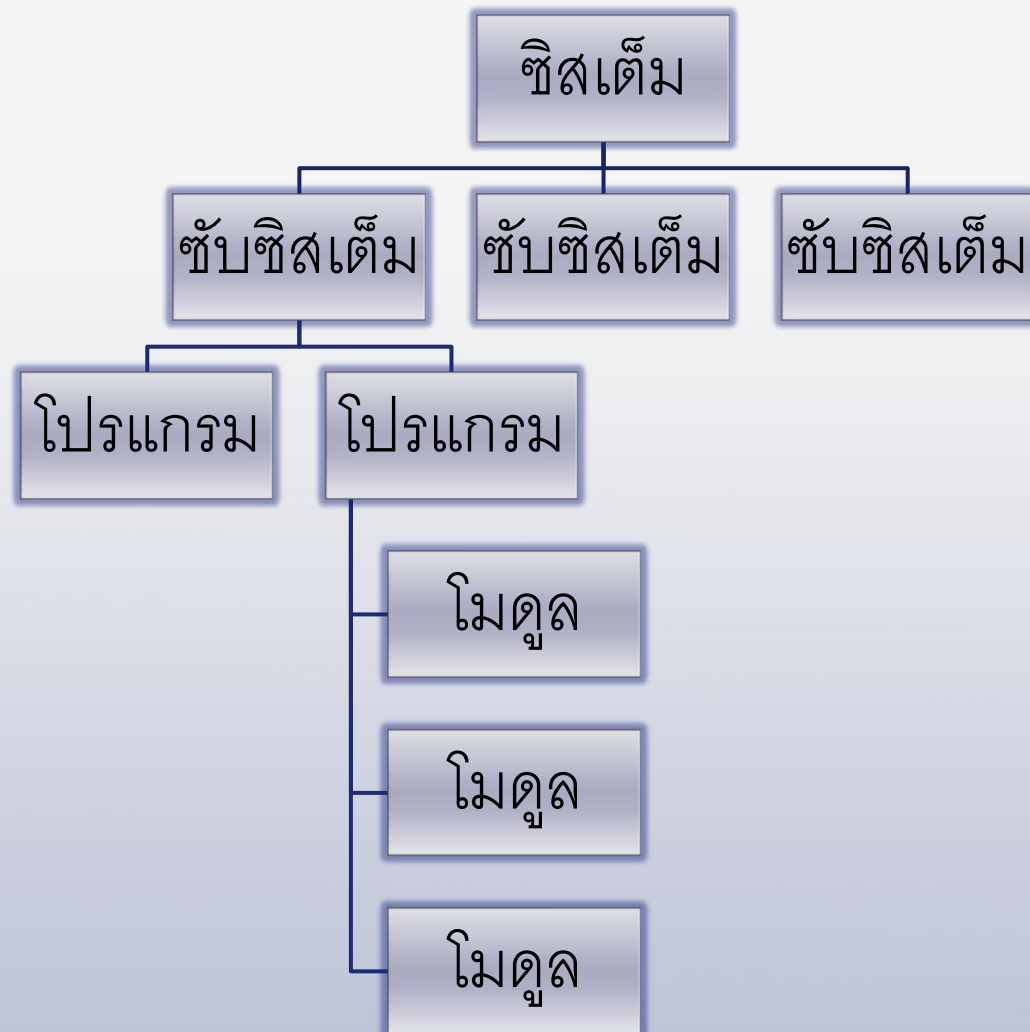
สิ่งที่ต้องระวังในการนิยามความต้องการ



องค์ประกอบของซอฟต์แวร์เอ็นจิเนียริง



•การแบ่งงานย่อย



•What is a software process?

