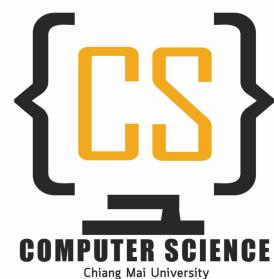


# CS 361 – Software Engineering

## What is Software Engineering?

Kamonphop Srisopha  
Churee Techawut



Faculty of Science, Chiang Mai University  
คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

# Agenda

- What is Software Engineering?
- Role of a Software Engineer
- Importance of Software Engineering

# What is Software Engineering?

# History of Software Engineering

First coined by NATO (ถูกเสนอขึ้นในการประชุมที่จัดโดยคณะกรรมการด้านวิทยาศาสตร์ขององค์กร NATO)



- การประชุมจัดขึ้นที่เมือง Garmisch, Germany
- October 1968
- ถกวิกฤติซอฟต์แวร์ในขณะนั้น ว่าโลกกำลังพึ่งพาซอฟต์แวร์มากขึ้น และขนาดซอฟต์แวร์ก็ใหญ่ขึ้น การพัฒนาต่างๆไม่เป็นไปตามเป้า เลยเวลา เกินงบประมาณ และไม่ง่ายต่อการเปลี่ยนแปลง
- แลกเปลี่ยนความรู้และประสบการณ์ พัฒนาแนวทางและเทคนิคใหม่ๆ ในการพัฒนาซอฟต์แวร์
- ผลลัพธ์ของการประชุมคือรายงาน 100 หน้าที่ชื่อว่า “*Software Engineering*”
- ทำให้คำว่า Software Engineering ได้ถูกยอมรับโดยกว้างขวาง

# History of Software Engineering

## SOFTWARE ENGINEERING

Report on a conference sponsored by the

NATO SCIENCE COMMITTEE

Garmisch, Germany, 7th to 11th October 1968

*Chairman: Professor Dr. F. L. Bauer*

*Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms*

*Editors: Peter Naur and Brian Randell*

January 1969

# History of Software Engineering

## จาก Report

“The phrase ‘**software engineering**’ was deliberately chosen as being provocative, in implying the need for software manufacture to be based on the types of theoretical foundations and practical disciplines, that are traditional in the established branches of engineering.”

วลี "วิศวกรรมซอฟต์แวร์" ถูกเลือกใช้โดยเจตนาที่อยากจะปลุกเร้าความรู้สึกหรือความคิดในการแสดงให้เห็นถึงความจำเป็นในการผลิตซอฟต์แวร์ที่ต้องอ้างอิงหรืออยู่บนฐานหลักการทำงานทฤษฎี และหลักการเชิงปฏิบัติ ซึ่งในแขนงต่างๆ ของ วิศวกรรมศาสตร์ ที่ได้ถูกยอมรับก็มีสิ่งเหล่านี้เป็นพื้นฐาน

# What is Software Engineering?



สามารถวัดผลเชิง  
ปริมาณ

**IEEE:** “The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.”

อย่างมีระบบ

อย่างเป็นระเบียบ



การประยุกต์ใช้วิธีการที่มีระบบ เป็นระเบียบ และ<sup>;</sup>สามารถวัดได้ในการพัฒนา, การดำเนินการ, และ การบำรุงรักษาซอฟต์แวร์ หรืออีกนัยก็คือการ ประยุกต์ใช้วิศวกรรมกับซอฟต์แวร์นั้นเอง

# What is Software Engineering

The process of solving customers' problems by the systematic development and evolution of large, high-quality software systems within cost, time, and other constraints.

# What is Software Engineering

The process of **solving customers' problems** by the systematic development and evolution of large, high-quality software systems within cost, time, and other constraints.

