



## บทที่ 2

ความรู้พื้นฐานเกี่ยวกับการทดสอบซอฟต์แวร์

## วัตถุประสงค์การเรียนรู้

- 1) ความหมายของการทดสอบซอฟต์แวร์
  - (K1) ระบุวัตถุประสงค์ทั่วไปของการทดสอบ
  - (K2) แยกการทดสอบจากการดีบั๊ก
- 2) ความจำเป็นของการทดสอบซอฟต์แวร์
  - (K2) ยกตัวอย่างว่าทำไมการทดสอบจึงมีความจำเป็น
  - (K2) อธิบายความสัมพันธ์ระหว่างการทดสอบและการประกันคุณภาพ และยกตัวอย่างให้เห็นว่าการทดสอบมีส่วนช่วยให้มีคุณภาพสูงขึ้นอย่างไร
  - (K2) แยกแยะระหว่างข้อผิดพลาด ข้อบกพร่อง และความล้มเหลว
  - (K2) แยกแยะระหว่างสาเหตุของข้อบกพร่องและผลกระทบ

## วัตถุประสงค์การเรียนรู้ (2/3)

- 3) อธิบายหลักการทดสอบ 7 ประการได้
  - (K2) อธิบายหลักการทดสอบเจ็ดประการ
- 4) กระบวนการทดสอบขั้นพื้นฐาน
  - (K2) อธิบายผลกระทบของบริบทต่อกระบวนการทดสอบ
  - (K2) อธิบายกิจกรรมการทดสอบและงานที่เกี่ยวข้องภายในกระบวนการทดสอบ
  - (K2) สร้างความแตกต่างให้กับผลิตภัณฑ์งานที่รองรับกระบวนการทดสอบ
  - (K2) อธิบายคุณค่าของการรักษาความสามารถในการตรวจสอบย้อนกลับระหว่างเกณฑ์การทดสอบกับผลิตภัณฑ์งานทดสอบ

## วัตถุประสงค์การเรียนรู้ (3/3)

### 5) จิตวิทยาการทดสอบ

- (K1) ระบุปัจจัยทางจิตวิทยาที่มีอิทธิพลต่อความสำเร็จของการทดสอบ
- (K2) อธิบายความแตกต่างระหว่างความคิดที่จำเป็นสำหรับกิจกรรมการทดสอบและความคิดที่จำเป็นสำหรับกิจกรรมการพัฒนา

### 6) เส้นทางอาชีพของการทดสอบซอฟต์แวร์

- (K1) รู้จักเส้นทางอาชีพของการทดสอบซอฟต์แวร์

## เนื้อหาการเรียน

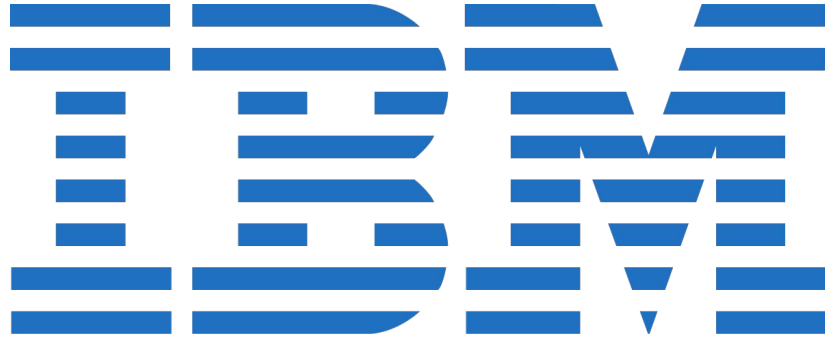
- 1) ความหมายของการทดสอบซอฟต์แวร์
- 2) ความจำเป็นของการทดสอบซอฟต์แวร์
- 3) หลักการทดสอบ 7 ประการ
- 4) กระบวนการทดสอบขั้นพื้นฐาน
- 5) จิตวิทยาการทดสอบ

# หัวข้อที่ 2.1

ความหมายของการทดสอบซอฟต์แวร์

## 2.1 ความหมายของการทดสอบซอฟต์แวร์

“เป็นกระบวนการของการประเมินและตรวจสอบว่าผลิตภัณฑ์ซอฟต์แวร์หรือแอปพลิเคชัน ทำในสิ่งที่ควรทำ”



