

# ทำความรู้จักกับ Manual Testing

# Manual testing คืออะไร



การทำการทดสอบแบบ Manual คือ การทดสอบที่ผู้ทดสอบ (QA หรือ Tester) จะทำการทดสอบซอฟต์แวร์ หรือระบบ ตามเทสเคส หรือกรณีทดสอบ โดยไม่ได้ใช้เครื่องมืออัตโนมัติใดๆ (Automated tools)



จุดประสงค์ของการทดสอบแบบ Manual คือ การตรวจสอบว่าซอฟต์แวร์ มีข้อผิดพลาดหรือไม่ และซอฟต์แวร์นั้นทำงานได้ตามความต้องการหรือไม่

# ทำไม Manual testing ถึงจำเป็น

- ใช้สำรวจการทำงานของซอฟต์แวร์เบื้องต้น
- ใช้ความคิดสร้างสรรค์ และความรู้เฉพาะทางของ Tester เพื่อค้นหาข้อผิดพลาด
- ใช้ตรวจสอบความง่ายในการใช้งานและประสบการณ์ของผู้ใช้ (Usability testing)
- มีความยืดหยุ่นต่อการเปลี่ยนแปลง เมื่อเทียบกับการทำ Automated Test
- ใช้ทดสอบในช่วงแรกๆ ของการพัฒนาซอฟต์แวร์เพื่อให้ได้ฟีดแบคที่รวดเร็ว ก่อนที่จะทำ Automated Test

# Manual testing



**VS.**

# Automated testing

---

# Automated testing คือ

Automated testing คือ กระบวนการทดสอบที่ใช้เครื่องมือ หรือสคริปต์ในการทดสอบซอฟต์แวร์โดยอัตโนมัติ เพื่อตรวจสอบการทำงานของซอฟต์แวร์

จุดประสงค์ของการทำ Automated testing

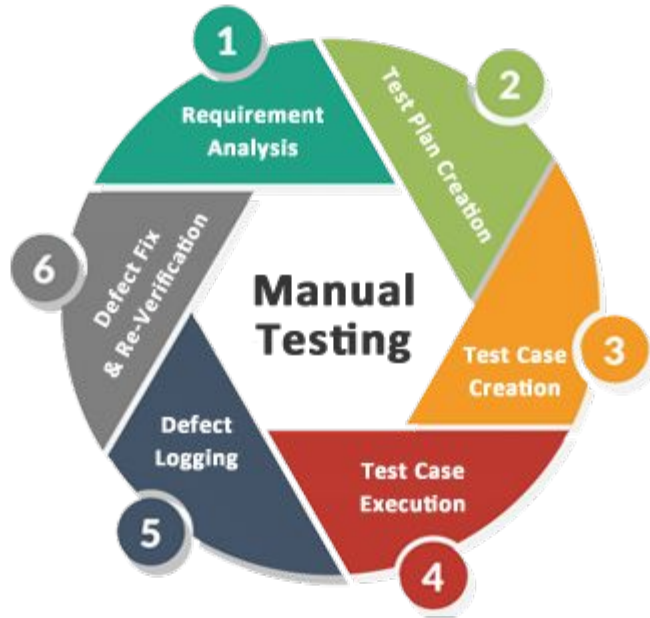
- ลดเวลาที่ใช้ในการทดสอบซ้ำๆ เช่น regression test, smoke test
- ทำให้การทดสอบมีประสิทธิภาพมากขึ้น เช่น สามารถรันสคริปต์ทิ้งไว้ได้ข้ามคืน
- ทำให้การทดสอบมีความสม่ำเสมอ เพราะรันด้วยสคริปต์เดิมทุกครั้ง

# ความแตกต่าง Manual testing vs. Automated testing

Criteria	Manual testing	Automated testing
ความเร็วในการทดสอบ	ใช้เวลามากเนื่องจากใช้คนทำ	ใช้นเวลาน้อยกว่า Manual test เพราะเป็นการรันสคริปต์
การลงทุนในการทำ (Investment)	ใช้เวลาและแรงงานน้อยกว่าตอนเริ่มสร้าง	ใช้เวลาและแรงงานมากกว่าในการเริ่มต้นสร้าง
ประเภทของการทดสอบ	Exploratory testing, Usability testing หรือการทดสอบที่ต้องใช้คนในการตัดสินใจ	Regression testing, Smoke testing หรือการทดสอบเดิมๆ ที่ต้องทำซ้ำบ่อยๆ

# กระบวนกรของ Manual Testing

## PROCEDURE FOLLOWED IN MANUAL TESTING



1. ทำความเข้าใจความต้องการของ Business (Business requirement)
2. การวางแผนงานการทดสอบ (Test planning)
3. การสร้างเทสเคส (Test case development)
4. การดำเนินการทดสอบ (Test execution)
5. การรายงานผลการทดสอบและการติดตามสถานะของ Bug และ Defect
6. การทดสอบซ้ำและการทำ regression testing