- ▶ เป็นการทำงานตามกิจกรรมต่าง ๆ เพื่อการกำหนดคุณสมบัติ การออกแบบ การเขียนหรือสร้าง และการทดสอบ
- ▶ กิจกรรมโดยทั่วไป สามารถแบ่งได้ดังนี้
 - **Specification** - การกำหนดหน้าที่การทำงานและเงื่อนไขของซอฟต์แวร์ที่จะถูกสร้าง
 - **Development** - การผลิตซอฟต์แวร์ที่ตรงกับรูปแบบที่ได้กำหนดไว้
 - **Validation** - การตรวจสอบซอฟต์แวร์ให้ตรงกับความต้องการของลูกค้า
 - **Evolution** - ซอฟต์แวร์มีวิวัฒนาการตามการเปลี่ยนแปลงของผู้ใช้



•กระบวนการซอฟต์แวร์ (Software Process)

1.ข้อกำหนดซอฟต์แวร์ (software specification)

นิยามหน้าที่ต่างๆที่ต้องมีในซอฟต์แวร์ และระบุข้อจำกัดต่างๆ ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ เช่น กฎหมาย , อัตราภาษี , กฎระเบียบต่างๆที่เกี่ยวข้องในการพัฒนาซอฟต์แวร์

2. การออกแบบและพัฒนาซอฟต์แวร์ (Software Design and Implementation)

กิจกรรมนี้ทำการสร้าง / พัฒนาซอฟต์แวร์ให้ตรงกับข้อกำหนด (specification) ด้วยการทำการมวิธีการพัฒนาซอฟต์แวร์ (Methodology) มาใช้กับการพัฒนาซอฟต์แวร์



•กระบวนการซอฟต์แวร์ (Software Process)

3. การตรวจสอบความถูกต้องของซอฟต์แวร์ (software validation)

กิจกรรมนี้ทำการตรวจสอบความถูกต้องของซอฟต์แวร์ เพื่อให้เกิดความมั่นใจว่าซอฟต์แวร์ที่ผลิตขึ้นได้ตรงกับความต้องการของลูกค้า

4. วิวัฒนาการของซอฟต์แวร์ (software evolution)

ในทางปฏิบัติ เมื่อซอฟต์แวร์ใช้งานได้ระยะหนึ่งแล้ว ผู้ใช้หรือลูกค้าอาจมีความต้องการเพิ่มเติมหรือเปลี่ยนแปลงความต้องการบางอย่าง ดังนั้นขั้นตอนการพัฒนาซอฟต์แวร์ ต้องมีการเตรียมการบางอย่างเพื่อจัดการกับเหตุการณ์ที่คาดหมายว่าจะเกิดขึ้นในอนาคต



•คุณสมบัติของซอฟต์แวร์ที่มีคุณภาพ

• Understandability

- มีการนิยามขอบเขตกระบวนการที่ชัดเจนและง่ายต่อความเข้าใจ

• Visibility

- ทำให้กิจกรรม กระบวนการมีความชัดเจนที่สุด

• Supportability

- เครื่องมือช่วยในการวิเคราะห์ซอฟต์แวร์ สามารถช่วยสนับสนุนกิจกรรมกระบวนการในขอบเขตได้

• Acceptability

- กระบวนการที่กำหนดสามารถยอมรับและใช้ได้โดยวิศวกรต่อผลิตภัณฑ์ซอฟต์แวร์นั้น



•คุณสมบัติของซอฟต์แวร์ที่มีคุณภาพ

▶ มีความถูกต้อง (**Correctness**)

- ความถูกต้องของซอฟต์แวร์กับความต้องการของผู้ใช้มีความตรงกัน

▶ มีความน่าเชื่อถือ (**Reliability**)

- ความน่าเชื่อถือในผลลัพธ์และข้อมูล

▶ ใช้งานง่าย (**User Friendliness**)

- ซอฟต์แวร์มีลักษณะการใช้งานที่เป็นมิตรต่อผู้ใช้งาน ใช้งานง่าย เรียนรู้ง่าย มีข้อความช่วยเหลือหรือคำอธิบายที่ครบถ้วน

▶ บำรุงรักษาง่าย (**Maintainability**)

- กระบวนการสามารถวิวัฒนาการเพื่อตอบสนองต่อการเปลี่ยนแปลงความต้องการขององค์กรในอนาคต



