

บทที่ 1 การแนะนำวิศวกรรมซอฟต์แวร์เบื้องต้น

โดย

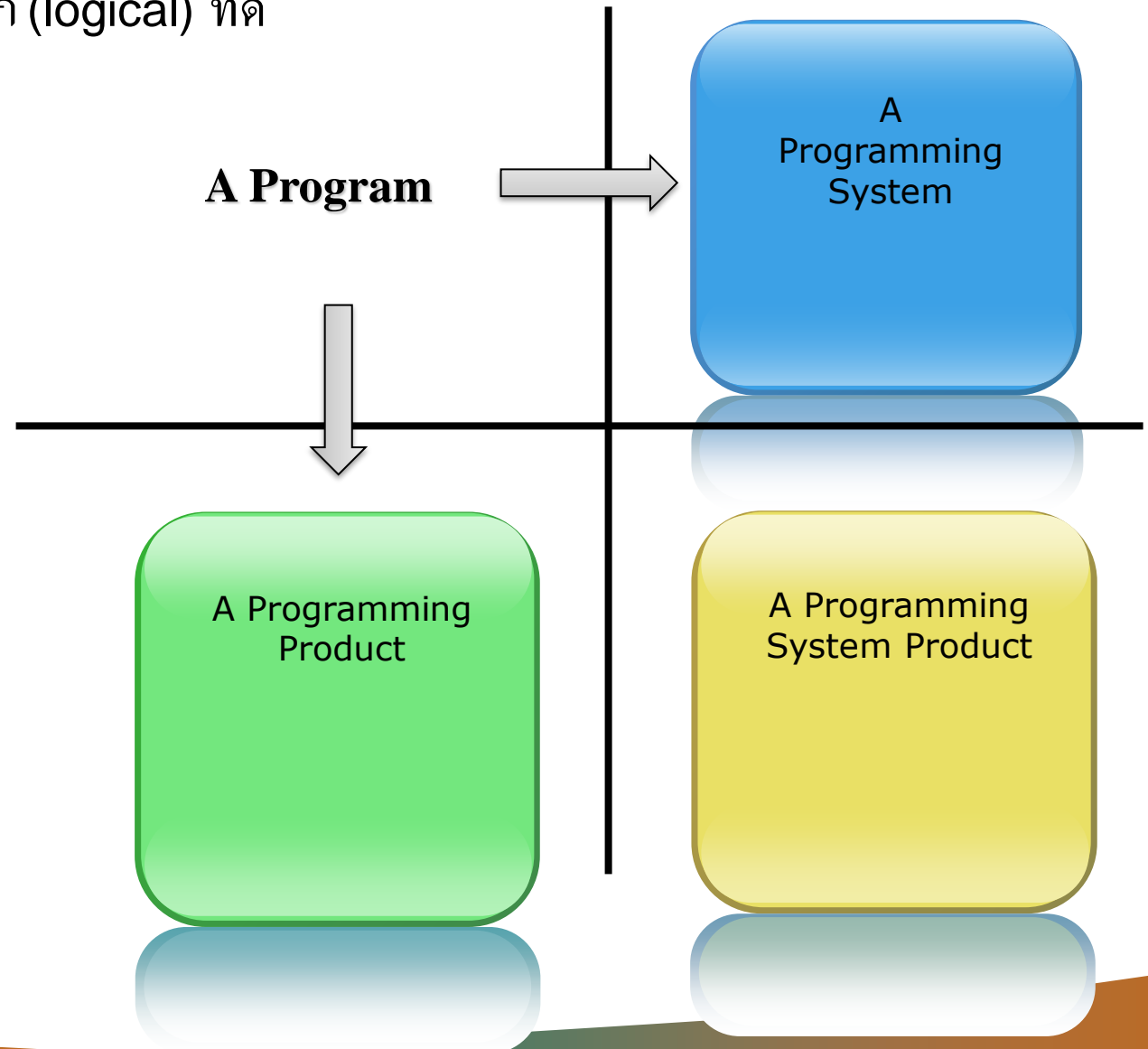
ผศ.ดร.วรรัตน์ สงฆ์แป้น

สาขาวิชาวิทยาการคอมพิวเตอร์

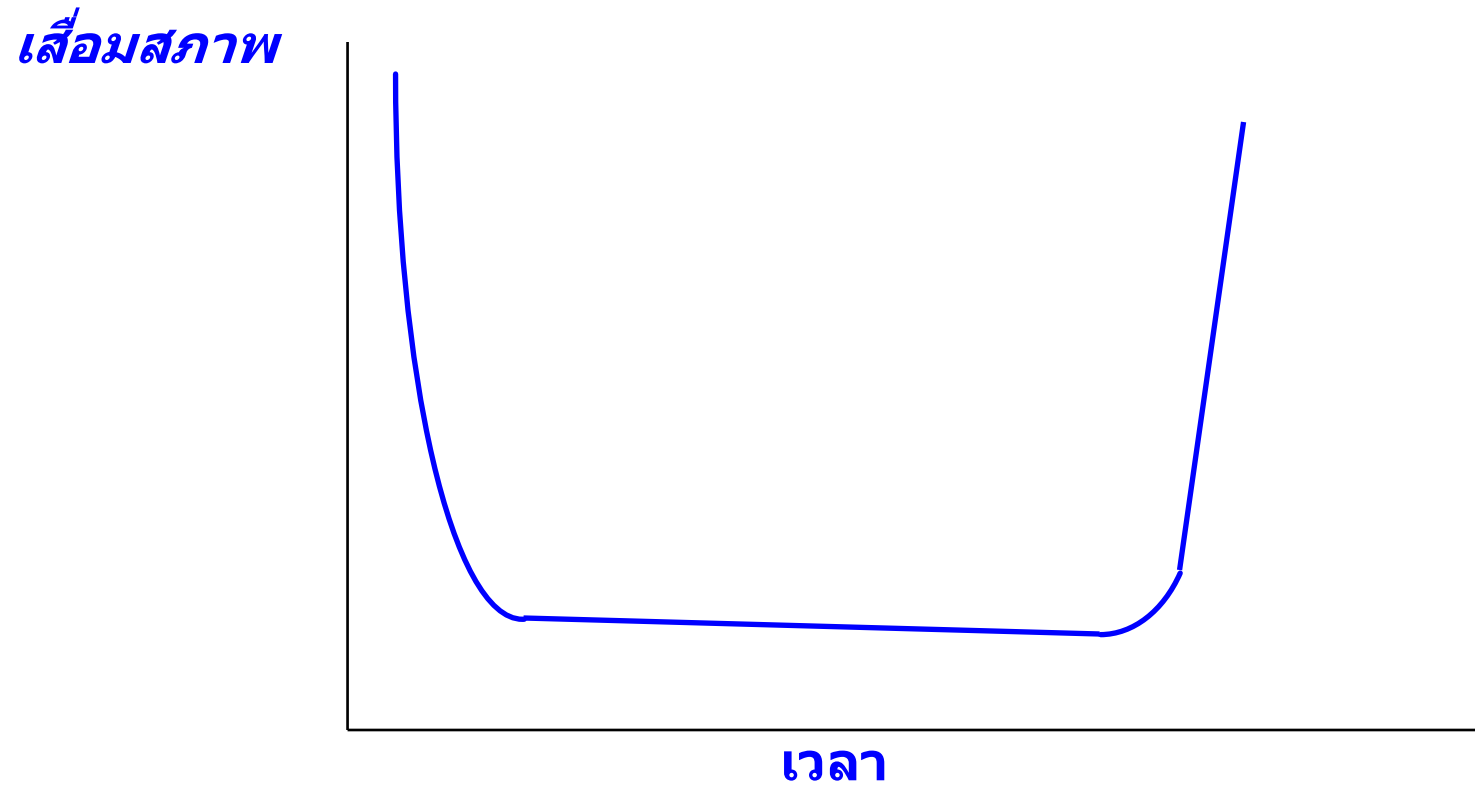
วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น

ซอฟต์แวร์ (Software) คือ อะไร

- ซอฟต์แวร์มีลักษณะเป็นองค์ประกอบทางตรรก (logical) ที่ดี
- ซอฟต์แวร์ต้องผ่านการพัฒนา (developed)
ไม่ใช่การผ่านการผลิต ตามความรู้สึกรั่วๆไป
- ซอฟต์แวร์ไม่ควรมีการเสื่อมสภาพ