# ทำความเข้าใจความต้องการของ Business (Business requirement)

- ทำความเข้าใจ Business Domain หรือความรู้เฉพาะเกี่ยวกับธุรกิจของซอฟต์แวร์ ที่เราพัฒนา
- ทำความเข้าใจ Architecture และ Design ต่างๆ
- วิเคราะห์ Test conditions หรือเงื่อนไขในการทดสอบจากความต้องการทางธุรกิจ



ตัวอย่าง Test conditions ของ shopping cart หรือตะกร้าสินค้าของ e-commerce web

- การเพิ่มสินค้าในตะกร้า
- การอัปเดตสินค้าในตะกร้า
- การลบสินค้าจากตะกร้า

# การวางแผนงานการทดสอบ (Test planning)

การสร้างเอกสารที่ประกอบไปด้วย ภาพรวมของวิธีการ และจุดประสงค์ของการทำทดสอบ รวมถึง

- ขอบเขต (Scope) ของซอฟต์แวร์ที่จะทำการทดสอบ
- สิ่งที่ต้องเตรียมสำหรับการทดสอบ เช่น test environments, test data
- แผนงาน และเวลาที่จะใช้ในการทดสอบโดยประเมิน และประเภทของการทดสอบต่างๆ

# การสร้างเทสเคส (Test case development)



การสร้างเทสเคส หรือกรณีทดสอบ โดยการใช้เทคนิคการสร้างเทสเคสต่างๆ และการเตรียมข้อมูลเพื่อการทดสอบ (test data)

เทสเคส หรือกรณีทดสอบ ประกอบไปด้วย

- เงื่อนไขที่ต้องเตรียมก่อนทดสอบ (Pre-conditions)
- ลำดับขั้นตอนการทดสอบ (Test steps)
- ข้อมูลที่ใช้ในการทดสอบ
- ผลลัพธ์ที่คาดหวังของการทดสอบ (Expected result)

# การดำเนินการทดสอบ (Test execution)

ขั้นตอนสำคัญ เพื่อทดสอบว่า ซอฟต์แวร์ หรือระบบที่พัฒนา  
ทำงานได้ตรงตามความต้องการ โดยจะทำการทดสอบตาม  
เทสเคส หรือ กรณียกทดสอบที่เราเตรียมไว้

เมื่อทำการทดสอบ จะมีการอัปเดตสถานะ เช่น Passed,  
Failed หรือ Skipped และ comment เพิ่มเติม



# การรายงานผลการทดสอบและการติดตามสถานะของ Bug และ Defect



- ถ้าพบข้อผิดพลาด หรือ bug ระหว่างการทดสอบ จะต้องบันทึกลงในระบบ Test management tool
- การรายงานผลการทดสอบ เกิดขึ้นหลังจากทำการทดสอบเสร็จ เพื่อบันทึกว่าโดยรวมแล้วผลการทดสอบผ่านหรือไม่

# การทดสอบซ้ำ และการทำ regression testing

## Re-validation หรือ Re-testing

เมื่อข้อผิดพลาดที่รายงานไปได้รับการแก้ไข QA หรือ Tester จะทำการทดสอบจุดที่ผิดพลาดอีกครั้ง เพื่อให้มั่นใจว่าไม่พบข้อผิดพลาดเดิมแล้ว

## Regression Test

การทดสอบที่ตรวจสอบว่า ซอฟต์แวร์ หรือระบบยังทำงานได้ปกติ หลังจากมีการเปลี่ยนแปลง เช่น การแก้ไขข้อผิดพลาด หรือมี feature ใหม่

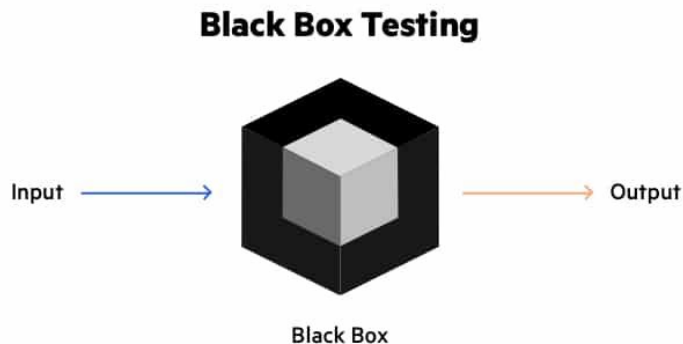
# เทคนิคการทำ Manual Testing

# Black Box Testing

การทดสอบแบบ Black box คือ วิธีการทดสอบที่ผู้ทดสอบจะตรวจสอบการทำงานของซอฟต์แวร์ โดยไม่มีความรู้เกี่ยวกับโค้ด โครงสร้าง หรือรายละเอียดเกี่ยวกับการทำงานภายในซอฟต์แวร์

คุณลักษณะสำคัญของการทดสอบแบบ Black box

- ผู้ทดสอบไม่จำเป็นต้องรู้เกี่ยวกับ logic หรือตรรกะของโค้ด
- เทสเคส จะสร้างจากความต้องการทางธุรกิจ และของผู้ใช้งาน