## Solving customers' problems (แก้ไขปัญหาของลูกค้า)

- เป้าหมายหลักของ SE คือการสร้างซอฟต์แวร์ให้ตอบโจทย์ลูกค้าหรือผู้ใช้
- เช่น ลูกค้าอาจต้องการระบบซอฟต์แวร์ที่นำมาติดตามสินค้าในสต็อก, Software Engineer ก็ต้องทำงานร่วมกับลูกค้าเพื่อให้เข้าใจปัญหาและความต้องการ แล้วก็พัฒนาซอฟต์แวร์มาเพื่อแก้ปัญหา และให้ตรงกับความต้องการเหล่านั้น

# What is Software Engineering

The process of solving customers' problems by the **systematic development and evolution** of large, high-quality software systems within cost, time, and other constraints.

## Systematic development and evolution

(การพัฒนาและต่อยอดซอฟต์แวร์โดยใช้กระบวนการอย่างเป็นระบบ)

- SE ไม่ใช่พัฒนาซอฟต์แวร์โดยการโynโค้ดรวมกัน แต่เกี่ยวกับ การปฏิบัติตามกระบวนการที่เป็นระบบ เพื่อให้พัฒนาซอฟต์แวร์ ได้อย่างมีประสิทธิภาพ
- ไม่ใช่แค่กระบวนการทางเทคนิคในการพัฒนาเท่านั้น แต่รวมถึง การจัดการโครงการและการพัฒนาเครื่องมือ วิธีการและอื่นๆเพื่อสนับสนุนการผลิตซอฟต์แวร์

# What is Software Engineering

The process of solving customers' problems by the systematic development and evolution of **large, high-quality software systems** within cost, time, and other constraints.

## Large, high-quality software systems

(ระบบซอฟต์แวร์ที่มีขนาดใหญ่และคุณภาพสูง)

- เทคนิคทาง SE เป็นสิ่งที่จำเป็นให้สามารถพัฒนาระบบซอฟต์แวร์ที่ซับซ้อนและขนาดใหญ่เพื่อให้มีคุณภาพสูง
- จำเป็นต้องมีความเข้าใจอย่างถ่องแท้เกี่ยวกับ สถาปัตยกรรม การออกแบบ การทดสอบ เพื่อประกันคุณภาพ
- ต้องสามารถทำงานได้อย่างมีประสิทธิภาพในสภาพแวดล้อมแบบทีมและจัดการโครงการที่ซับซ้อนได้

# What is Software Engineering

The process of solving customers' problems by the systematic development and evolution of large, high-quality software systems within **cost, time, and other constraints.**

## Cost, time, and other constraints

(เงื่อนไขที่จำกัดต่างๆ เช่น ต้นทุน, เวลา, และอื่นๆ)

- SE ไม่ใช่แค่การพัฒนาซอฟต์แวร์คุณภาพสูงเท่านั้น แต่เป็นการพัฒนาซอฟต์แวร์ที่มีความสอดคล้องกับข้อจำกัดต่างๆ เนื่องจากทรัพยากรมีจำกัด
- ต้องคำนึงถึงปัจจัยต่างๆ เช่น คุณภาพ ต้นทุน เวลา และ scope (เช่น ประโยชน์ของซอฟต์แวร์ที่พัฒนามีค่ามากกว่าต้นทุนที่เสียไปหรือไม่)
- ต้องจัดการกับความเสี่ยงและเหตุการณ์ณต่างๆ

# สรุปความหมายของ Software Engineering

- เกี่ยวกับการพัฒนาซอฟต์แวร์
- เกี่ยวกับการจัดการความซับซ้อนของตัวซอฟต์แวร์
- เกี่ยวกับการพัฒนาที่ใช้หลักการที่มีระบบและเป็นระเบียบ
- เกี่ยวกับการพัฒนาซอฟต์แวร์เพื่อให้สนองความต้องการผู้ใช้
- เกี่ยวกับการประกันคุณภาพซอฟต์แวร์
- เกี่ยวกับการบำรุงรักษาซอฟต์แวร์
- เกี่ยวกับการจัดการทรัพยากรบุคคล
- เกี่ยวกับทำงานเป็นทีม หรือ การทำงานร่วมกันของหลายภาคส่วน
- เกี่ยวกับการบริหารจัดการเพื่อให้พัฒนาซอฟต์แวร์ได้มีคุณภาพ ภายใต้ ข้อกำหนดหรือเงื่อนไขต่างๆ เช่น เวลา เงินทุน scope เป็นต้น

