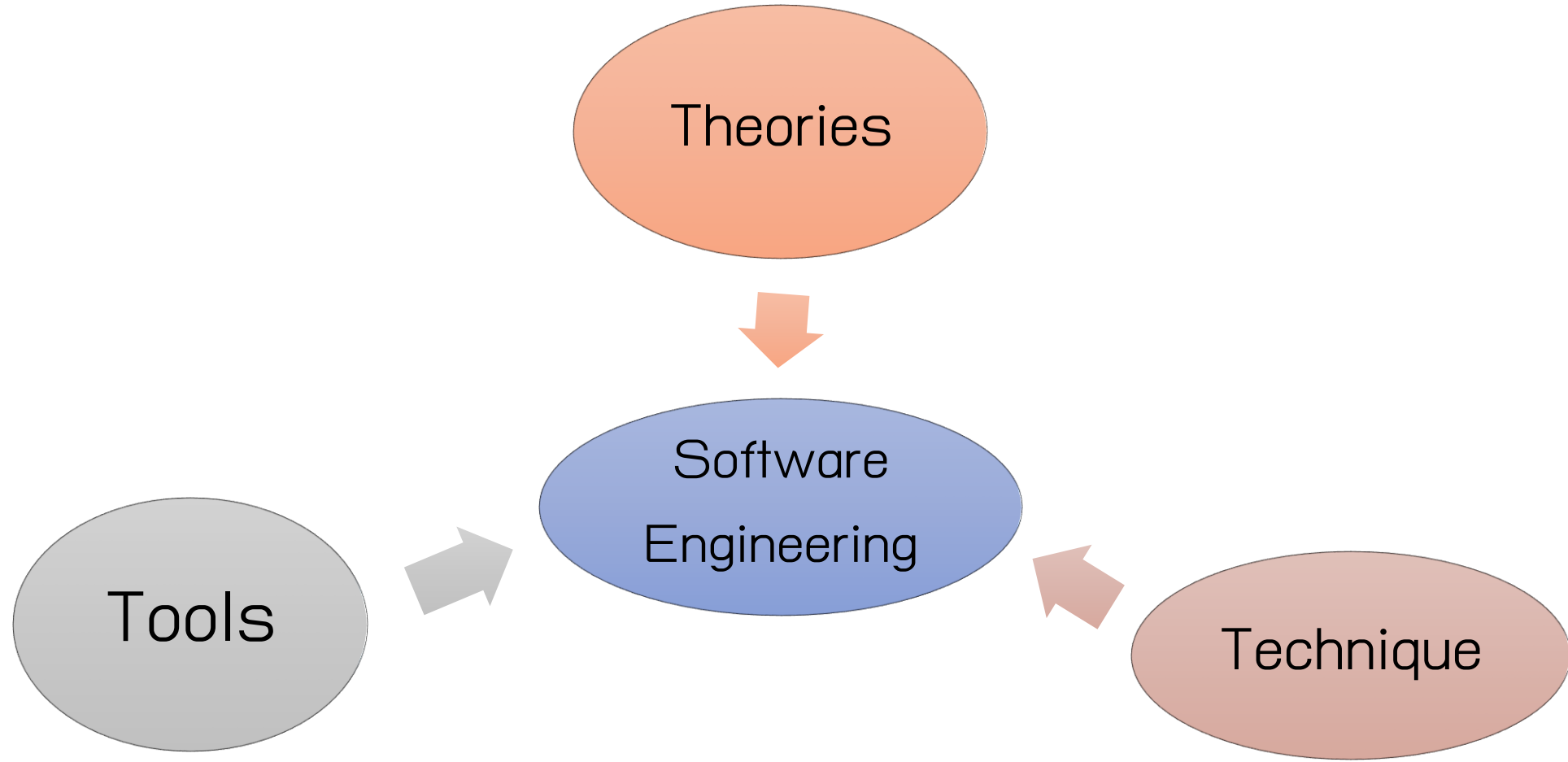


หน่วยที่ 1

แนะนำวิชาวิศวกรรมซอฟต์แวร์

ความหมาย หลักการและความสำคัญของ วิศวกรรมซอฟต์แวร์