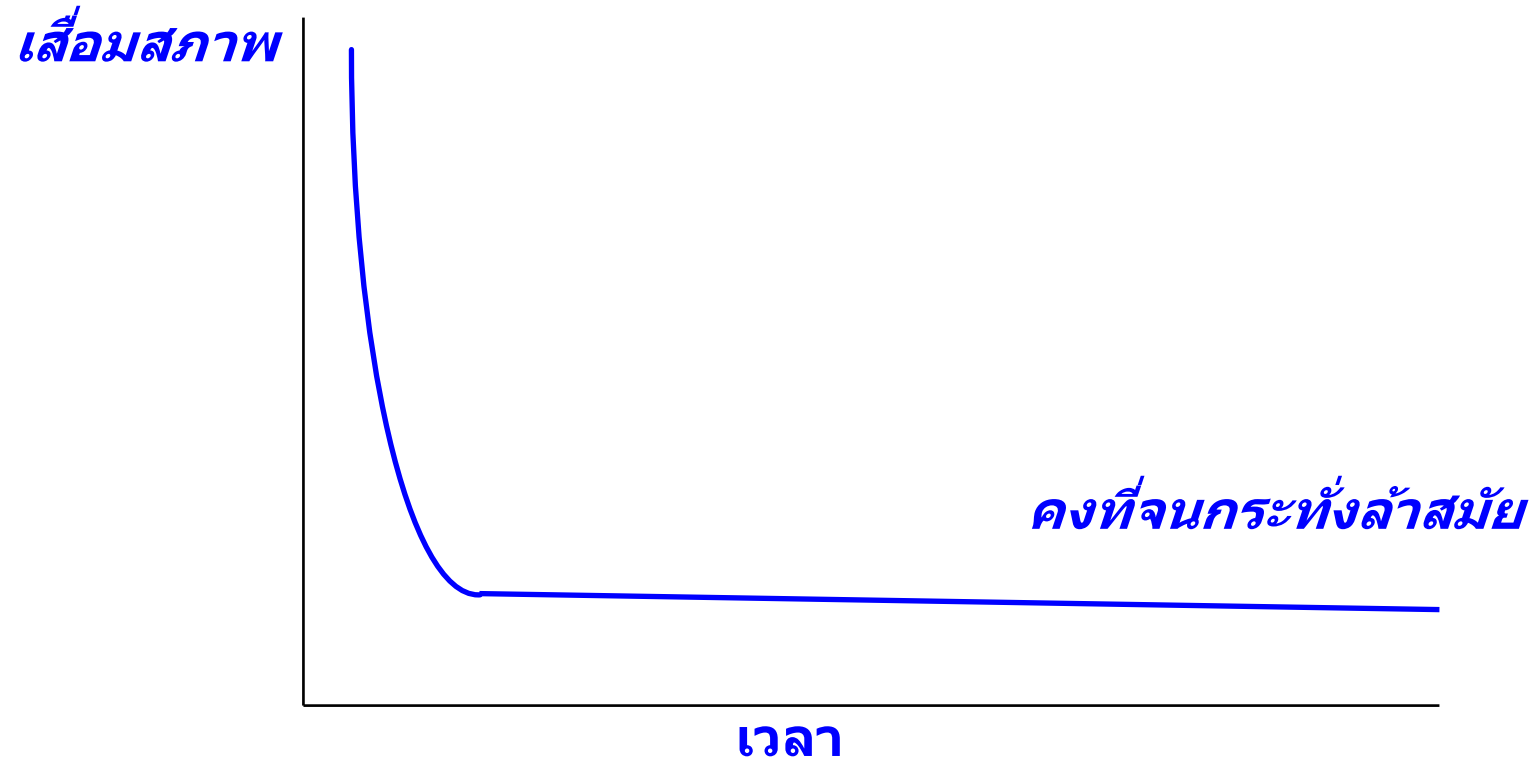


การเสื่อมสภาพของ Hardware



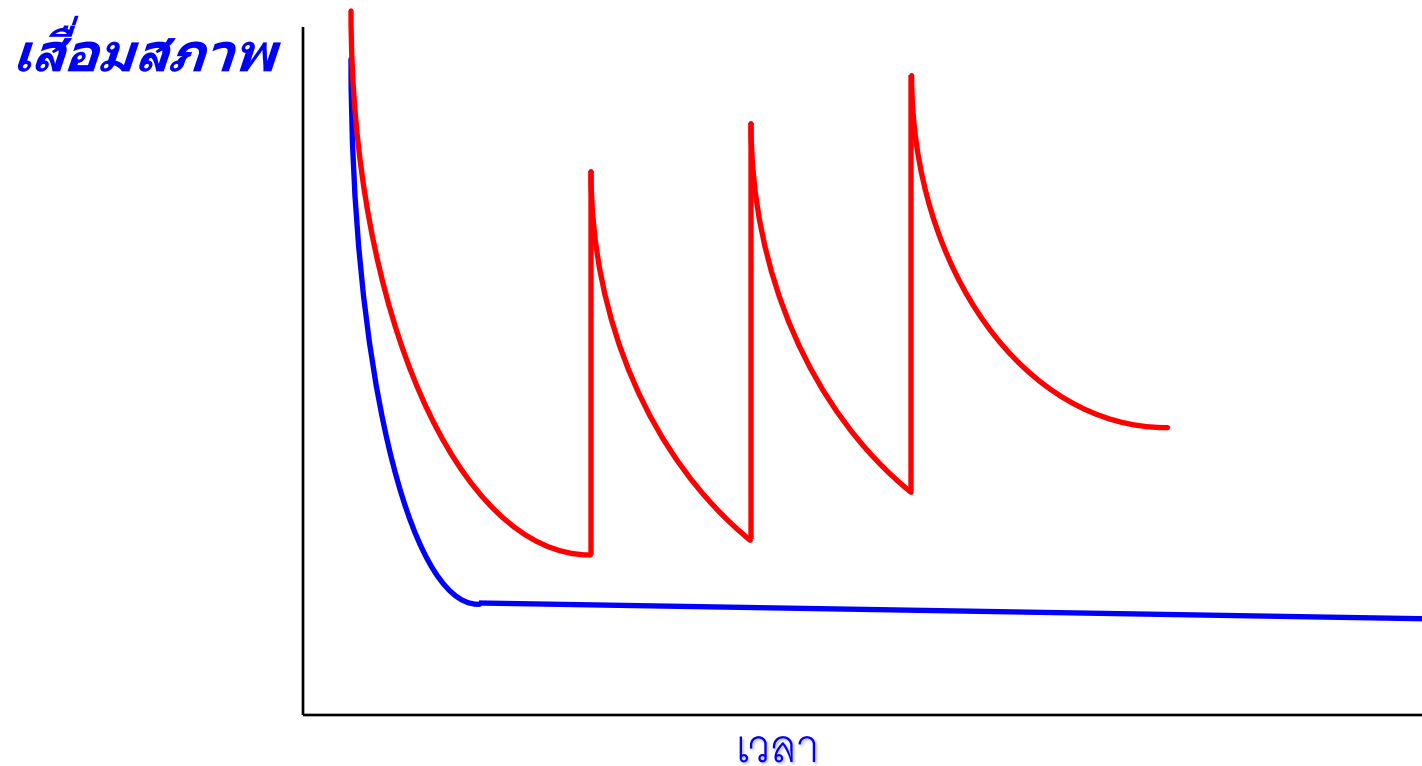
กราฟแสดงความถี่ของความผิดพลาดของ Hardware