# What makes SE different from other Es?



นักศึกษาคิดว่าการสร้าง software แตกต่างอะไรบ้างเมื่อเทียบกับ  
การสร้างอย่างอื่น (เช่น อาคาร, เครื่องบิน, สะพาน, หรือ รถยนต์)

# Role of a Software Engineer

# More Than Just Writing Code!

**Software Engineering** กล่าวถึงขั้นตอนทั้งหมดในการสร้างซอฟต์แวร์ ซึ่งการเขียนโปรแกรมเป็นแค่ส่วนประกอบหนึ่งในขั้นตอนเหล่านั้น

## Common tasks for Software Engineers:

### 1. Requirements Engineering

- เรากำลังจะพัฒนาซอฟต์แวร์อะไร? เอามันมาแก้ปัญหาอะไรของลูกค้า?

### 2. Specification Writing and Documentation

- เราจะเอาความต้องการของลูกค้ามาเขียนอย่างไรให้ทีมนักพัฒนาซอฟต์แวร์เข้าใจว่าต้องทำอะไรอย่างไร?

### 3. Software Architecture and Design

### 4. Programming

- ใช้ framework, ใช้เครื่องมือ, และปฏิบัติตามหลักการเขียนโค้ดของแต่ละภาษา

“เป็นแค่ส่วนหนึ่งใน SE”

# More Than Just Writing Code!

## Common tasks for Software Engineers (Continued):

### 5. Software Testing (Verification and Validation)

- วิธีการทดสอบ, ทดสอบอะไร อย่างไร, ใช้ continuous integration (CI) อย่างไร

### 6. Release and Deployment

### 7. Maintenance and Evolution

- จะบำรุงรักษาซอฟต์แวร์ที่พัฒนาขึ้นอย่างไรให้มั่นยังใช้ประโยชน์ได้ในอนาคตและพัฒนาต่ออยอดได้

And a Lot Of

# Discussions and Meetings



THE CALOR APP DEVELOPMENT TEAM LEADS A GROUP DISCUSSION WITH FARMWORKERS FROM FABBRI FARMS. PHOTO COURTESY FAITH FLOREZ

# A Software Engineer Can Take on Many Roles

- Software/System Developer/Engineer
- DevOps Engineer
- Requirement Engineer (Business/System Analyst)
- Software Architect
- Tester (Quality Assurance Engineer)
- Database Administrator
- Security Engineer
- UX/UI Designer
- Front/Back-End (Full-Stack) Developer
- Project Manager/ Scrum Master
- Release Manager
- And Many More.....

**More on this later in  
the semester!**

# Software Engineering as CS Course

# Software Engineering as a CS Course

## วิชานี้ไม่ง่ายที่จะสอนภายในหนึ่งเทอม

หากจะสร้าง software โดยคำนึงถึงกิจกรรมของ SE ทั้งหมด (หรือลงลึก) มักจะใช้เวลานานกว่าหนึ่งภาคการศึกษา อีกทั้ง software ขนาดใหญ่ไม่สามารถสร้างขึ้นมาภายใน 3 เดือนได้. จะนับ scale ของ software ที่ทำได้เทียบกับ software ที่นักศึกษาต้องเจอในอุตสาหกรรมซอฟต์แวร์ไม่ได้มาก

## วิชานี้ไม่ง่ายที่นักศึกษาจะเข้าใจหรือสนับสนุนให้กับเนื้อหา

นักศึกษาอยู่ในช่วงชั้นปีที่ต้องการเรียนเกี่ยวกับเนื้อหาเด่น ๆ เรื่องใหม่ๆที่กำลังเป็นกระแส (elective courses เช่น machine learning, artificial intelligence, virtual reality, mobile app, etc.)