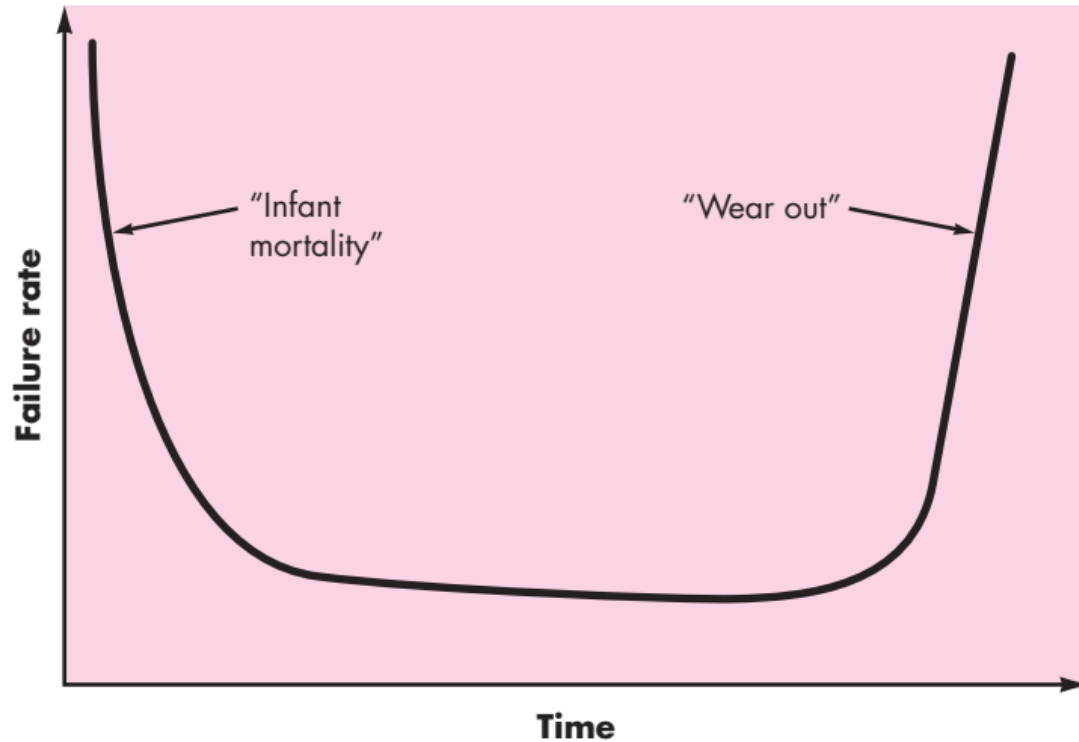
วิศวกรรมซอฟต์แวร์



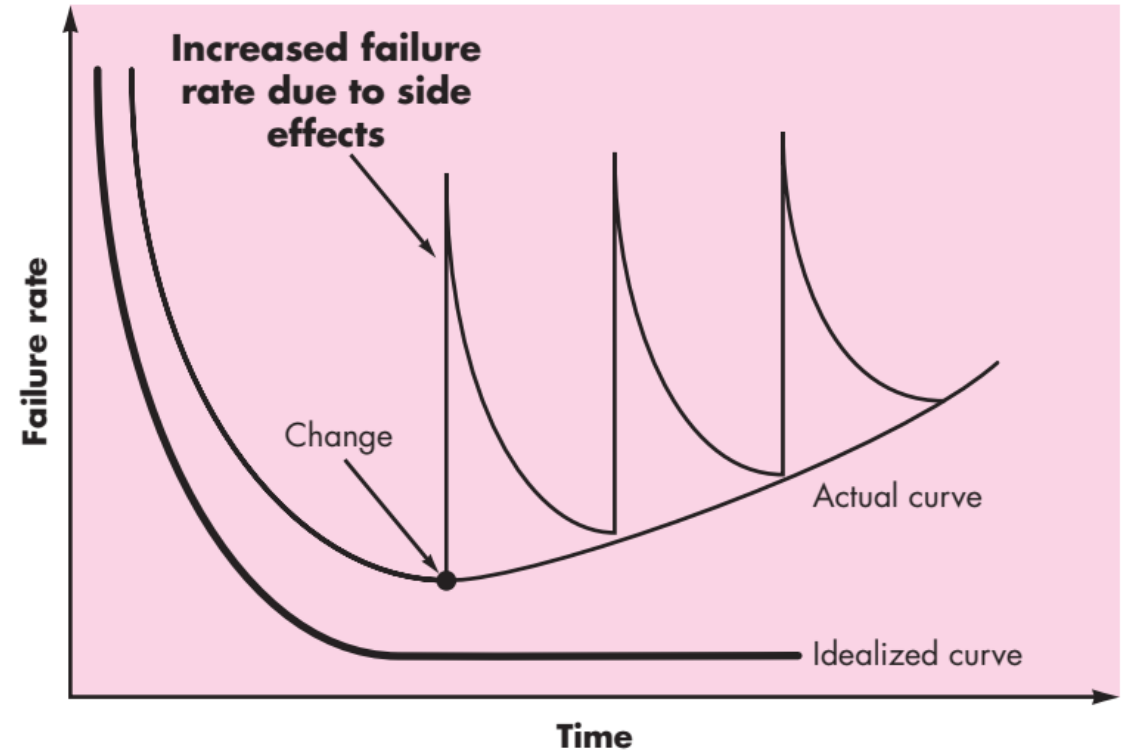
Software engineering Fact!!!