การเสื่อมสภาพของ Software (อุดมคติ)



กราฟแสดงความถี่ของความผิดพลาดของ Software (อุดมคติ)

การเสื่อมสภาพของ Software (ความเป็นจริง)



กราฟแสดงความถี่ของความผิดพลาดของ Software (ความเป็นจริง)

ลักษณะของซอฟต์แวร์ที่ดี

- บำรุงรักษาได้ง่าย (Maintainability)
 - ซอฟต์แวร์จะต้องง่ายต่อการบำรุงรักษา สามารถเปลี่ยนแปลง (change) ปรับเปลี่ยนให้เหมาะสม (adaptive) และตอบสนอง (Response) ได้อย่างรวดเร็วและทันทั่วทั้งที่ โดยปราศจากผลกระทบข้างเคียง ในกรณีที่เกิดวิกฤติการณ์ที่ไม่พึงประสงค์
- เชื่อถือได้ (Dependability)
 - ซอฟต์แวร์จะต้องคงไว้ซึ่งความสามารถในการสร้างความน่าเชื่อถือ (Reliability) ได้อย่างมีประสิทธิภาพ โดยจะต้องผ่านการทดสอบและตรวจรับ (Verification and Validation) การทำงานของฟังก์ชันทั้งหมด

ลักษณะของซอฟต์แวร์ที่ดี

- มีประสิทธิภาพ (Efficiency)
 - ซอฟต์แวร์จะต้องก่อให้เกิดความประหยัด หรือสิ้นเปลืองน้อยที่สุด สามารถใช้ทรัพยากรต่างๆ ได้อย่างคุ้มค่าและเหมาะสมในระดับที่ไม่เกินขีดความสามารถของทรัพยากรที่มีอยู่ ไม่ว่าจะเป็นการจัดสรรหน่วยความจำ ขนาดของพื้นที่จัดเก็บ ความรวดเร็วในการประมวลผล
- นำไปใช้งานได้ (Usability)
 - ซอฟต์แวร์จะต้องสะดวก และง่ายต่อการใช้งาน สามารถเสริมสร้างการเรียนรู้ได้อย่างรวดเร็ว ไม่ว่าจะเป็นการออกแบบจอภาพที่นำทางการใช้งานของผู้ใช้ได้ หรือแม้แต่มือถือประกอบการติดตั้งและใช้งานที่เหมาะสม เป็นต้น

ความสำคัญและที่มาของวิศวกรรมซอฟต์แวร์

- ช่องว่างระหว่าง **users** และ **programmers**
- การเปลี่ยนแปลง (**change**) ของเทคโนโลยี เครื่องมือต่างๆ
- การพัฒนาฮาร์ดแวร์ที่มีประสิทธิภาพมาก ทำให้ซอฟต์แวร์ที่ ต้องใช้ มีความซับซ้อนมากตามไปด้วย
- การตอบสนองต่อวิกฤตการณ์ **The responses to the crisis:**

“วิศวกรรมซอฟต์แวร์”

ความหมายของ วิศวกรรมซอฟต์แวร์

- วิศวกรรมซอฟต์แวร์ (Software Engineering) หมายถึง การนำหลักวิชาการด้านวิศวกรรมมาดูแลกระบวนการผลิตซอฟต์แวร์ ตั้งแต่ขั้นแรกจนถึงขั้นตอนบำรุงรักษาหลังการใช้งาน เพื่อให้ซอฟต์แวร์ที่ได้มีคุณภาพสูงสุดภายใต้ข้อจำกัดด้านเวลาและต้นทุน
- ประโยชน์ที่ได้รับ คือ
 - กระบวนการผลิตซอฟต์แวร์ที่มีประสิทธิภาพ เป็นระบบมีมาตรฐานกำหนดวิธีการทำงานอย่างชัดเจน ซึ่งการตรวจสอบคุณภาพของซอฟต์แวร์
 - มีเอกสารหรือกระบวนการควบคุมกำกับการทำงานตลอดจนส่งงาน
 - มีการตรวจสอบและประกันคุณภาพของซอฟต์แวร์ที่ผลิตก่อนส่งถึงมือผู้บริโภค