# Software Engineering as a CS Course

นักศึกษา



**Software  
Engineering**

# This Course vs Other CS Courses

กระบวนวิชาที่นักศึกษาได้เรียนมาก่อนหน้า อาจจะสอนการเขียนโค้ดโดยภาษาต่างๆ (เช่น *Intro to programming*) หรือสอนให้เขียนโค้ดให้คำนึงถึงประสิทธิภาพ (เช่น *Algorithms, OOP*) หรืออาจเน้นเนื้อหาลงลึกด้านใดด้านหนึ่ง (*electives* เช่น *VR*)



วิชานี้จะเน้นการนำทักษะที่นักศึกษาได้เรียนรู้ก่อนหน้าและในอนาคตไปใช้ในการทำงานเป็นทีม เพื่อสร้างซอฟต์แวร์ที่นักศึกษาและทีมสามารถมั่นใจได้ว่าจะมีคุณภาพและความคุ้มค่าต่อการลงทุนของผู้ใช้งาน และไม่ล้มเหลว



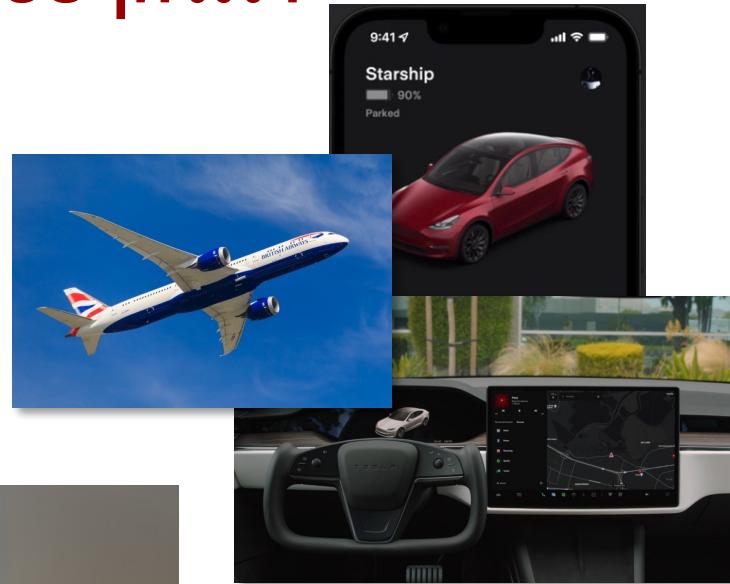
# Why is Software Engineering Important?

# Why is Software Engineering Important?

ซอฟต์แวร์มีอยู่ทุกที่รอบๆตัวเรา



Banking



Transportation



Education



Manufacturing



Healthcare



Fitness



Communication

และในด้านอื่นๆอีกมากมาย

# Why is Software Engineering Important?

ผลกระทบจากซอฟต์แวร์ก็เช่น  
ขึ้นเช่นกันและผลกระทบเป็นวงกว้าง

NEWS

TESLA

British Airw flight delays affecting L

BGR

Home → Science → Space

## James Webb Space Telescope just hit a software glitch

PayPal 'credits' US \$92 quadrillion in e

① 17 July 2013

GETTY IMAGES

More than 200,000 erroneous payments have been taken from car park users in Worcester