- ระบบเศรษฐกิจของประเทศที่พัฒนาแล้ว ล้วนแต่พึ่งพา Software
- ทุกระบบในโลก ต่างก็มี software controlled เพิ่มมากขึ้น มากขึ้นเรื่อย ๆ
- ราคาของ software มักจะเป็นตัวกำหนดราคาของระบบคอมพิวเตอร์
- ราคาของการ maintain มักจะสูงกว่าราคาในการ developed
 - ในระบบที่ใช้งานเป็นระยะเวลายาวนาน ค่าบำรุงรักษาจะสูงกว่าค่าสร้างซอฟต์แวร์
- Software engineering เน้นการพัฒนาซอฟต์แวร์ที่คุ้มค่า (cost-effective)

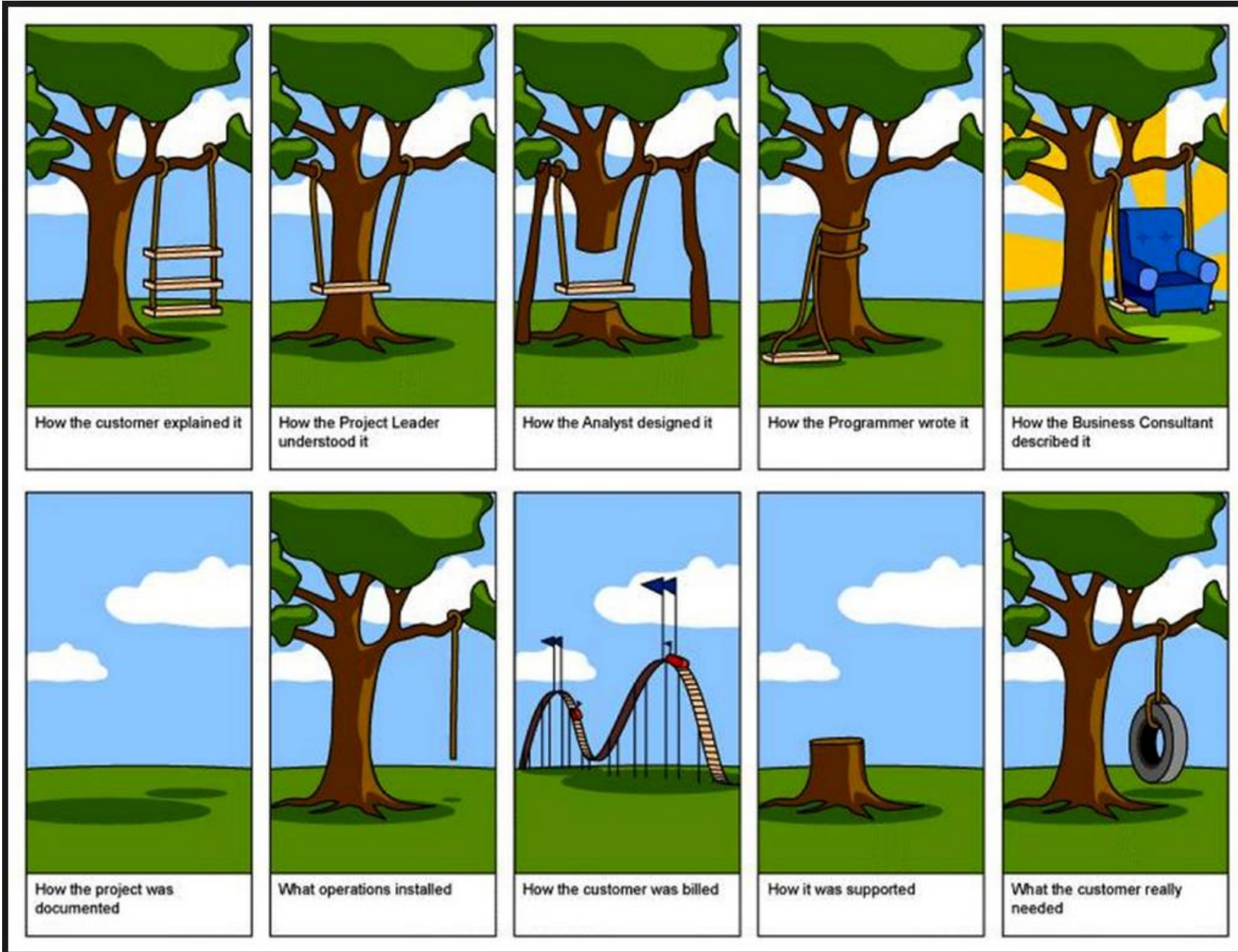
Hardware vs Software – Failure curve



Hardware



Software



Professional software development

ซอฟต์แวร์คืออะไร?

Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.

Software คือ โปรแกรมคอมพิวเตอร์รวมถึงเอกสารที่เกี่ยวข้อง

- ซอฟต์แวร์อาจจะถูกสร้างขึ้นตามความต้องการของผู้ใช้เฉพาะราย
- หรือสร้างขึ้นเพื่อวางจำหน่ายโดยทั่วไป
 - โปรแกรมสำเร็จรูปประมวลผลคำ
 - ระบบปฏิบัติการ

คุณสมบัติของซอฟต์แวร์ที่ดี

A Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.