ลักษณะของวิศวกรรมซอฟต์แวร์

- เกี่ยวข้องกับการสร้างโปรแกรม/ซอฟต์แวร์ ขนาดใหญ่ และซับซ้อน
- สามารถจัดการเกี่ยวกับความซ้ำซ้อนได้
- เน้นการทำงานร่วมกันของบุคลากร
- สามารถเปลี่ยนแปลงได้ง่ายเมื่อจำเป็น
- เน้นการพัฒนาให้มีประสิทธิภาพ
- สอนองความต้องการของผู้ใช้

วิวัฒนาการของวิศวกรรมซอฟต์แวร์

1945-1965

จุดเริ่มต้นของ

วิศวกรรมซอฟต์แวร์

- กล่าวขานและนำมาใช้
จริงครั้งแรกในปลายปี

1950-1960

- องค์การนาโต้ (Nato)

ได้มีการจัดสัมมนา

วิศวกรรมซอฟต์แวร์ในปี

1968

- จุดประกายทำให้เป็นที่
รู้จักกันอย่างแพร่หลาย

1965-1985

วิกฤติซอฟต์แวร์

- มีปัญหาเรื่องของต้นทุน
ค่าใช้จ่าย

- ระยะการดำเนินการ
โครงการไม่เหมาะสม เกิด
ความล้มเหลว

- จำงวิศวกรรมซอฟต์แวร์
มีอาชีพที่สามารถผลิต
ซอฟต์แวร์ให้มีคุณภาพสูง

1985-present

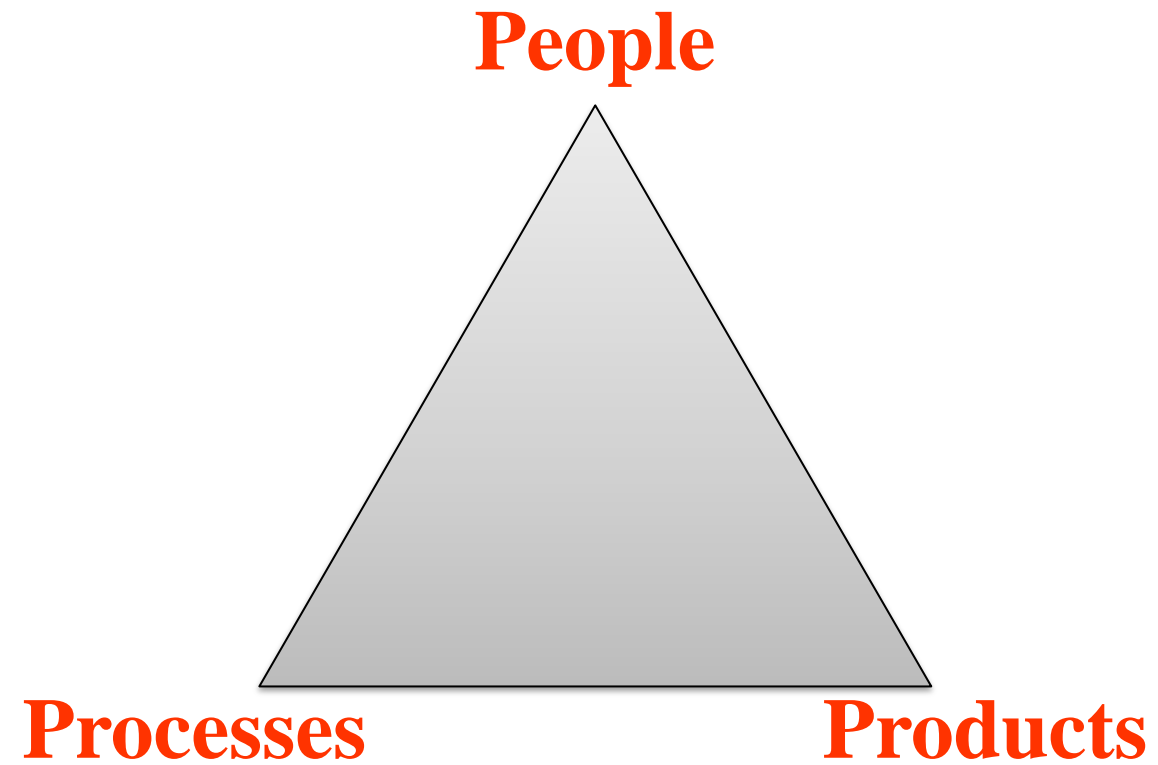
ฟองสบู่แตก

แก้ปัญหาวิกฤติซอฟต์แวร์
อย่างแท้จริง ไม่ว่าจะเป็น

เครื่องมือ เทคนิค และ
กระบวนการต่างๆ ซึ่งนำไปสู่
การพัฒนาซอฟต์แวร์อย่างมี

อาชีพ และถูกแบบแผน
เน้นให้วิศวกรซอฟต์แวร์
คำนึงถึงคุณภาพของซอฟต์แวร์
ด้วยการปรับปรุงกระบวนการ
ผลิตซอฟต์แวร์อย่างสม่ำเสมอ
เพื่อไม่ให้เกิดความผิดพลาด
ต่อไป

Three P's



Three P's

■ People

- education
- skills
- communication
- style
- etc.

■ Product

- requirements
- design
- source code
- executable
- user documentation
- test cases
- test results
- change request
- etc.