## 2.1 ความหมายของการทดสอบซอฟต์แวร์ (2/3)

“การทดสอบซอฟต์แวร์คือกระบวนการของการตรวจสอบระบบคอมพิวเตอร์ / โปรแกรม เพื่อพิจารณาว่าเป็นไปตามข้อกำหนดที่ระบุหรือไม่และให้ผลลัพธ์ที่ต้องการ เป็นผลให้สามารถระบุข้อบกพร่องในผลิตภัณฑ์ซอฟต์แวร์ / โครงการได้”

GURU<sup>99</sup>



## 2.1 ความหมายของการทดสอบซอฟต์แวร์ (3/3)

“การทดสอบซอฟต์แวร์คือกระบวนการของการตรวจสอบระบบคอมพิวเตอร์ / โปรแกรม เพื่อพิจารณาว่าเป็นไปตามข้อกำหนดที่ระบุหรือไม่และให้ผลลัพธ์ที่ต้องการ เป็นผลให้สามารถระบุข้อบกพร่องในผลิตภัณฑ์ซอฟต์แวร์ / โครงการได้”

## 2.1 ความหมายของการทดสอบซอฟต์แวร์

“กระบวนการตรวจสอบ หาข้อบกพร่องว่าผลิตภัณฑ์ซอฟต์แวร์ให้ผลลัพธ์เป็นไปตามเงื่อนไขความต้องการของผู้ใช้ที่ระบุไว้ และสามารถแก้ไขข้อบกพร่องก่อนส่งมอบผลิตภัณฑ์ที่มีคุณภาพให้กับผู้ใช้”

## 2.1.3 วัตถุประสงค์ของการทดสอบซอฟต์แวร์

1. เพื่อป้องกันข้อบกพร่อง โดยตรวจสอบจากงานต่อไปนี้ เช่น ข้อกำหนด (Requirements) เรื่องราวของผู้ใช้ (User Story) การออกแบบ และโค้ดโปรแกรม
2. เพื่อตรวจสอบว่ามีการปฏิบัติตามข้อกำหนดทั้งหมดหรือไม่
3. เพื่อตรวจสอบการทำงานของผลิตภัณฑ์ซอฟต์แวร์ให้ตรงตามความต้องการของผู้ใช้งาน
4. เพื่อสร้างความมั่นใจในระดับคุณภาพของผลิตภัณฑ์ซอฟต์แวร์
5. เพื่อค้นหาข้อบกพร่อง (Defect) และความล้มเหลว (Failure) เพื่อลดความเสี่ยงของคุณภาพผลิตภัณฑ์ซอฟต์แวร์ที่ไม่เพียงพอ
6. เพื่อให้ข้อมูลที่เพียงพอแก่ผู้มีส่วนได้ส่วนเสียใช้ตัดสินใจเกี่ยวกับระดับคุณภาพของผลิตภัณฑ์ซอฟต์แวร์ที่นำมาทดสอบ
7. เพื่อทดสอบผลิตภัณฑ์ซอฟต์แวร์ให้ปฏิบัติตามข้อกำหนด มาตรฐาน สัญญา กฎหมาย หรือข้อบังคับต่าง ๆ

## หัวข้อที่ 2.2

ความจำเป็นของการทดสอบซอฟต์แวร์

## 2.2.1 การทดสอบสู่ความสำเร็จ

1. ให้ผู้ทดสอบตรวจสอบข้อกำหนดความต้องการของผู้ใช้
2. ให้ผู้ทดสอบทำงานอย่างใกล้ชิดกับผู้ออกแบบระบบในขณะที่ระบบกำลังได้รับการออกแบบ
3. ให้ผู้ทดสอบทำงานอย่างใกล้ชิดกับนักพัฒนาในขณะที่ได้อยู่ระหว่างการพัฒนา
4. มีนักทดสอบช่วย การทวนสอบ (Verify) และ การสอบทวน (Validate) ผลิตภัณฑ์ซอฟต์แวร์ก่อนการจำหน่าย (Release)

## 2.2.2 การประกันคุณภาพ (Quality Assurance: QA) และการทดสอบ (Software Testing)

“การประกันคุณภาพ” จะเน้นการปฏิบัติตามกระบวนการที่เหมาะสม เพื่อให้มั่นใจว่าจะบรรลุถึงระดับคุณภาพที่เหมาะสม