ซอฟต์แวร์ที่ดี

- ควรประกอบด้วยฟังก์ชันการทำงานที่ผู้ใช้ต้องการ
- มีประสิทธิภาพในการทำงาน
- สามารถบำรุงรักษาง่าย
- เชื่อถือได้
- มีประโยชน์

วิศวกรรมซอฟต์แวร์คืออะไร?

Software engineering is an engineering discipline that is concerned with all aspects of software production.

วิศวกรรมซอฟต์แวร์เป็นสาขาวิศวกรรมที่เกี่ยวข้องกับ**ทุกมิติ**ของการผลิตซอฟต์แวร์

- Requirements
- Design
- Construction
- Verification and validation
- Evolution
- Reliability
- Security

สาระสำคัญของวิศวกรรมซอฟต์แวร์ประกอบด้วยอะไรบ้าง?

Software specification, software development, software validation and software evolution.

- การออกข้อกำหนดของซอฟต์แวร์
- การพัฒนาซอฟต์แวร์
- การตรวจสอบซอฟต์แวร์
- การพัฒนา/ปรับปรุงซอฟต์แวร์

ข้อแตกต่างระหว่าง software engineering และ computer science

Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

วิทยาการคอมพิวเตอร์ มุ่งเน้นไปที่ทฤษฎีและพื้นฐาน

วิศวกรรมซอฟต์แวร์ เกี่ยวข้องกับการปฏิบัติในการพัฒนาและนำเสนอซอฟต์แวร์ที่เป็นประโยชน์

ข้อแตกต่างระหว่าง software engineering และ system engineering

System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

วิศวกรรมระบบเกี่ยวข้องกับทุกด้านของการพัฒนาระบบคอมพิวเตอร์ (ซึ่งรวมถึงฮาร์ดแวร์ ซอฟต์แวร์ และกระบวนการทางวิศวกรรม) ในขณะที่วิศวกรรมซอฟต์แวร์เป็นเพียงส่วนหนึ่งของกระบวนการทั่วไปนี้

วิศวกรรมซอฟต์แวร์ เป็น subset ของ วิศวกรรมระบบ

ความท้าทายของวิศวกรรมซอฟต์แวร์ มีอะไรบ้าง?

Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.

การพัฒนาซอฟต์แวร์สมัยใหม่ ต้องเผชิญกับ

ความต้องการที่หลากหลาย ที่เพิ่มขึ้นเรื่อยๆ

ระยะเวลา ในการส่งมอบที่ลดลง

การพัฒนาซอฟต์แวร์ที่มี **ความน่าเชื่อถือ** มากขึ้น

ต้นทุนของวิศวกรรมซอฟต์แวร์มีอะไรบ้าง

Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.

ค่าใช้จ่ายต้นทุนการพัฒนา ประมาณ 60%

ค่าใช้จ่ายในการทดสอบ ประมาณ 40%

สำหรับซอฟต์แวร์ที่กำหนดเอง ค่าใช้จ่ายในการบำรุงรักษามักจะสูงกว่าต้นทุนการพัฒนา

วิธีการหรือเทคนิคที่ดีที่สุดสำหรับวิศวกรรมซอฟต์แวร์

While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.

- เทคนิคต่าง ๆ จะเหมาะสมกับระบบต่าง ๆ กันไป
 - เกม อาจจะพัฒนาโดยใช้ชุดของต้นแบบ
 - ระบบควบคุม เน้นความปลอดภัย ต้องมีข้อกำหนดที่ครบถ้วนและสามารถวิเคราะห์ได้
- เราไม่สามารถบอกได้ว่าวิธีหนึ่งดีกว่าอีกวิธีหนึ่ง

อิทธิพลของเทคโนโลยีอินเทอร์เน็ตที่มีต่อวิศวกรรมซอฟต์แวร์

The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

นับตั้งแต่มีอินเทอร์เน็ต ก็ทำให้ผู้คนเข้าถึงบริการด้านซอฟต์แวร์มากขึ้น และเพิ่มโอกาสในการพัฒนาระบบ distributed service-based

การพัฒนาระบบบนเว็บได้นำไปสู่ความก้าวหน้าที่สำคัญในการเขียนโปรแกรมภาษาและการนำซอฟต์แวร์มาใช้ใหม่