■ Process

- planning
- coordination
- management
- measuring
- analyzing
- designing
- coding
- etc.

บุคลากรที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์

- Project Manager
- System Analyst
- Programmer
- Software Engineer

ผู้จัดการโครงการ (Project Manager)

- เป็นผู้ที่ใช้ความรู้ในด้านการบริหารและการจัดการ เพื่อที่จะทำให้โครงการนั้นสำเร็จ
- หน้าที่
 - วางแผนโครงการ
 - จัดการเกี่ยวกับบุคลากร
 - ควบคุมและตรวจสอบการดำเนินงาน

นักวิเคราะห์ระบบ (System Analyst)

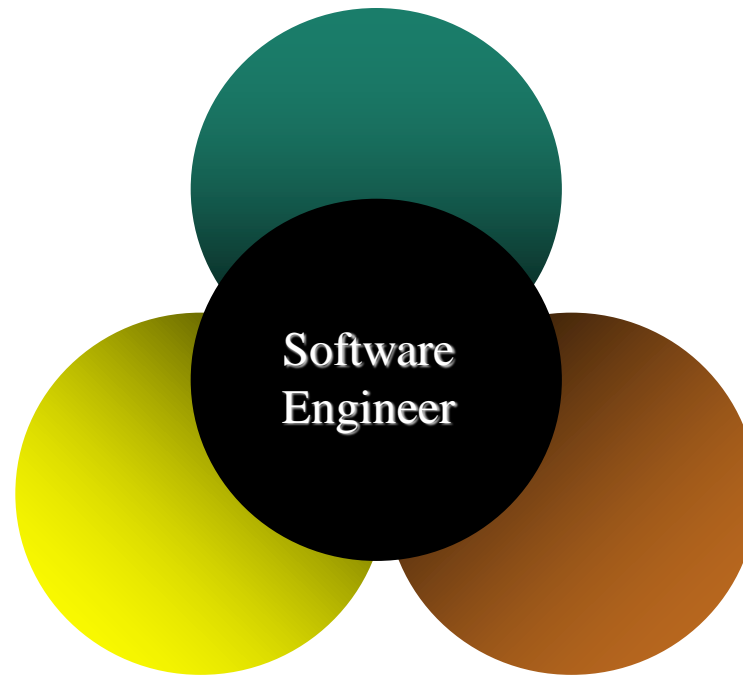
- เป็นผู้วิเคราะห์ความต้องการของลูกค้า และออกแบบระบบเพื่อสนองความต้องการของลูกค้า
- หน้าที่
 - วิเคราะห์ความต้องการ (Requirement Analysis)
 - ออกแบบระบบที่ตอบสนองความต้องการของลูกค้า

นักโปรแกรม (Programmer)

- เป็นผู้พัฒนาโปรแกรม โดยอาศัยความรู้ทางเทคนิค
- หน้าที่
 - ออกแบบ และ เขียนโปรแกรม

วิศวกรซอฟต์แวร์ (Software Engineer)

- เป็นงานที่รวมบทบาทของบุคลากรทั้งสามกลุ่ม



วิศวกรซอฟต์แวร์ (Software Engineer)

- ความรู้ด้านการผลิตซอฟต์แวร์ เกี่ยวกับทักษะ เครื่องมือ และกระบวนการจัดการที่ช่วยสนับสนุนให้การผลิตซอฟต์แวร์ประสบความสำเร็จจนได้
- ความรู้ด้านการบริหารโครงการ เกี่ยวกับการวางแผน การกำหนดกิจกรรม การจัดสรรทรัพยากร การติดตามและควบคุมการดำเนินการต่างๆ ให้เป็นไปตามจุดมุ่งหมายภายใต้ขอบเขตของเวลา ต้นทุนและ คุณภาพ
- ความรู้ด้านการจัดการ เกี่ยวกับกระบวนการนำทรัพยากรการบริหารมาใช้ให้บรรลุวัตถุประสงค์ตามขั้นตอนการบริหาร คือ การวางแผน การจัดองค์กร การชี้นำและการควบคุมงานให้สัมฤทธิ์ผล
- ความรู้ด้านธุรกิจ เกี่ยวกับกิจกรรมต่างๆ ของการซื้อขายสินค้า การผลิตสินค้า หรือการจัดหาบริการตามความต้องการของผู้บริโภค

วิศวกรซอฟต์แวร์ (Software Engineer)

- ความรู้ด้านประชาสัมพันธ์ เกี่ยวกับการติดต่อสื่อสารกับผู้รับข่าวสาร ไม่ว่าจะเป็นลูกค้าหรือผู้ร่วมงานในองค์กร
- ความน่าเชื่อถือ เกี่ยวข้องกับการสร้างความน่าเชื่อถือให้กับลูกค้า ทีมงาน และบุคคลทุกฝ่ายที่เกี่ยวข้องให้ยอมรับในศักยภาพของตน เช่น ความสามารถผลิตซอฟต์แวร์ให้ประสบความสำเร็จตามเป้าหมายที่วางไว้ได้
- ความรู้ลึกไว เกี่ยวกับความรู้ และการคาดคะเนผลกระทบของปัญหาที่จะเกิดขึ้นระหว่างตนกับผู้อื่น
- ความเป็นผู้นำ เกี่ยวกับการใช้อำนาจหน้าที่และจูงใจให้ผู้อื่นปฏิบัติตาม
- ความอดทนต่อภาวะกดดัน เกี่ยวกับความตึงเครียด ซึ่งจะส่งผลกระทบต่อการปฏิบัติโครงการ และเมื่อต้องเผชิญปัญหาต่างๆ ที่ถาโถมเข้ามา ก็สามารถหาวิธีจัดการกับปัญหาเหล่านี้ได้เป็นอย่างดี