กิจกรรม “การทดสอบซอฟต์แวร์” เป็นส่วนหนึ่งของ “การควบคุมคุณภาพ” การทดสอบซอฟต์แวร์จะทำให้เกิดความสำเร็จในด้านคุณภาพในแต่ละกระบวนการ และส่งผลให้ได้ผลิตภัณฑ์ที่มีประสิทธิภาพและตรงกับความต้องการของผู้ใช้

### 2.2.3 ข้อผิดพลาด (Errors), ข้อบกพร่อง (Defects), และ ความล้มเหลว (Failures)

1. ระยะเวลาในการทำงาน
2. ความผิดพลาดของมนุษย์
3. สมาชิกในทีมที่ไม่มีประสบการณ์หรือมีทักษะไม่เพียงพอ
4. การสื่อสารที่ผิดพลาดระหว่างสมาชิกในทีม รวมถึงการสื่อสารที่ผิดพลาดเกี่ยวกับข้อกำหนด (Requirement) และการออกแบบ
5. ความซับซ้อนของโค้ด การออกแบบ สถาปัตยกรรม ปัญหาพื้นฐานที่ต้องแก้ไข และ/หรือเทคโนโลยีที่ใช้
6. ความเข้าใจผิดเกี่ยวกับส่วนประสานภายในระบบและระหว่างระบบ โดยเฉพาะอย่างยิ่งเมื่อการโต้ตอบภายในระบบและระหว่างระบบดังกล่าวมีจำนวนมาก
7. เทคโนโลยีใหม่ที่ไม่คุ้นเคย

### 2.2.3 ข้อผิดพลาด (Errors), ข้อบกพร่อง (Defects), และ ความล้มเหลว (Failures)

1. ซอฟต์แวร์ไม่ได้ทำอะไรบางอย่างที่ข้อกำหนดระบุว่าควรทำ
2. ซอฟต์แวร์ทำอะไรบางอย่างที่ข้อกำหนดระบุว่าไม่ควรทำ
3. ซอฟต์แวร์ทำสิ่งที่ข้อกำหนดไม่ได้กล่าวถึง
4. ซอฟต์แวร์ไม่ได้ทำอะไรบางอย่างที่ข้อกำหนดของผลิตภัณฑ์ไม่ได้กล่าวถึง แต่ควรทำ
5. ซอฟต์แวร์ที่นำมาใช้งานยาก หรือต้องใช้ระยะเวลาในการเรียนรู้มาก



## หัวข้อที่ 2.3

หลักการพื้นฐานของการทดสอบ 7 ประการ

## 2.3 หลักการทดสอบ 7 ประการ

1. Testing shows presence of defects
2. Exhaustive testing is impossible
3. Early Testing
4. Defect Clustering
5. Pesticide Paradox
6. Testing is context dependent
7. Absence of errors – Fallacy

## 2.3 หลักการทดสอบ 7 ประการ (2/8)

หลักการที่ 1 การทดสอบซอฟต์แวร์ช่วยลดความน่าจะเป็นของข้อบกพร่องที่ยังไม่ได้ค้นพบในซอฟต์แวร์  
(Testing shows presence of defects)

แต่แม้ว่าจะไม่พบข้อบกพร่อง แต่ก็ไม่ได้เป็นข้อพิสูจน์ถึงความถูกต้อง แต่ถ้ามีการทดสอบเพื่อป้องกันการเกิดข้อ  
ควรระวังทั้งหมดและทำให้ผลิตภัณฑ์ซอฟต์แวร์ไม่มีข้อผิดพลาดถึง 99% และเป็นการป้องกันซอฟต์แวร์ไม่ตรงกับ  
ความต้องการและ/หรือความต้องการของลูกค้า

## 2.3 หลักการทดสอบ 7 ประการ (3/8)

หลักการที่ 2 การทดสอบอย่างละเอียดไม่สามารถทำได้ (Exhaustive testing is impossible)

แต่ต้องสามารถประมาณการหรือวิเคราะห์ถึงจำนวนการทดสอบที่เหมาะสมที่สุดตามการประเมินความเสี่ยงของโปรแกรม