REUTERS®

3 minute read · November 3, 2021 6:03 AM GMT+7 · Last Updated a year ago

## Tesla recalls nearly 12,000 vehicles over crash risk caused by software error

theVerge

TECH / AMAZON / WEB

## Amazon's website crashed as soon as it went live

REUTERS

HEALTHCARE & PHARMA NOVEMBER 22, 2013 / 9:55 AM / UPDATED 9 YEARS AGO

## Days before launch, Obamacare website failed to handle even 500 users

By Roberta Rampton

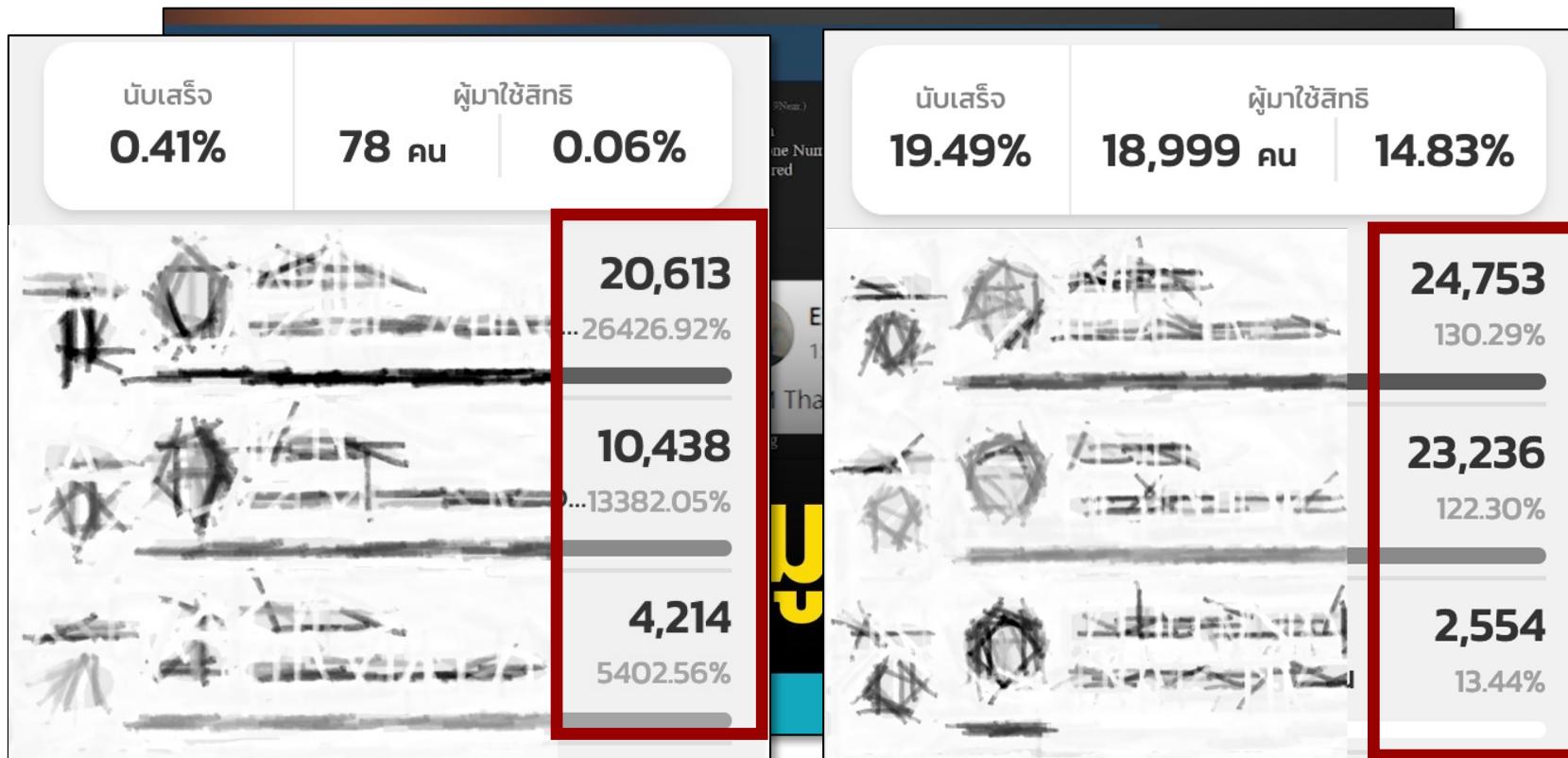
5 MIN READ

f t

26

# Why is Software Engineering Important?

ผลกระทบจากซอฟต์แวร์ก็จะ  
ขึ้นเช่นกันและกระทบเป็นวงกว้าง



ເປົ້າເຊື້ອ (%) ຈຳນວນເສີຍເກີນ 100  
Developer ໄມໄດ້ທົດສອບໃຫ້ ອໍາລັດ ເກີດກາຮຸຈິດ?