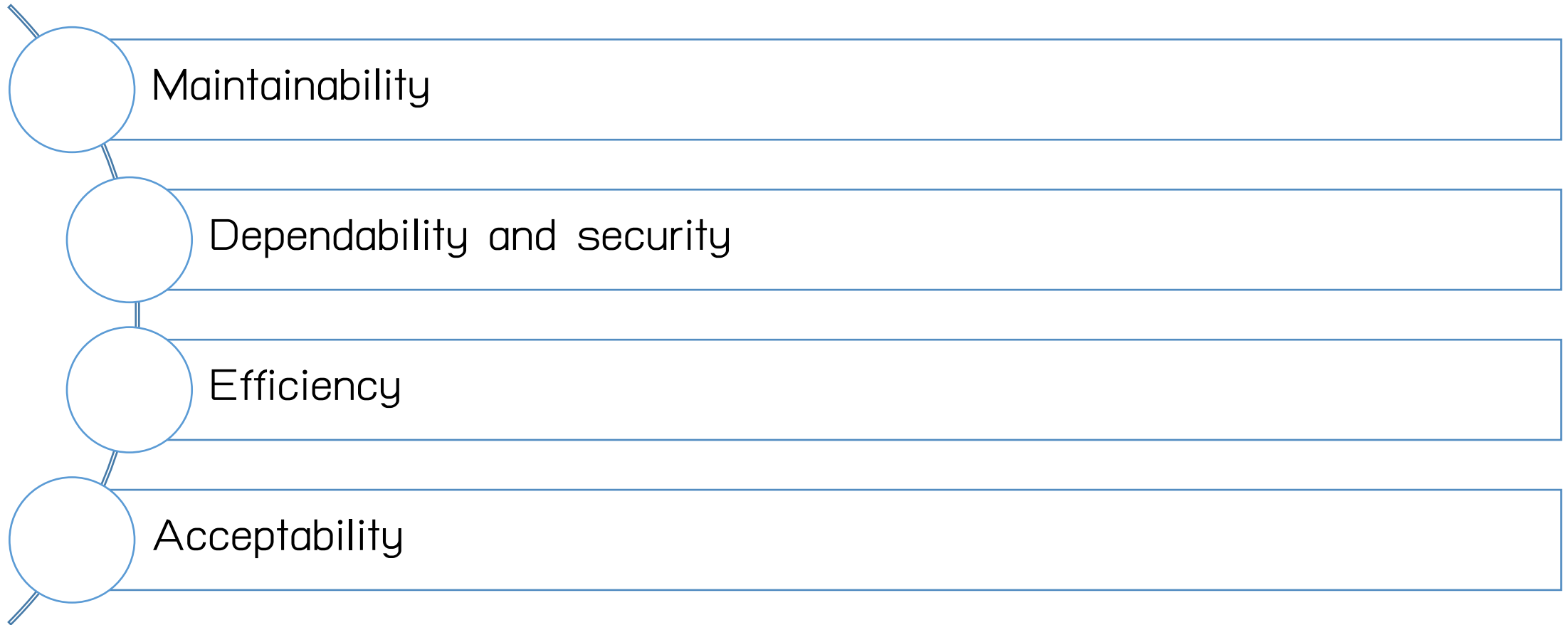
ผลิตภัณฑ์ซอฟต์แวร์

- ผลิตภัณฑ์ทั่วไป (Generic products)
 - เป็นซอฟต์แวร์สำหรับใช้งานกับระบบอิสระ (Stand-alone systems) ที่มีการวางจำหน่ายแก่บุคคลทั่วไป
เช่น ซอฟต์แวร์สำหรับ PC software ได้แก่ graphics programs, project management tools; CAD software; หรือซอฟต์แวร์สำหรับตลาดเฉพาะด้านเช่นโปรแกรมบริหารจัดการโรงพยาบาล
- ผลิตภัณฑ์สร้างขึ้นโดยเฉพาะ (Customized products)
 - เป็นซอฟต์แวร์ที่สร้างขึ้นตามความต้องการเฉพาะอย่าง.
เช่น ซอฟต์แวร์สำหรับระบบสมองกลฝังตัว ระบบไฟจราจร ระบบอาณัติสัญญาณของรถไฟฟ้า ระบบควบคุมชิปนาฬิกา เป็นต้น

ข้อกำหนดสำหรับซอฟต์แวร์ (Product specification)

- ซอฟต์แวร์ทั่วไป (Generic products)
 - ข้อกำหนด จะถูกออกแบบโดย ผู้พัฒนาซอฟต์แวร์
 - การตัดสินใจว่าจะ ปรับปรุง/เปลี่ยนแปลง ซอฟต์แวร์ขึ้นอยู่กับ ผู้พัฒนาซอฟต์แวร์
- ซอฟต์แวร์เฉพาะงาน (Customized products)
 - ข้อกำหนด จะถูกออกแบบโดย ผู้ใช้ซอฟต์แวร์
 - การตัดสินใจว่าจะ ปรับปรุง/เปลี่ยนแปลง ซอฟต์แวร์ขึ้นอยู่กับ ผู้ใช้ซอฟต์แวร์

