

กิจกรรม สัปดาห์ที่ 2: คุณภาพซอฟต์แวร์ และ เมตริกซ์คุณภาพ

วัตถุประสงค์ของกิจกรรม

เมื่อสิ้นสุดกิจกรรมนี้ นักศึกษาจะสามารถ:

- ประเมินระบบซอฟต์แวร์โดยใช้มาตรฐาน ISO 25010
- คำนวณเมตริกซ์คุณภาพสำคัญจากข้อมูลโครงการจริง
- สร้างแดชบอร์ด (Quality Dashboard) โดยใช้ Google Sheets
- วิเคราะห์เมตริกซ์และให้ข้อเสนอแนะ
- เขียนรายงานการประเมินคุณภาพเชิงวิชาชีพ

ข้อกำหนดเบื้องต้น

ก่อนเริ่มกิจกรรมนี้ โปรดตรวจสอบว่าคุณมี:

- เข้าถึงระบบห้องสมุด (Library Management System) เวอร์ชัน buggy
- ข้อบกพร่อง (Bugs) ที่พบจากกิจกรรมสัปดาห์ที่ 1 (รายงานข้อบกพร่องของคุณ)
- Google Sheets หรือ Microsoft Excel
- เครื่องคิดเลข (หรือใช้สูตรสเปรดชีต)
- เครื่องมือบันทึกข้อมูล

กิจกรรมที่ 1: การประเมินคุณภาพตามมาตรฐาน ISO 25010

วัตถุประสงค์

ประเมินระบบห้องสมุดเทียบกับ 6 ลักษณะสำคัญของมาตรฐาน ISO 25010

คำแนะนำ

ขั้นตอนที่ 1: การทดสอบระบบ (5 นาที)

เข้าถึงระบบห้องสมุดและดำเนินการตามงานเหล่านี้:

- ลงทะเบียนผู้ใช้ใหม่
- เข้าสู่ระบบ (Login)
- ค้นหาหนังสือ
- ดูรายละเอียดหนังสือ
- ยืมหนังสือ
- ดูหนังสือของฉัน
- คืนหนังสือ
- พยายามยืมหนังสือที่ยืมแล้ว
- ค้นหาด้วยอักขระไม่ถูกต้อง
- ทดสอบเวลาตอบสนอง (Response Time) ของระบบ

ขั้นตอนที่ 2: การประเมินคุณภาพ (15 นาที)

สำหรับแต่ละลักษณะคุณภาพ ให้ให้คะแนนระบบในระดับ 1-5:

- 1 = แย่มาก (Very Poor)
- 2 = แย่ (Poor)
- 3 = ยอมรับได้ (Acceptable)
- 4 = ดี (Good)
- 5 = ยอดเยี่ยม (Excellent)

กรอกข้อมูลในตารางการประเมินด้านล่าง:

แม่แบบการประเมิน

1. Functional Suitability (ความเหมาะสมของฟังก์ชัน)

เกณฑ์การประเมิน (Evaluation Criteria):

- มีฟีเจอร์ (Features) ที่จำเป็นทั้งหมด
- ฟีเจอร์ใช้งานได้อย่างถูกต้อง
- ฟีเจอร์ให้ผลลัพธ์ที่ถูกต้อง

ผลการทดสอบ (Test Results):

Feature: User Registration (การลงทะเบียนผู้ใช้)

Test: Create account with valid data (สร้างบัญชีด้วยข้อมูลที่ถูกต้อง)

Result: Pass Fail

Issue found: _____

Feature: Book Search (การค้นหาหนังสือ)

Test: Search by title (ค้นหาตามชื่อ)

Result: Pass Fail

Issue found: _____

Feature: Book Borrowing (การยืมหนังสือ)

Test: Borrow available book (ยืมหนังสือที่มีอยู่)

Result: Pass Fail

Issue found: _____

คะแนน (Rating): ___ / 5

เหตุผล (Justification):

2. Performance Efficiency (ประสิทธิภาพ)

เกณฑ์การประเมิน:

- [] เวลาตอบสนองยอมรับได้ (< 3 วินาที)
- [] ระบบสามารถจัดการผู้ใช้พร้อมกัน (Concurrent Users)
- [] การใช้ทรัพยากร (Resource Usage) สมเหตุสมผล

ผลการทดสอบ:

Test: Search response time (เวลาตอบสนองการค้นหา)

Measurement: _____ seconds

Expected: < 3 seconds

Status: Pass Fail

Test: Page load time (เวลาโหลดหน้า)

Measurement: _____ seconds

Expected: < 2 seconds

Status: Pass Fail

Test: Database query time (เวลาค้นหาฐานข้อมูล)

Measurement: _____ seconds

Expected: < 1 second

Status: Pass Fail

คะแนน: __ / 5

เหตุผล:

3. Usability (ความง่ายในการใช้งาน)

เกณฑ์การประเมิน:

- [] อินเทอร์เฟซ (Interface) เข้าใจง่าย
- [] ข้อความข้อผิดพลาด (Error Messages) ชัดเจน
- [] การนำทาง (Navigation) ง่าย
- [] ความช่วยเหลือ/เอกสาร (Help/Documentation) พร้อมใช้

ผลการทดสอบ:

Test: First-time user can register without help (ผู้ใช้ครั้งแรกสามารถลงทะเบียนโดยไม่ต้องขอความช่วยเหลือ)

Result: Pass Fail

Time taken: _____ minutes

Test: User can find and borrow a book within 2 minutes (ผู้ใช้สามารถค้นหาและยืมหนังสือได้ภายในเวลา 2 นาที)

Result: Pass Fail

Time taken: _____ minutes

Test: Error messages are helpful (ข้อความข้อผิดพลาดช่วยเหลือได้)

Example error message: _____

Clear? Yes No

Test: Interface is visually appealing (อินเทอร์เฟซดูสวยงาม)

Rating: Good Acceptable Poor

คะแนน: __ / 5

เหตุผล:

4. Reliability (ความน่าเชื่อถือ)

เกณฑ์การประเมิน:

- ระบบไม่ขัดข้อง (Crash) ระหว่างการใช้งานปกติ
- ข้อมูล (Data) ได้รับการเก็บรักษาถูกต้อง
- ระบบกู้คืน (Recover) จากข้อผิดพลาดได้อย่างสวยงาม

ผลการทดสอบ:

Test: System stability during normal usage (เสถียรภาพของระบบระหว่างการใช้งานปกติ)

Number of crashes in 10 operations: _____

Expected: 0

Status: Pass Fail

Test: Data persistence (ความคงทนของข้อมูล)

Create book → Logout → Login → Book still exists? (สร้างหนังสือ → ออกจากระบบ → เข้าสู่ระบบ → หนังสือยังคงอยู่?)

Result: Pass Fail

Test: Error handling (การจัดการข้อผิดพลาด)

Trigger error (e.g., invalid input) (เรียกข้อผิดพลาด เช่น ข้อมูลที่ไม่ถูกต้อง)

System crashes? Yes No

Shows error message? Yes No

คะแนน: __ / 5

เหตุผล:

5. Security (ความปลอดภัย)

เกณฑ์การประเมิน:

- รหัสผ่าน (Passwords) ได้รับการป้องกัน
- ต้องมีการตรวจสอบสิทธิ์ (Authentication) สำหรับการกระทำที่ป้องกัน
- ข้อมูลผู้ใช้ (User Data) ได้รับการป้องกัน

ผลการทดสอบ:

Test: Password visibility (การมองเห็นรหัสผ่าน)

Passwords visible in plain text? (รหัสผ่านมองเห็นเป็นข้อความธรรมดา?) Yes No

Passwords masked in UI? (รหัสผ่านถูกซ่อนในอินเทอร์เฟซ?) Yes No

Test: Authorization (การอนุญาต)