# Cost of Software Failure Is Very High

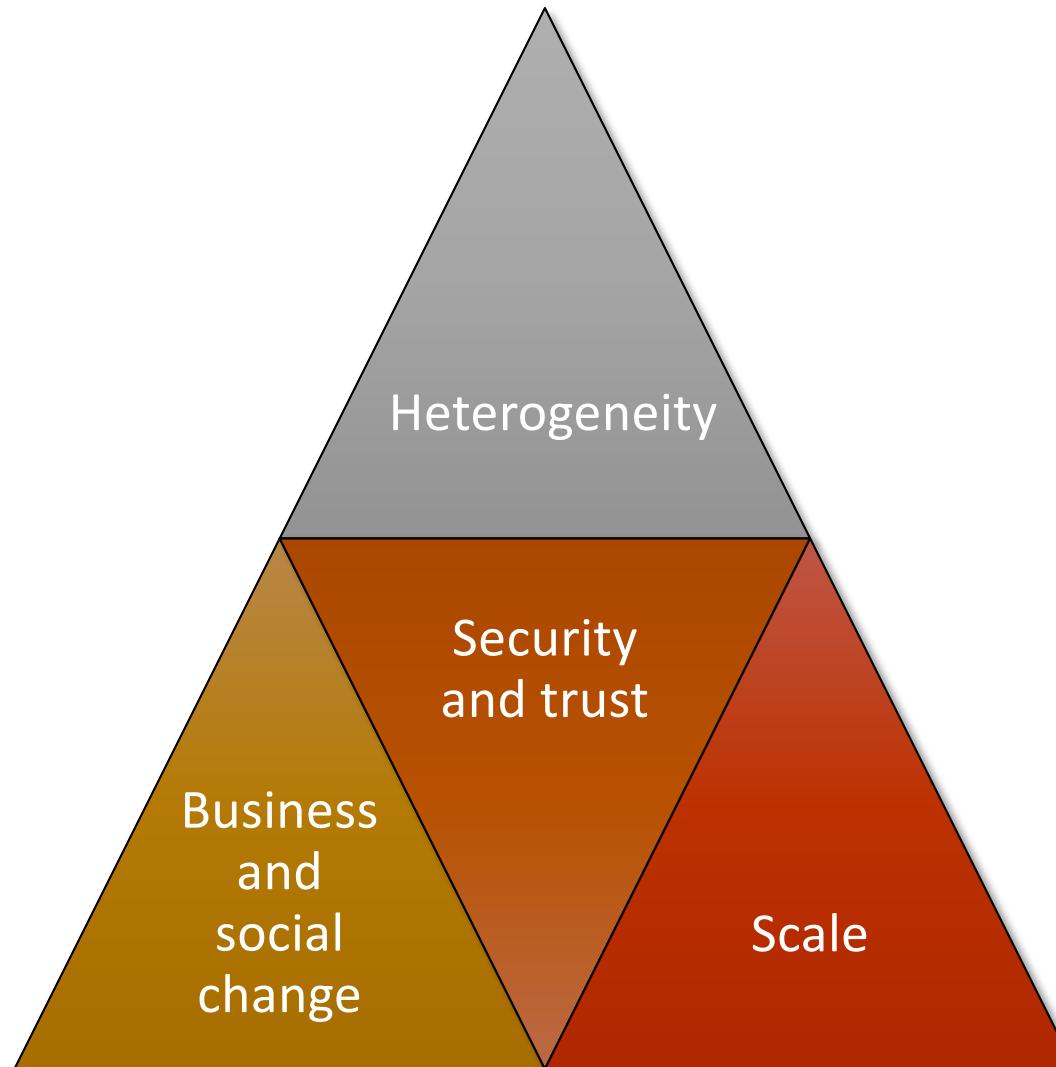
- Financial (including loss of sales)
- Loss of life
- Loss of equipment
- Inconvenience
- Confusion and Chaos
- Loss of trust