Essential attributes of good software



Product characteristic	Description
Maintainability	<ul style="list-style-type: none"> ซอฟต์แวร์ควรได้รับการเขียนในลักษณะที่สามารถพัฒนาเพื่อตอบสนองความต้องการที่เปลี่ยนแปลงไปของลูกค้า เมื่อสภาพแวดล้อมทางธุรกิจเปลี่ยนแปลงไป การเปลี่ยนแปลงซอฟต์แวร์จะเป็นเรื่องที่ไม่สามารถหลีกเลี่ยงได้
Dependability and security	<ul style="list-style-type: none"> ความน่าเชื่อถือของซอฟต์แวร์ประกอบด้วยลักษณะเฉพาะหลายอย่าง เช่น ความมั่นคงและความปลอดภัย ซอฟต์แวร์ที่น่าเชื่อถือ <u>ต้องทำให้เกิดความเสียหาย</u> ไม่ว่าจะเป็นทางกายภาพหรือทางเศรษฐกิจ (ถึงแม้ระบบจะล้มเหลวก็ตาม) ผู้ใช้ที่อาจสร้างอันตรายต่อระบบ จะต้องไม่สามารถเข้าถึงหรือทำลายระบบได้

Product characteristic	Description
Efficiency	<ul style="list-style-type: none"> ซอฟต์แวร์ไม่ควรใช้ทรัพยากรระบบอย่างสิ้นเปลือง ไม่ว่าจะเป็นหน่วยความจำหรือโปรเซสเซอร์ ประสิทธิภาพรวมถึงเวลาในการตอบสนอง เวลาการประมวลผล การใช้หน่วยความจำ ฯลฯ
Acceptability	<ul style="list-style-type: none"> ซอฟต์แวร์ต้องเป็นที่ยอมรับจากผู้ใช้งานประเภทต่าง ๆ ตามที่ได้ออกแบบไว้ หมายความว่าต้องเข้าใจง่าย ใช้งานได้ และสามารถทำงานร่วมกับระบบเดิมๆ ที่ผู้ใช้ใช้งานอยู่ได้

Software engineering

- วิศวกรรมซอฟต์แวร์เป็นสาขาวิศวกรรมที่เกี่ยวข้องกับทุกแง่มุมในการผลิตซอฟต์แวร์ตั้งแต่ขั้นเริ่มต้นของข้อกำหนดระบบจนถึงการบำรุงรักษาระบบหลังจากที่ได้เริ่มใช้งานแล้ว
- แง่มุมทางด้านวิศวกรรม
 - การใช้ทฤษฎีและวิธีการที่เหมาะสมในการแก้ปัญหาที่เกิดขึ้นกับข้อจำกัดภายในองค์กรและการเงิน
- แง่มุมในการผลิตซอฟต์แวร์
 - ไม่ใช่เพียงแค่กระบวนการทางเทคนิคในการพัฒนาเท่านั้น ยังรวมถึงการบริหารโครงการ การพัฒนาเครื่องมือ และวิธีการอื่น ๆ เพื่อสนับสนุนการผลิตซอฟต์แวร์

Importance of software engineering

- บุคคลและสังคม ต้องพึ่งพาระบบซอฟต์แวร์ขั้นสูงมากขึ้นเรื่อย ๆ
 - เราจำเป็นต้องผลิตระบบ (รวมถึงซอฟต์แวร์ที่ใช้ควบคุม) ที่น่าเชื่อถือและวางใจได้ ได้อย่างประหยัดและรวดเร็ว
- การใช้วิศวกรรมซอฟต์แวร์ จะช่วยให้เกิดความคุ้มค่าในระยะยาว (มากกว่าคิดไปทำไป)
 - หากเราใช้วิธีการด้านวิศวกรรมซอฟต์แวร์สำหรับผลิตซอฟต์แวร์ แทนการเขียนโปรแกรมแบบเก่า ๆ ก็จะมีความคุ้มค่าต่อทุกฝ่ายที่เกี่ยวข้อง
 - ค่าใช้จ่ายส่วนใหญ่ที่เกิดขึ้น มักจะเป็นค่าใช้จ่ายในการเปลี่ยนแปลงซอฟต์แวร์หลังจากที่ใช้งานแล้ว
 - หากขั้นตอนการพัฒนาเป็นไปตามหลักวิศวกรรม จะช่วยลดความยุ่งยากในการเปลี่ยนแปลงซอฟต์แวร์หลังจากที่ส่งมอบไปแล้ว

ความหลากหลายทาง Software engineering

- ระบบซอฟต์แวร์มีหลายประเภท และไม่มีเทคนิคซอฟต์แวร์อเนกประสงค์ (universal) หรือเทคนิคสำเร็จรูปที่สามารถใช้ได้กับการสร้างซอฟต์แวร์ได้ทุก ๆ รูปแบบ
- วิธีการและเครื่องมือด้านวิศวกรรมซอฟต์แวร์ ขึ้นอยู่กับประเภทของแอปพลิเคชันที่พัฒนาขึ้น
 - แตกต่างกันไปตามความต้องการของลูกค้า และพื้นฐานของทีมพัฒนา

ประเภทของโปรแกรมประยุกต์ (application types)