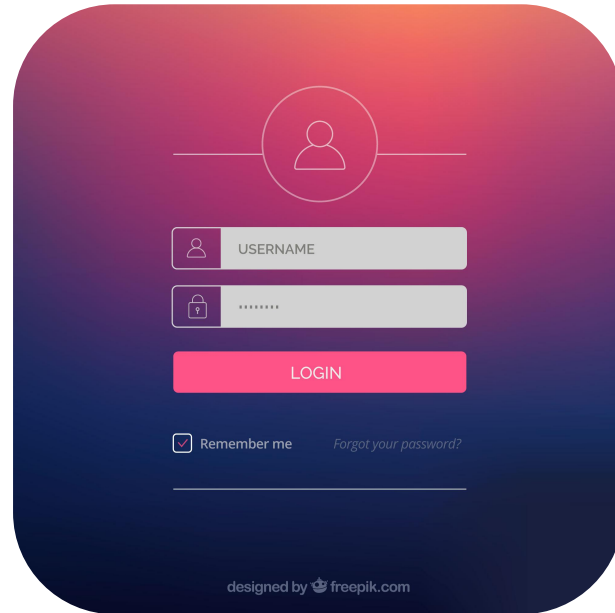


# ตัวอย่าง Black box testing

การทดสอบหน้า Login ของแอปพลิเคชัน

โดย ผู้ใช้งานกรอก username และ password เพื่อเข้าใช้งานแอปพลิเคชัน

1. ผู้ใช้งานกรอก username และ password ที่ถูกต้อง  
⇒ ผลลัพธ์ที่คาดหวัง คือ ผู้ใช้งานสามารถ login เข้าสู่ระบบได้
2. ผู้ใช้งานกรอก username และ password ที่ไม่ถูกต้อง  
⇒ ผลลัพธ์ที่คาดหวัง คือ ผู้ใช้งานไม่สามารถ login เข้าสู่ระบบได้



# White Box Testing

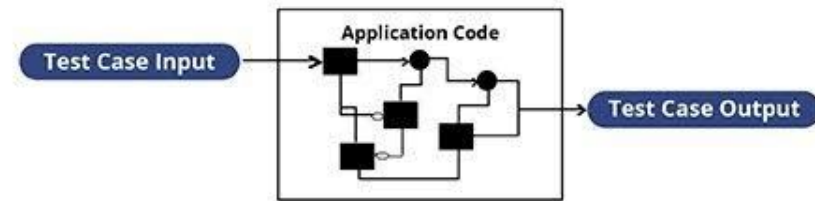
---

ในการทดสอบแบบ White box ผู้ทดสอบซอฟต์แวร์ จะมีความรู้เกี่ยวกับโค้ด ตรรกะ และโครงสร้างของโค้ด

คุณลักษณะสำคัญของการทดสอบแบบ White box

- ผู้ทดสอบต้องรู้เกี่ยวกับ logic หรือตรรกะของโค้ด
- เน้นการตรวจสอบความถูกต้องของโค้ด และความครอบคลุม (coverage)

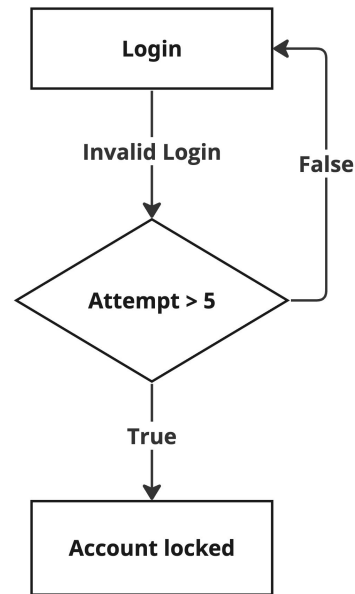
## WHITE BOX TESTING APPROACH



# ตัวอย่าง White box testing

การทดสอบหน้า Login ของแอปพลิเคชัน โดยทดสอบเงื่อนไขของการ Login ไม่ถูกต้องเกินจำนวนที่กำหนดไว้

- ใช้การทดสอบแบบ **branch coverage** หรือทดสอบเงื่อนไข if statement ที่ถูกเรียกใช้งานทั้ง true และ false
- เงื่อนไข true : Login ไม่ถูกต้องเกินจำนวนที่กำหนด
  - ผลลัพธ์ คือ account ถูกล็อก
- เงื่อนไข false : Login ไม่ถูกต้อง แต่ยังไม่เกินจำนวนที่กำหนด
  - ผลลัพธ์ คือ account ไม่ถูกล็อก