วิศวกรซอฟต์แวร์ (Software Engineer)

- ความยืดหยุ่นสูง เกี่ยวกับความสามารถในการปรับตัวให้เข้ากับการเปลี่ยนแปลงของสภาพแวดล้อม โดยอยู่บนพื้นฐานของผลประโยชน์ร่วมกันทุกๆ ฝ่าย
- ความรับผิดชอบสูง เกี่ยวกับการรับมอบหมายงานหรือหน้าที่ ซึ่งจะต้องบริหาร โครงการผลิตซอฟต์แวร์ให้ประสบความสำเร็จตามเจตนารมณ์ขององค์กร
- ความยุติธรรม เกี่ยวกับความถูกต้อง และจริยธรรมอันดีงาม ปราศจากการเลือกปฏิบัติหรือเอนเอียงไปด้านใดด้านหนึ่งโดยขาดเหตุผลอันสมควร

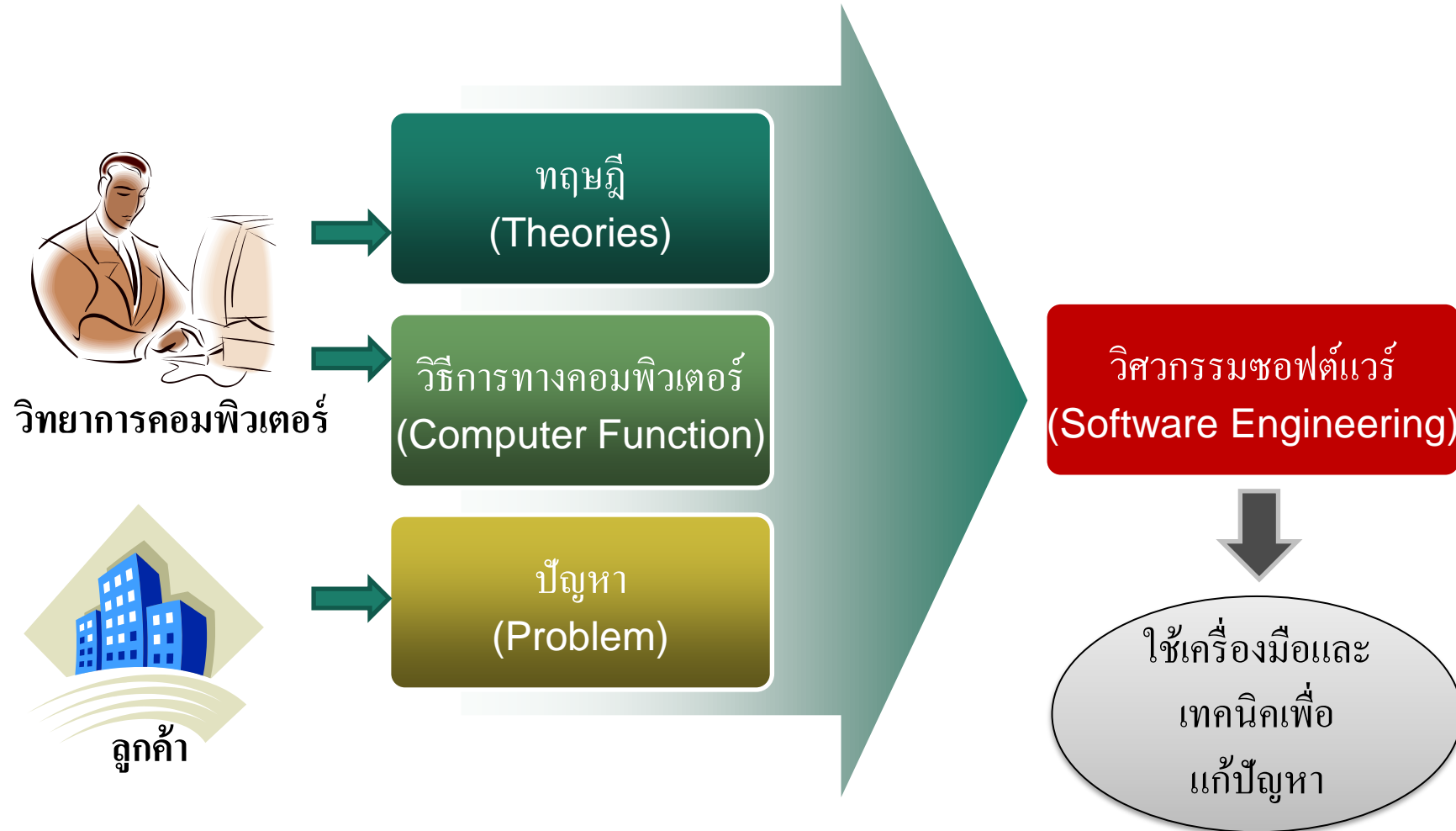
องค์ประกอบของวิศวกรรมซอฟต์แวร์

- ส่วนที่ 1 วิศวกรรมระบบ (System Engineering) หมายถึง กระบวนการศึกษาและวิเคราะห์ของระบบที่มีความสลับซับซ้อน
- ส่วนที่ 2 วิศวกรรมการผลิต (Development Engineering) ซึ่งเป็นกระบวนการแปรสภาพความต้องการของระบบ (System Requirements) ให้กลายเป็นซอฟต์แวร์อันเป็นเป้าหมายสำคัญทางด้านวิศวกรรมซอฟต์แวร์

ความแตกต่างระหว่าง

- **วิทยาการคอมพิวเตอร์ (Computer science)**
 - เป็นหลักวิชาที่ว่าด้วยทฤษฎีและวิธีการทางคอมพิวเตอร์และระบบซอฟต์แวร์ในเชิงลึก
- **วิศวกรรมซอฟต์แวร์ (Software Engineering)**
 - เป็นหลักวิชาที่ว่าด้วยปัญหาในทางปฏิบัติของการผลิตซอฟต์แวร์ ไม่ว่าจะเป็นการบริหารจัดการทรัพยากร การออกแบบระบบหรือซอฟต์แวร์ การเลือกเทคโนโลยี ผลกระทบของสภาพแวดล้อมที่มีต่อระบบ การประมาณการด้านต่างๆ การทดสอบ การประกันคุณภาพและการบำรุงรักษาระบบ

ความสัมพันธ์ระหว่างวิทยาการคอมพิวเตอร์และวิศวกรรมซอฟต์แวร์



ความแตกต่างจากวิศวกรรมระบบ