- Stand-alone applications

- เป็นระบบ application ที่รันบนเครื่องคอมพิวเตอร์ (เช่นพีซี) ประกอบด้วยฟังก์ชันการทำงานที่จำเป็นทั้งหมดและไม่จำเป็นต้องเชื่อมต่อกับเครือข่าย

- Interactive transaction-based applications

- Application ที่รันบนคอมพิวเตอร์ระยะไกลและเข้าถึงได้โดยผู้ใช้จากเครื่องพีซีหรือ terminal ของตนเอง ซึ่งรวมถึงweb application

ประเภทของโปรแกรมประยุกต์ (application types)

- Embedded control systems

- เป็นซอฟต์แวร์ที่ควบคุมและจัดการอุปกรณ์ฮาร์ดแวร์แบบฝังตัว ในโลกนี้มีระบบฝังตัวมากกว่าระบบอื่น ๆ

- Batch processing systems

- เป็นระบบที่ออกแบบมาเพื่อประมวลผลข้อมูลเป็นกลุ่มใหญ่ สามารถนำข้อมูลจำนวนมากมาประมวลผลเพื่อสร้างผลลัพธ์ที่ต้องการ เป็นครั้งๆ ไป

ประเภทของโปรแกรมประยุกต์ (application types)

- Entertainment systems

- เป็นระบบที่เน้นใช้งานส่วนบุคคล และถูกออกแบบมาเพื่อความบันเทิงแก่ผู้ใช้เป็นหลัก

- Systems for modelling and simulation

- เป็นระบบที่พัฒนาโดยนักวิทยาศาสตร์และวิศวกรในการจำลองกระบวนการหรือสถานการณ์ทางกายภาพซึ่งรวมถึงแบบจำลองของวัตถุหลายสิ่งที่แตกต่างกันแต่ทำงานประสานกัน

ประเภทของโปรแกรมประยุกต์ (application types)

- Data collection systems

- เป็นระบบที่รวบรวมข้อมูลจากสภาพแวดล้อมโดยใช้ชุดเซ็นเซอร์และส่งข้อมูลเหล่านั้นไปยังระบบอื่น ๆ เพื่อการประมวลผล

- Systems of systems

- เป็นระบบที่ประกอบด้วยระบบซอฟต์แวร์อื่น ๆ จำนวนมาก

Software engineering fundamentals

ถึงแม้ประเภทของซอฟต์แวร์จะมีหลากหลาย แต่หลักการพื้นฐานบางอย่าง ก็สามารถใช้ในการสร้างซอฟต์แวร์ทุกประเภทโดยไม่คำนึงถึงเทคนิคการพัฒนาที่ใช้ เช่น

- ความเข้าใจในระบบและกระบวนการพัฒนาอย่างถ่องแท้
- ความเชื่อถือได้และประสิทธิภาพ
- ทำความเข้าใจและการจัดการข้อกำหนดและความต้องการของซอฟต์แวร์ (สิ่งที่ซอฟต์แวร์ควรทำ) ให้รอบคอบ เนื่องจากมีความสำคัญเป็นลำดับต้นๆ
- ถ้าเป็นไปได้ (และเหมาะสม) ควรใช้ซอฟต์แวร์ที่พัฒนาแล้วแทนที่จะเขียนซอฟต์แวร์ใหม่

Internet software engineering

- ปัจจุบันนี้เว็บเป็นแพลตฟอร์มสำหรับการรัน application ที่ได้รับความนิยมมากขึ้นเรื่อย ๆ
 - มีการพัฒนาระบบบนเว็บมากกว่าระบบ stand alone
 - บริการเว็บ (web services) ช่วยให้สามารถเข้าถึง application ได้ผ่านทางเว็บ
 - บริการเว็บช่วยให้เข้าถึงข้อมูลได้หลากหลาย ทั้งที่สดใหม่หรือเก็บถาวร
 - Cloud computing เป็นแนวทางในการให้บริการคอมพิวเตอร์ซึ่ง application ทำงานจากระยะไกลบน 'cloud'
 - ผู้ใช้ไม่ได้ซื้อซอฟต์แวร์ (หรือแม้แต่ฮาร์ดแวร์) แต่เป็นระบบที่จ่ายตามการใช้งาน

Web software engineering

- Software reuse

- การใช้ซอฟต์แวร์ซ้ำเป็นวิธีการที่สำคัญสำหรับการสร้างระบบบนเว็บ
- เมื่อจะสร้างระบบใหม่สักระบบหนึ่ง เราควรนึกถึงการนำส่วนประกอบและระบบซอฟต์แวร์ที่มีอยู่เดิมมาใช้เป็นอันดับแรก