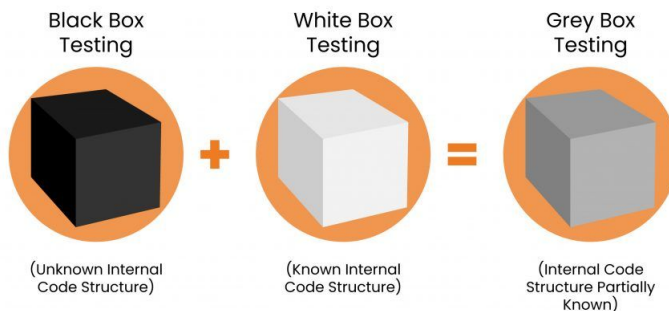
# Grey Box Testing

การทดสอบแบบ Grey box คือ วิธีการทดสอบที่รวมวิธีแบบ Black box และ White box

คุณลักษณะสำคัญของการทดสอบแบบ Grey box

- ผู้ทดสอบมีความรู้เกี่ยวกับการทำงานของโค้ดบางส่วน รวมถึงการเข้าถึง database schema หรือ API document

## Types Of Testing Methods



# ตัวอย่าง Grey box testing

---

การทดสอบหน้า Login ของแอปพลิเคชัน

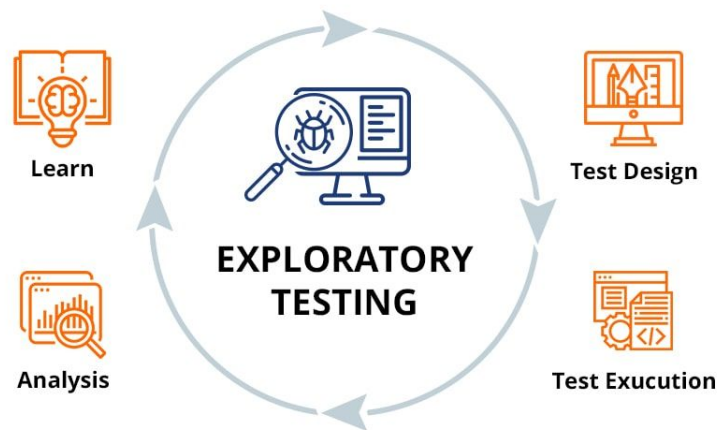
1. ตรวจสอบใน database ว่าเมื่อผู้ใช้งาน Login ผิดเกินจำนวนที่กำหนดไว้ สถานะของ account จะเปลี่ยนเป็นล็อก
2. ตรวจสอบการ Login โดยใช้ username และ password ที่ไม่ถูกต้อง ผ่าน API testing เพื่อตรวจสอบ response และ error ที่ได้กลับมา

# Exploratory Testing

การทดสอบที่ใช้เพื่อเรียนรู้การทำงานของซอฟต์แวร์หรือระบบ โดยการคิดว่าจะทดสอบอะไร และทำการทดสอบในระยะเวลาที่จำกัด

ขั้นตอนของการทำ Exploratory testing คือ

- การออกแบบการทดสอบ
- การทำการทดสอบ
- การสังเกตและเรียนรู้ระบบว่าสามารถทำงานได้อย่างไรบ้าง

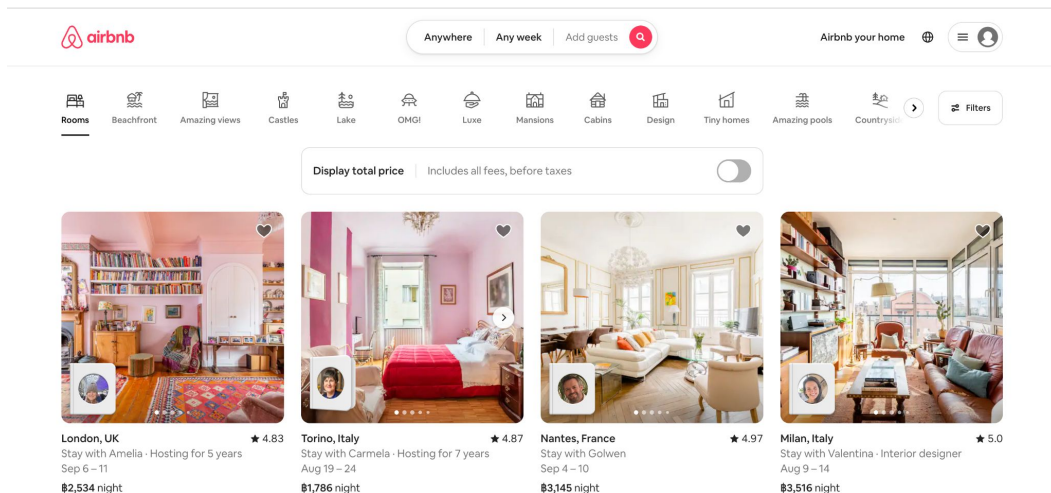


# Mini workshop

---

# ทำการทดสอบ Exploratory Testing

ทดสอบการค้นหที่พึก (Search) บนเว็บไซต์ airbnb (<https://www.airbnb.com/>)