- **วิศวกรรมระบบ (System Engineering)** เกี่ยวข้องกับทุกๆ ด้านของการพัฒนาและการเปลี่ยนแปลงของระบบที่มีความซับซ้อน โดยมีซอฟต์แวร์เป็นแกนหลักในการทำงานของระบบ
- **วิศวกรรมระบบ** จึงเกี่ยวข้องกับการพัฒนาฮาร์ดแวร์ นโยบาย ออกแบบ พัฒนาระบบงานและระบบไปพร้อมๆ กับการวิศวกรรมซอฟต์แวร์
- **วิศวกรรมระบบ** จะก่อให้เกิดการกำหนดระบบ ระบุถึงสถาปัตยกรรมทั้งระบบ แล้วนำส่วนประกอบที่แตกต่างกันมาประสานเข้าด้วยกันจนกลายเป็น 1 ระบบ

ความแตกต่างจากการวิเคราะห์และออกแบบระบบ

- **การวิเคราะห์และออกแบบระบบ (System Analysis and Design)** เป็นการศึกษา วิเคราะห์ และแยกแยะปัญหาที่เกิดขึ้นในระบบ แล้วทำการออกแบบ และกำหนดคุณสมบัติทางเทคนิค โดยนำระบบคอมพิวเตอร์มาประยุกต์ใช้เพื่อแก้ปัญหาที่ได้ทำการวิเคราะห์มาแล้ว
- **นักวิเคราะห์ระบบ (System Analyst)** เป็นผู้รับผิดชอบงานวิเคราะห์และออกแบบโดยตรง
- **วิศวกรรมซอฟต์แวร์** จะทำหน้าที่เกี่ยวกับการผลิตซอฟต์แวร์เพื่อการค้า แม้ว่ากระบวนการที่นำมาใช้พัฒนาซอฟต์แวร์หรือระบบจะคล้ายคลึงกันก็ตาม แต่ขั้นตอนการวิเคราะห์ระบบเป็นเพียงกิจกรรมหนึ่งในวิศวกรรมซอฟต์แวร์เท่านั้น

สรุป

- **วิศวกรรมซอฟต์แวร์** เกิดจากการเปลี่ยนแปลง หลังยุคปฏิวัติอุตสาหกรรม เพื่อนำมาประยุกต์ใช้กับกระบวนการผลิตซอฟต์แวร์ เพื่อต้องการให้ผลิตภัณฑ์ซอฟต์แวร์มีคุณภาพสูงสุด
- ความหมายของ **“วิศวกรรมซอฟต์แวร์”** คือ การนำหลักวิชาการด้านวิศวกรรมมาดูแลกระบวนการผลิตซอฟต์แวร์ ตั้งขึ้นตอนแรกจนถึงการบำรุงรักษา เพื่อให้ซอฟต์แวร์มีคุณภาพสูงสุดให้ตามคุณสมบัติซอฟต์แวร์ที่ดี ด้วยข้อจำกัดด้านเวลาและต้นทุน
- **วิศวกรรมซอฟต์แวร์** มีองค์ประกอบ 2 ส่วน คือ **วิศวกรรมระบบและวิศวกรรมการผลิต**