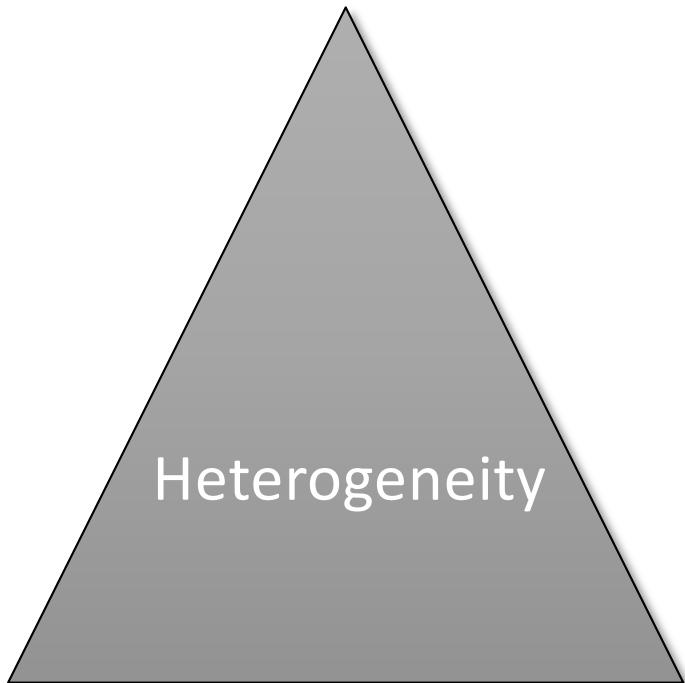
(2022) The cost of poor software quality in the US has grown to at least **\$2.41 trillion**. Operational problems contributed the largest share, with **\$1.56 trillion** in losses. The cost of cybersecurity failures has grown significantly

Read more: <https://bit.ly/3GHU8Y2>

# General Issues Affecting Software



# General Issues Affecting Software



ในปัจจุบันมีความต้องการระบบที่ต้องการทำงานเป็นระบบหรือซอฟต์แวร์แบบ

- กระจาย (Distributed systems)\*
- ที่สามารถเชื่อมต่อกับเครือข่ายที่ประกอบด้วยเครื่องคอมพิวเตอร์และอุปกรณ์เคลื่อนที่หลากหลายชนิด (hardware),
- เข้าถึงได้โดยหลากหลายวิธี (web browsers),
- ที่พัฒนาโดยใช้หรือรองรับภาษา (programming languages) และ Operating System ที่หลากหลาย

\* ในระบบนี้คอมพิวเตอร์หลายเครื่องจะทำงานร่วมกัน (คอมพิวเตอร์จับกลุ่มกันทำงานขนาดใหญ่)

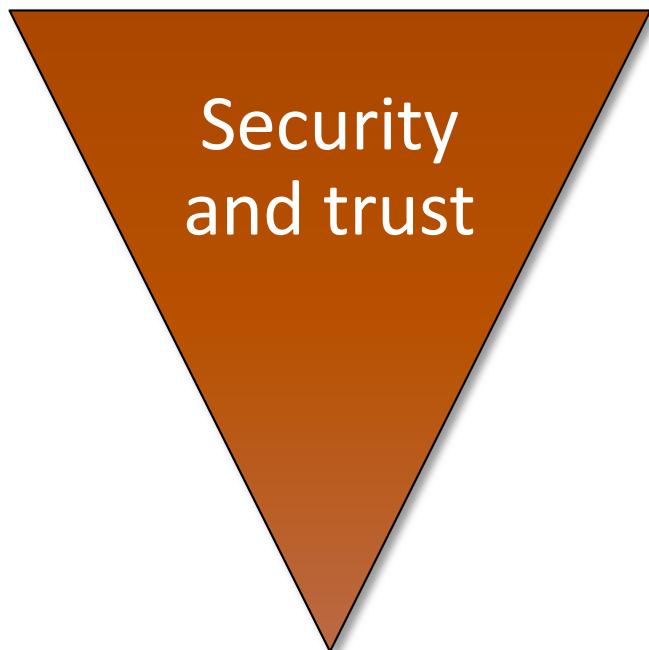
# General Issues Affecting Software



ธุรกิจและสังคมกำลังเปลี่ยนแปลงอย่างรวดเร็วพร้อมกับเศรษฐกิจที่กำลังเจริญเติบโตและเทคโนโลยีใหม่ๆที่มีมาให้ใช้มากขึ้น ( เช่น AI ที่กำลังได้รับกระแสสนใจ หรือ AR/VR ที่กระแสสนใจกำลังเพิ่มขึ้น)