# การทดสอบการค้นหาที่פק

## (Search function)

1. Search by Region  
⇒ ผลการค้นหาจะต้องตรงตาม Region ที่เลือกไว้
2. Search โดยพินพืหาสถานที่ หรือประเทศ  
⇒ ผลการค้นหาจะต้องตรงตามสถานที่ หรือประเทศที่ระบุ
3. Search โดยระบุวันเชคอิน เชคเอาท์ แบบต้องตรงวันเท่านั้น (Exact date)  
⇒ ผลการค้นหาจะต้องตรงตามวันที่ระบุไว้
4. Search โดยระบุจำนวน และประเภทของผู้เข้าพัก  
⇒ ผลการค้นหาจะต้องตรงตามจำนวน และประเภทของผู้เข้าพัก

# ตัวอย่างเอกสารต่างๆ ในการทำ Manual Testing

---

# แผนการทดสอบ (Test plan)

docs.google.com/document/d/1TtUkbD9BZ7PyK8901TPu2JVA...y/EPHIA181zPp5M/edit

File Edit View Insert Format Tools Extensions Help

Normal text Arial 11 B I U

Share

## แผนการทดสอบเว็บแอปพลิเคชัน E-commerce XYZ

โดย Tester ABC

[หน้า \(Introduction\)](#)  
[วัตถุประสงค์ \(Objective\)](#)  
[ขอบเขตของการทดสอบ \(Scope\)](#)  
[ขอบเขตการทดสอบเฉพาะกลุ่ม \(In Scope\)](#)  
[ขอบเขตการทดสอบนอกขอบเขต \(Out of Scope\)](#)  
[ประเภทของการทดสอบในแผนการทดสอบ \(Test Types\)](#)  
[เอกสารและสิ่งส่งส่งที่เกี่ยวข้องกับการทดสอบ \(Test Deliverables\)](#)  
[Test Environment ที่ใช้ในการทดสอบ](#)  
[เกณฑ์ในการเริ่มการทดสอบ \(Entry Criteria\)](#)  
[เกณฑ์ในการจบการทดสอบ \(Exit Criteria\)](#)  
[ความเสี่ยงและปัจจัย และแผนการความเสี่ยง \(Risk and Contingencies\)](#)  
[ทรัพยากรที่จำเป็น \(Resource\)](#)  
[ระยะเวลาที่ใช้ในการทดสอบโดยประมาณ และตารางงาน \(Estimation and Schedule\)](#)

บทนำ (Introduction)

จุดประสงค์หลักของเอกสารแผนการทดสอบนี้ ให้อธิบายรายละเอียดเกี่ยวกับกลยุทธ์ในการทดสอบเว็บแอปพลิเคชัน E-commerce XYZ และใช้เป็นแนวทางในการทำการทดสอบตามแผนอย่างเคร่งครัด ไม่จำเป็นต้องทำตามแบบอย่างของ E-commerce XYZ นี้ทุกประการ โดยแผนการทดสอบนี้จัดทำขึ้นเพื่อให้ใช้งานตามการใช้งานได้ส่วนประกอบหลัก และได้รับอนุมัติจากทีมที่เกี่ยวข้องในการใช้งาน

วัตถุประสงค์ (Objective)

เป้าหมายของการทดสอบเว็บแอปพลิเคชัน E-commerce XYZ คือ

- ตรวจสอบการทำงานของเว็บแอปพลิเคชันที่กำหนดไว้ในขอบเขตของการทดสอบ
- ตรวจสอบความถูกต้อง และความเป็นตัวของตัวเอง และการใช้งานของผู้ใช้งาน
- ตรวจสอบความเสถียรภาพของระบบและประสิทธิภาพในการใช้งานของผู้ใช้งาน
- ตรวจสอบการใช้งาน platform หรือเครื่องมือต่างๆ และ browsers ที่แตกต่างกัน

## PROJECT TEST PLAN

## IN SCOPE

## OUT OF SCOPE

## RISKS

## INTRODUCTION

## ENVIRONMENTS

## PEOPLE

## TOOLS

## TIMESCALES



# รายงานผลการทดสอบ

## (Test report)

Test Report					
Test Cycle		System Test			
EXECUTED	PASSED				130
	FAILED				0
	(Total) TESTS EXECUTED				130
	(PASSED + FAILED)				
PENDING					0
IN PROGRESS					0
BLOCKED					0
(Sub-Total) TEST PLANNED					130
(PENDING + IN PROGRESS + BLOCKED + TEST EXECUTED)					

Functions	Description	% TCs Executed	% TCs Passed	TCs pending	Priority	Remarks
New Customer	Check new Customer is created	100%	100%	0	High	
Edit Customer	Check Customer can be edited	100%	100%	0	High	
New Account	Check New account is added	100%	100%	0	High	
Edit Account	Check Account is edit	100%	100%	0	High	
Delete Account	Verify Account is delete	100%	100%	0	High	
Delete customer	Verify Customer is Deleted	100%	100%	0	High	
Mini Statement	Verify Ministatement is generated	100%	100%	0	High	
Customized Statement	Check Customized Statement is generated	100%	100%	0	High	

### E-commerce Website Test Report

Project: XYZ E-commerce Website Testing

Test Period: July 1, 2023 - July 15, 2023

#### 1. Summary:

The testing phase for the XYZ E-commerce website was conducted to validate the functionality, usability, and performance of the platform. This report provides an overview of testing activities, test execution results, and identified defects.