- Incremental and agile development

- ระบบเว็บได้รับการพัฒนาและแจกจ่ายออกมาเรื่อย ๆ ตามความต้องการที่เปลี่ยนไปทุกวัน
- ปัจจุบันนี้ได้รับการยอมรับโดยทั่วไปว่า เราไม่สามารถระบุข้อกำหนดทั้งหมดสำหรับระบบใดๆ ได้ล่วงหน้า ต้องทำไปแก้ไขไปพัฒนาเพิ่มเติมไปเรื่อย ๆ
- เทคนิคการพัฒนาแบบ Incremental และ agile จะช่วยให้พัฒนาซอฟต์แวร์ได้ทันความต้องการของตลาด

- Rich interfaces

- เมื่อเทคโนโลยีการพัฒนาอินเทอร์เน็ตเฟสเช่น AJAX และ HTML5 เกิดขึ้น จะสนับสนุนการสร้าง interface ที่หลากหลายภายในเว็บเบราว์เซอร์

Software engineering ethics

- วิศวกรรมซอฟต์แวร์เกี่ยวข้องกับ ความรับผิดชอบ มากกว่าเพียงแค่การใช้ ทักษะทางเทคนิค ในการทำงานเลี้ยงชีพ
- วิศวกรซอฟต์แวร์จะต้องปฏิบัติตนอย่าง ซื่อสัตย์และมีจริยธรรม หากต้องได้รับการยอมรับว่าเป็นมืออาชีพ
- พฤติกรรมทางจริยธรรมเป็นมากกว่าแค่การ ปฏิบัติตามกฎหมาย แต่เกี่ยวข้องกับ การปฏิบัติตามหลักการที่ถูกต้องตามหลักศีลธรรม

ความรับผิดชอบต่อซอฟต์แวร์ของวิศวกรซอฟต์แวร์

- Confidentiality (ความลับ)
 - วิศวกรควรให้ความสำคัญกับการรักษาความลับของนายจ้างหรือลูกค้า โดยไม่คำนึงถึงว่าได้มีการลงนามในข้อตกลงการรักษาความลับอย่างเป็นทางการหรือไม่
- Competence (ความสามารถ)
 - วิศวกรไม่ควรบิดเบือนความสามารถของตน ไม่ควรที่จะรับงานที่ตนไม่มีความสามารถ
- Intellectual property rights (สิทธิในทรัพย์สินทางปัญญา)
 - วิศวกรควรตระหนักถึงกฎหมายท้องถิ่น ที่เกี่ยวกับการใช้ทรัพย์สินทางปัญญา เช่น สิทธิบัตร ลิขสิทธิ์ ฯลฯ ควรระมัดระวังเพื่อให้แน่ใจว่าทรัพย์สินทางปัญญาของนายจ้างและลูกค้าได้รับความคุ้มครอง
- Computer misuse (การใช้คอมพิวเตอร์ในทางที่ผิด)
 - วิศวกรซอฟต์แวร์ไม่ควรใช้ทักษะทางเทคนิคในการใช้คอมพิวเตอร์ของผู้อื่นในทางที่ผิด การใช้คอมพิวเตอร์ในทางที่ผิดจากระดับง่าย ๆ (เช่น เล่นเกมบนเครื่องของนายจ้าง) ไปจนถึงการกระทำที่รุนแรงมาก (เช่น การเผยแพร่ไวรัส หรือแฮกระบบ)

สรุป

- วิศวกรรมซอฟต์แวร์เป็นสาขาวิศวกรรมที่เกี่ยวข้องกับทุกด้านของการผลิตซอฟต์แวร์
- คุณลักษณะผลิตภัณฑ์ซอฟต์แวร์ที่จำเป็นคือ ความสามารถในการบำรุงรักษา ความเชื่อถือได้ ความมั่นคง ความปลอดภัย ประสิทธิภาพ และการยอมรับ
- กิจกรรมเกี่ยวกับการออกข้อกำหนด การพัฒนา การตรวจสอบ และการบำรุงรักษา เป็นส่วนประกอบของกระบวนการซอฟต์แวร์
- ความคิดพื้นฐานของวิศวกรรมซอฟต์แวร์ จะสามารถใช้งานได้กับทุกประเภทของการพัฒนาระบบซอฟต์แวร์
- ระบบซอฟต์แวร์มีหลายประเภท ต้องการเครื่องมือและเทคนิคทางวิศวกรรมซอฟต์แวร์ที่เหมาะสมสำหรับการพัฒนา
- ความคิดพื้นฐานของวิศวกรรมซอฟต์แวร์สามารถใช้ได้กับระบบซอฟต์แวร์ทุกประเภท
- วิศวกรซอฟต์แวร์มีความรับผิดชอบต่อวิชาชีพด้านวิศวกรรมและสังคม ไม่ควรสนใจแค่เพียงปัญหาทางเทคนิค
- สังคมระดับมืออาชีพ มีจรรยาบรรณไว้คอยกำหนดมาตรฐานเกี่ยวกับพฤติกรรมที่คาดหวังของสมาชิก ในสังคมของวิศวกรซอฟต์แวร์ก็เช่นเดียวกัน

คำถาม???