ซอฟต์แวร์ที่พัฒนาต้องเปลี่ยนแปลงได้ง่ายและพัฒนาได้อย่างรวดเร็วเพื่อให้ทันการเปลี่ยนแปลงต่างๆข้างต้น

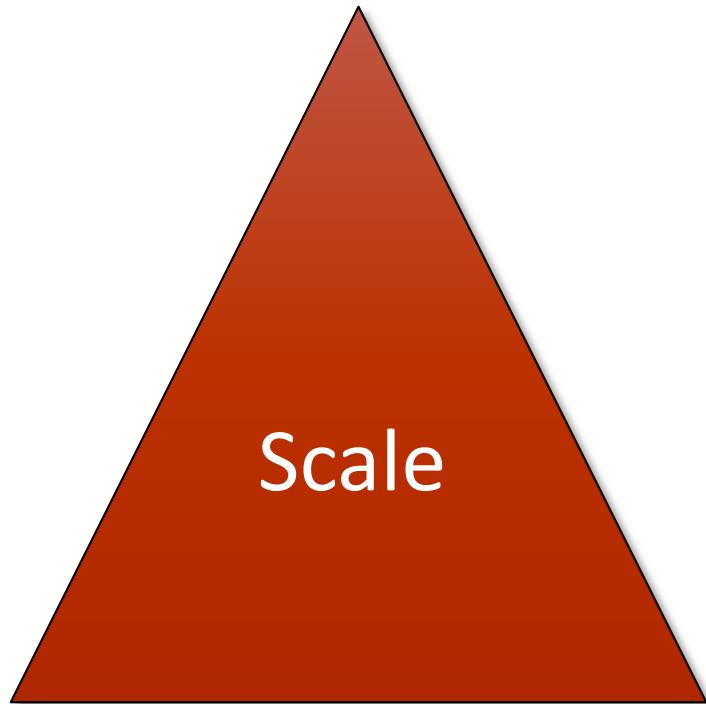
# General Issues Affecting Software



เนื่องจากซอฟต์แวร์เป็นส่วนหนึ่งของทุกด้านของชีวิตเรา จึงเป็นสิ่งสำคัญที่เราควรที่จะสามารถเชื่อมั่นในซอฟต์แวร์นั้นได้ เพื่อที่จะใช้มันได้อย่างปลอดภัย

เช่น software ทางการแพทย์, software ด้านการเงิน, software ที่ควบคุมระบบคมนาคม, software ที่เก็บข้อมูลส่วนตัว เอาไว้

# General Issues Affecting Software



ซอฟต์แวร์ต้องถูกพัฒนาให้ใช้กับหลายช่วง scale ไม่ว่าจะเป็น scale ขนาดเล็กที่อยู่ใน embedded systems ไปจนถึง scale ขนาดใหญ่ เช่นใน distributed system (cloud-based) เพื่อให้บริการกับประชาคมโลกหรือสาธารณะ หรือสามารถรองรับจำนวนคนใช้งานที่เปลี่ยนไปได้อย่างรวดเร็ว

# Let's Think About It!



นักศึกษาคิดว่า Software ที่มีคุณภาพ ควรมีคุณสมบัติ  
อะไร?



นักศึกษาเห็นว่า Task ใดที่ Software Engineer ต้องทำ  
มีความสำคัญที่สุด?

# Let's Do Some Research



จับกลุ่มกับเพื่อนที่นั่งข้างๆ หาตัวอย่าง  
Software Failure จาก Internet

1. Software อะไร เกิดอะไรขึ้น
1. ต้นเหตุของ Failure (Technical, Managerial, or something else) ?
  - มีคนแอบดึงสาย Ethernet Cable, Harddisk พัง
  - Bug
  - โฆษณาเกินจริง
2. ถ้าเกิดเหตุการณ์อย่างนี้ขึ้นในอนาคต นศ. ควรจะรับมืออย่างไร?

ได้ความแล้วให้ นศ. โพสลงใน Discussion Board ใน Mango

# Questions?