\*\*\* ยกตัวอย่างกรณีทดสอบ Login \*\*\*

## 2.3 หลักการทดสอบ 7 ประการ (4/8)

### หลักการที่ 3      เริ่มการทดสอบเร็วที่สุด (Early Testing)

การทดสอบควรเริ่มต้นเร็วที่สุดเท่าที่จะเป็นไปได้ในวงจรชีวิตการพัฒนาซอฟต์แวร์ เพื่อให้ข้อบกพร่องใด ๆ ในข้อกำหนดหรือขั้นตอนการออกแบบถูกบันทึกไว้ในช่วงเริ่มต้น ค่าใช้จ่ายในการแก้ไขปัญหาลดลงอย่างมากในการแก้ไขข้อบกพร่องในระยะแรกของการทดสอบ

## 2.3 หลักการทดสอบ 7 ประการ (5/8)

### หลักการที่ 4 จัดกลุ่มข้อบกพร่อง (Defect Clustering)

ซึ่งระบุว่าโมดูลจำนวนน้อยที่มีข้อบกพร่องส่วนใหญ่ที่ตรวจพบ นี่คือการประยุกต์ใช้หลักการ Pareto (ในทุกกิจกรรมจะสิ่งที่สำคัญหรือมีประโยชน์ที่จำเป็นจำนวนที่น้อย และมีสิ่งที่ไม่สำคัญหรือไม่มีประโยชน์จำนวนมากว่า ประมาณในอัตราส่วน 20 ต่อ 80) ในการทดสอบซอฟต์แวร์: ประมาณ 80% ของปัญหาพบได้ใน 20% ของโมดูล

ถ้าการทดสอบเดียวกันซ้ำแล้วซ้ำอีกในที่สุดกรณีทดสอบเดียวกันจะไม่พบข้อผิดพลาดใหม่

## 2.3 หลักการทดสอบ 7 ประการ (6/8)

### หลักการที่ 5 ความขัดแย้งกันของสารกำจัดศัตรูพืช (Pesticide Paradox)

การใช้ส่วนผสมของสารกำจัดศัตรูพืชชนิดเดียวกันเพื่อกำจัดแมลงในระหว่างการเพาะเลี้ยงจะนำไปสู่การเกิดแมลงที่ต้านทานต่อยาฆ่าแมลง

ดังนั้นการใช้สารกำจัดแมลงในแมลงจึงไม่เป็นผล เช่นเดียวกับการทดสอบซอฟต์แวร์ ถ้ามีชุดเดียวกันของการทดสอบซ้ำ ๆ จะดำเนินการวิธีการจะไร้ประโยชน์สำหรับการค้นพบข้อบกพร่องใหม่

เพื่อแก้ปัญหานี้กรณีทดสอบจำเป็นต้องได้รับการตรวจสอบและแก้ไขเสมอและควรเพิ่มกรณีทดสอบใหม่ ๆ เพื่อช่วยในการหาข้อบกพร่องเพิ่มเติม

## 2.3 หลักการทดสอบ 7 ประการ (7/8)

### หลักการที่ 6 การทดสอบขึ้นอยู่กับบริบทซึ่งโดยทั่วไป (Testing is context dependent)

การทดสอบขึ้นอยู่กับบริบทซึ่งโดยทั่วไป เช่น หากจะทดสอบเว็บไซต์อีคอมเมิร์ซจะแตกต่างจากที่การทดสอบซอฟต์แวร์สำเร็จรูปทั่วไป

ซอฟต์แวร์ที่พัฒนาแล้วทั้งหมดไม่เหมือนกัน ต้องอาศัยวิธีการหรือเทคนิคและชนิดของการทดสอบที่แตกต่างกันไปขึ้นอยู่กับประเภทของแอปพลิเคชัน

ตัวอย่างเช่นการทดสอบระบบ POS ใด ๆ ที่ร้านค้าปลีกจะแตกต่างจากการทดสอบเครื่องเอทีเอ็ม



## 2.3 หลักการทดสอบ 7 ประการ (8/8)

### หลักการที่ 7 การใช้เหตุผลผิด (Absence of errors – Fallacy)

การทดสอบซอฟต์แวร์ไม่ได้เป็นเพียงแค่การค้นหาข้อบกพร่อง แต่ยังเป็นการตรวจสอบว่าซอฟต์แวร์ตรงตามความต้องการทางธุรกิจหรือไม่ การไม่พบข้อผิดพลาดคือการผิดพลาด