#### 2. Testing Scope:

The testing scope covered user registration, product browsing, cart management, checkout process, payment processing, and order confirmation functionalities of the e-commerce website.

#### 3. Testing Approach:

A combination of manual black box testing and automated regression testing was used to ensure comprehensive coverage.

#### 4. Test Execution Results:

Module	Total Test Cases	Passed	Failed	Blocked	Not Executed
User Registration	2	2	0	0	0
Product Browsing	3	3	0	0	0
Cart Management	3	3	0	0	0
Checkout and Payment	4	3	1	0	0
Order Confirmation	2	2	0	0	0
Total	14	13	1	0	0

# การใช้งาน Test Management Tool และ Bug Tracking Tools

# ทำความรู้จักกับ Test management tool และ Bug tracking tool

---

TestRail

 Jira Software

 ZEPHYR

 XRAY

Test Management Tool คือ เครื่องมือที่ใช้  
ในการจัดการการทดสอบ โดยจะสามารถ  
จัดการแผนการทดสอบ เทสเคส รวมถึง  
ผลการทดสอบ และสามารถสืบค้น bug ที่  
เจอระหว่างการทดสอบ โดยสามารถติดตาม  
สถานะของการแก้ไขได้ (bug tracking  
tools)

# ตัวอย่างของ Test management tool

**TITAN / TITAN-45**  
Validate "Calendar" is the first application to load.

[Edit](#) [Comment](#) [Admin](#) [Clone](#) [More Actions](#) [Execute](#)

**Details**

Type: Test Status: TO DO [\(View Workflow\)](#)  
 Priority: Medium Resolution: Unresolved  
 Labels: None

**Description**

Here is a place for requirements. TestcaseTreeExport\_1511889287129\_csv

**Test Details**

**Test Executions**

[Freeze version and status](#) [Columns](#)

Version	Status	Test Cycle	Folder	Defects	Executed By	Actions
Unscheduled	FAIL	Ad hoc	-	-	admin	<a href="#">E</a> <a href="#">D</a>
Version 1.0	UNEXECUTED	Sprint 1 Tests	New Folder	-	-	<a href="#">E</a> <a href="#">D</a>
Version 1.0	FAIL	Sprint 3	IOS	0   1	admin	<a href="#">E</a> <a href="#">D</a>
Version 1.0	UNEXECUTED	Sprint 3	Android	-	-	<a href="#">E</a> <a href="#">D</a>
Version 1.0	UNEXECUTED	Sprint 3	-	-	-	<a href="#">E</a> <a href="#">D</a>
Version 1.0	PASS	Sprint 2	New Folder	1   0	TITAN-69 admin	<a href="#">E</a> <a href="#">D</a>
Version 1.0	FAIL	Sprint 2	Calendar	0   1	admin	<a href="#">E</a> <a href="#">D</a>

Showing 9 of 9

Test Summary [Cycle Summary](#) Traceability

**Test Cycles and Folders**

Search cycle / folder

**Test Executions**

[Calendar](#) [List](#) [Detail](#)

Total Executed : 4 Total Executions : 4  
 Total Execution Time : 0m Total Logged Time : 0m Executions Not Tracked : 4

[Select All](#) [Delete](#) [Columns](#) [+ Add Tests](#)

ID	Status	Summary	Defects	Component	Label	Executed By	Actions
TITAN-51	PASS	Validate events are loading to the calendar	-	Calendar		admin	<a href="#">E</a> <a href="#">D</a>
TITAN-52	PASS	Validate event dates correspond to CMS master	-	Calendar		admin	<a href="#">E</a> <a href="#">D</a>
TITAN-45	FAIL	Validate "Calendar" is the first application to load.	0   1	Calendar	calendar	admin	<a href="#">E</a> <a href="#">D</a>
TITAN-78	FAIL	Validate Calendars Default to Day View	-	Calendar	snoke	admin	<a href="#">E</a> <a href="#">D</a>

1 - 4 < 1 - 4 > 10



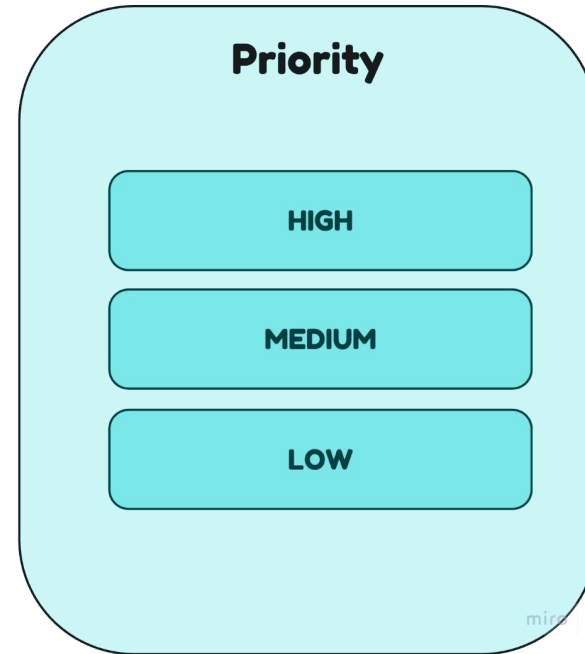
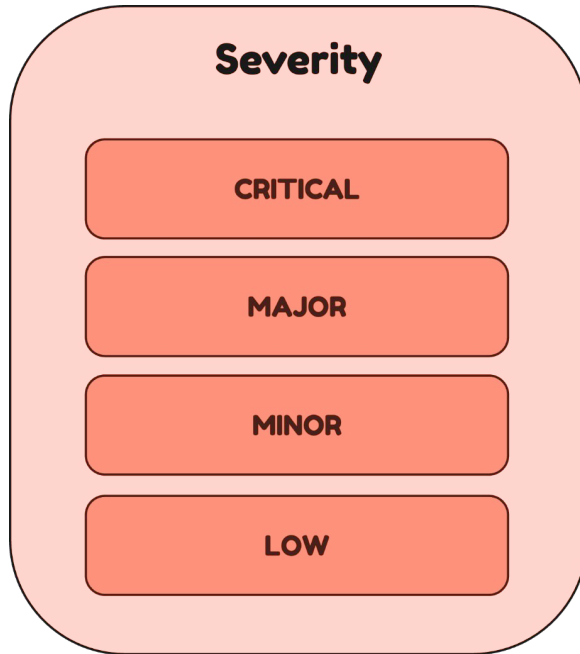
# การจัดการและการบันทึก Bug และ Defect

---

# การบันทึกบัค (Bug Report)

<b>Topic/Title</b>	หัวข้อ
<b>Bug description</b>	คำอธิบายเกี่ยวกับบัค
<b>Severity</b>	ระดับความรุนแรงของบัค
<b>Priority</b>	ระดับความสำคัญในการแก้ไข
<b>Steps to reproduce</b>	ขั้นตอนการทดสอบที่เจอปัญหา
<b>Expected result</b>	ผลลัพธ์ที่คาดหวัง
<b>Actual result</b>	ผลที่เกิดขึ้นจากการทำตามขั้นตอนการทดสอบ
<b>Environment</b>	Environment ที่เราใช้ทดสอบ เช่น Test, Staging
<b>Attachment/Screenshot/Video</b>	เอกสารแนบ หรือ screenshot หรือ video ที่อัดขั้นตอนการทดสอบ

# Bug Severity vs Priority



# Bug Severity = ระดับความร้ายแรง

CRITICAL

**ระดับร้ายแรงมาก** เช่น บัคที่ทำให้ระบบล่ม ไม่สามารถใช้งานได้

MAJOR

**ระดับร้ายแรง** เช่น บัคที่ทำให้การทำงานหลักไม่ปกติ แต่ไม่ทำให้ระบบล่ม

MINOR

**ระดับปานกลาง** เช่น บัคที่ทำให้การทำงานไม่ปกติ แต่การทำงานหลักยังใช้งานได้

LOW

**ระดับต่ำ** เช่น บัคที่มีผลกระทบต่อระบบ พวก cosmetic bug ต่างๆ

# Priority = ลำดับความสำคัญในการแก้ไข

HIGH

**สำคัญมาก** ต้องทำการแก้ไขอย่างเร่งด่วน หรือทันที

MEDIUM

**สำคัญปานกลาง** ต้องทำการแก้ไข แต่ไม่จำเป็นต้องเร่งด่วน หรือทันที

LOW

**สำคัญน้อย** จะทำการแก้ไขเมื่อทีมแก้ไขบัคระดับ High และ Medium แล้ว

# ตัวอย่างการบันทึก Bug บน Jira

Jira Software

Your work ▾

Projects ▾

Filters ▾

Dashboards ▾

Teams ▾

Apps ▾

Create

demo

Software project

You're on the Free plan

UPGRADE

PLANNING

Timeline

Backlog

Board

DEVELOPMENT

Code

Project pages

QAlity

Add shortcut

Project settings

Projects / demo / Add epic / DEMO-1

Checkout Payment processing failed with valid credit card details

Attach

Add a child issue

Link issue ▾

QAlity

...

Severity

Major

Priority

High

Description

**Description:** During the payment processing step of the checkout process, attempting to complete the order with valid credit card details results in an error message. The payment process is not completed successfully, and users receive an error indicating "Invalid card information."

**Steps to Reproduce:**

1. Add items to the cart and proceed to the checkout process.
2. Enter valid credit card details
3. Click the "Complete Order" button.

**Expected Result:** The payment processing should allow users to proceed with valid credit card details, and no error messages should be displayed.

**Actual Result:** An error message appears indicating "Invalid card information," and users cannot complete the order.

Attachments (2)

credit card details.txt  
09 Sep 2023, 10:14 PM

Bank Credit card

Invalid card details. Please use an error code.