## 2.4 กระบวนการทดสอบขั้นพื้นฐาน : กิจกรรมการทดสอบและงาน

1. การวางแผนการทดสอบ (Test planning)
2. การติดตามและการควบคุมการทดสอบ (Test monitoring and control)
3. การวิเคราะห์แบบทดสอบ (Test analysis)
4. การออกแบบการทดสอบ (Test design)
5. การเตรียมการทดสอบ (Test implementation)
6. การดำเนินการทดสอบ (Test execution)
7. การปิดการทดสอบ (Test completion)

## 2.4 กิจกรรมการทดสอบและงาน : Test monitoring and control

การตรวจสอบการทดสอบเกี่ยวข้องกับการเปรียบเทียบความคืบหน้าจริงกับความคืบหน้าตามแผนอย่างต่อเนื่องโดยใช้เมตริก (Metrics) การตรวจสอบการทดสอบที่กำหนดไว้ในแผนการทดสอบ การควบคุมการทดสอบเกี่ยวข้องกับการดำเนินการที่จำเป็นเพื่อให้เป็นไปตามวัตถุประสงค์ของแผนการทดสอบ

โดยเงื่อนไขที่ใช้ประเมินการบรรลุความสำเร็จของการทำงานในกระบวนการติดตามและการควบคุมการทดสอบ จะใช้เกณฑ์การออกจากการทดสอบ (Exit criteria) หรือ คำจำกัดความของการเสร็จสิ้น (Definition of done)

## 2.5 จิตวิทยาการทดสอบ

1. เริ่มต้นด้วยความร่วมมือไม่ใช่การแข่งขัน
2. เน้นประโยชน์ของการทดสอบ
3. สื่อสารผลการทดสอบและข้อค้นพบอื่น ๆ อย่างเป็นกลางและเน้นข้อเท็จจริงโดยไม่วิจารณ์บุคคลที่สร้างรายการที่มีข้อบกพร่อง
4. พยายามเข้าใจว่าอีกฝ่ายรู้สึกอย่างไรและเหตุผลที่พวกเขาอาจตอบสนองในทางลบต่อข้อมูล
5. ยืนยันว่าอีกฝ่ายว่าเข้าใจสิ่งที่นำเสนอและอาจให้มีการถามกลับหากมีข้อสงสัยเพิ่มเติม



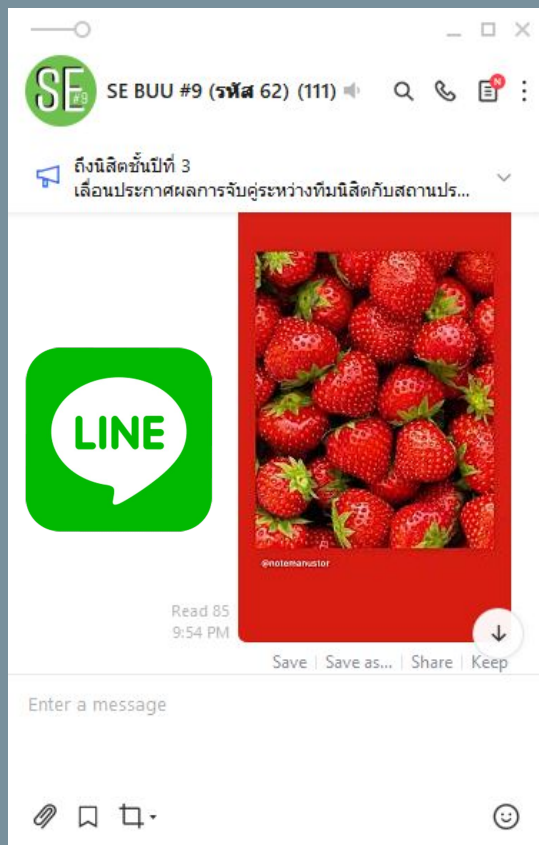
## 2.6 เส้นทางอาชีพของการทดสอบซอฟต์แวร์



## แบบทดสอบ



## แบบฝึกหัด



## สรุปขั้นตอน การจัดการเรื่องร้องเรียน/ร้องทุกข์