Error\_message...ard.png  
09 Sep 2023, 10:14 PM

To Do ▾

Actions ▾

Details

Assignee

PT Pirada Tipparat

Labels

None

Sprint

None

Story point estimate

None

Reporter

PT Pirada Tipparat

Created August 9, 2023 at 11:28 PM

Updated 4 minutes ago

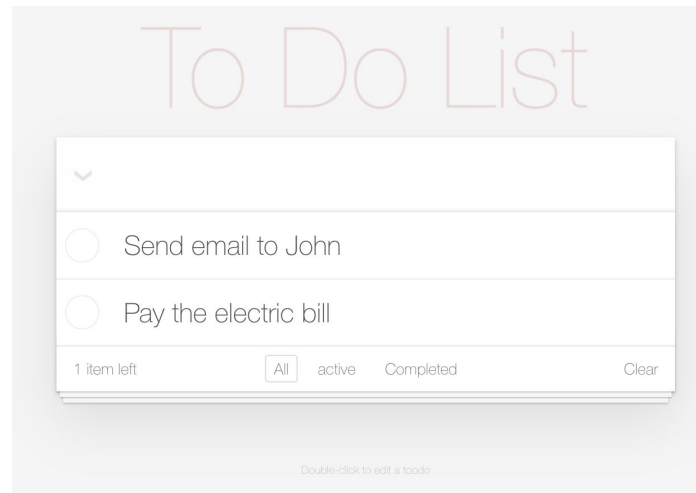
Configure

# Mini workshop

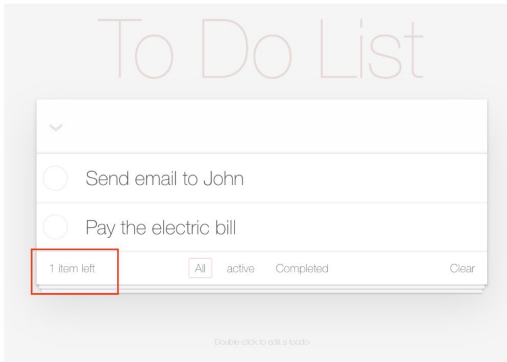
---

# โจทย์: บันทึกรัก bug ที่เจอในเวปไซต์

1. ไปที่เวปไซต์ <https://todolist.james.am/#/>
2. หา bug หรือข้อผิดพลาดบนเวปไซต์
3. บันทึกรัก bug หรือข้อผิดพลาดที่เจอลงบน google doc ตาม template ที่กำหนด





Topic/Title	จำนวน item left แสดงค่าผิด
Bug description	จำนวน item left ที่ใช้แสดงจำนวน item ใน To do list แสดงค่าผิด โดยแสดงค่าน้อยกว่าจำนวน item ที่เพิ่มเข้าไป 1 จำนวน เช่น มี item จำนวน 2 item แต่จำนวน item left จะแสดงเป็น 1 item left
Severity	Medium
Priority	Medium
Steps to reproduce	<ol style="list-style-type: none"> <li>1. ไปที่เว็บไซต์ <a href="https://todolist.james.am/#/">https://todolist.james.am/#/</a></li> <li>2. เพิ่ม item ใน To do list จำนวน 2 item ดังนี้ <ul style="list-style-type: none"> <li>- Send email to John</li> <li>- Pay the electric bill</li> </ul> </li> <li>3. สังเกตจำนวน item left</li> </ol>
Expected result	จำนวน item left จะต้องแสดงเป็น 2 items left
Actual result	จำนวน item left แสดงเป็น 1 item left
Attachment/ Screenshot/ Video	
Test URL	<a href="https://todolist.james.am/#/">https://todolist.james.am/#/</a>

# Conclusion and What's next

# Key Takeaway

# ความรู้พื้นฐานในการทำ Manual Test

- Manual Test คือ การทดสอบที่ผู้ทดสอบจะทำการทดสอบซอฟต์แวร์ หรือระบบตามทดสอบ หรือกรณีทดสอบ โดยไม่ได้ใช้เครื่องมืออัตโนมัติใดๆ
- กระบวนการของ Manual Testing
  - ทำความเข้าใจความต้องการของ Business (Business requirement)
  - การทำแผนงานการทดสอบ (Test planning)
  - การสร้างทดสอบ (Test case development)
  - การดำเนินการทดสอบ (Test execution)
  - การรายงานผลการทดสอบและการติดตามสถานะของ Bug และ Defect
  - การทดสอบซ้ำและการทำ regression testing

- **เทคนิคการทำ Manual testing**

- **Black box testing**
- **White box testing**
- **Grey box testing**
- **Exploratory testing**

- **การบันทึก และการจัดการ Bug**

- **Severity และ Priority**
- **รายละเอียดในการบันทึก Bug**

# สิ่งที่ควรเรียนรู้เพิ่มเติมต่อจากนี้

---

- การทำ API Testing เบื้องต้น และการทดสอบการใช้งาน (Functional) ตั้งแต่ต้นจนจบ