Can non-logged-in user borrow books? (ผู้ใช้ที่ไม่ได้เข้าสู่ระบบสามารถยืมหนังสือได้?) Yes No

Can user A access user B's borrowed books? (ผู้ใช้ A สามารถเข้าถึงหนังสือที่ยืมของผู้ใช้ B ได้?) Yes No

Test: Basic SQL Injection

Try: ' OR '1'='1 in search

System crashes? Yes No

Returns inappropriate data? (คืนข้อมูลที่ไม่เหมาะสม?) Yes No

คะแนน: __ / 5

เหตุผล:

6. Maintainability (ความสามารถในการบำรุงรักษา)

เกณฑ์การประเมิน:

- รหัส (Code) มีการจัดระเบียบที่ดี
- รหัสมีความเห็น (Comments)
- รหัสทำตามรูปแบบ (Style) ที่สอดคล้องกัน
- ฟังก์ชัน (Functions) มีขนาดที่เหมาะสม

รายการตรวจสอบการตรวจสอบรหัส (Code Review Checklist):

ตรวจสอบไฟล์รหัสต้นฉบับ:

- Code is organized into modules/files logically (รหัสจัดระเบียบเป็นโมดูล/ไฟล์อย่างมีเหตุผล)
- Functions/classes have descriptive names (ฟังก์ชัน/คลาสมีชื่อที่อธิบายได้)
- Code has comments explaining complex logic (รหัสมีความเห็นอธิบายตรรกะที่ซับซ้อน)
- Consistent naming convention (camelCase, snake_case, etc.) (ชื่อสอดคล้องกัน)
- No functions longer than 50 lines (ไม่มีฟังก์ชันที่ยาวกว่า 50 บรรทัด)
- No duplicate code (DRY principle followed) (ไม่มีโค้ดซ้ำ ปฏิบัติตาม DRY)

Number of code quality issues found: _____

คะแนน: ___ / 5

เหตุผล:

สรุปการประเมิน

คะแนนคุณภาพโดยรวม (Overall Quality Score):

Average Rating = (Sum of all 6 ratings) / 6 = _____ / 5

Interpretation (การตีความ):

4.0 - 5.0 = Excellent Quality (คุณภาพยอดเยี่ยม)

3.0 - 3.9 = Good Quality (คุณภาพดี)

2.0 - 2.9 = Acceptable Quality (ยอมรับได้แต่ต้องปรับปรุง)

1.0 - 1.9 = Poor Quality (คุณภาพแย่ มีปัญหาใหญ่)

จุดแข็ง 3 อันดับแรก (Top 3 Strengths):

- _____
- _____
- _____

พื้นที่ที่ต้องปรับปรุง 3 อันดับแรก (Top 3 Areas for Improvement):

- _____
- _____
- _____

กิจกรรมที่ 2: การคำนวณเมตริกซ์คุณภาพ

วัตถุประสงค์

คำนวณเมตริกซ์คุณภาพสำคัญจากข้อมูลโครงการและตีความผลลัพธ์

ข้อมูลที่กำหนด: ระบบห้องสมุด

INFORMATION (ข้อมูลโครงการ):

- Project Name: Library Management
- Version: 1.0
- Team: x developers, x testers
- Duration: 8 weeks (2 months)

CODE METRICS (เมตริกซอร์ส):

- Total Lines of Code: 12,500 LOC
- Number of Functions: 285
- Number of Classes: 42
- Number of Modules: 15

TEST DATA (ข้อมูลการทดสอบ from Week 1 and current week):

- Total Test Cases Written: 150
- Total Test Cases Executed: 150
- Test Cases Passed: 135
- Test Cases Failed: 12
- Test Cases Blocked: 3

DEFECT DATA (ข้อมูลข้อบกพร่อง):

Phase 1: Development & Unit Testing

- Defects found by developers: 18 defects
 - * Severity: 2 Critical, 8 Major, 8 Minor

Phase 2: Integration Testing

- Defects found by testers: 22 defects

* Severity: 1 Critical, 12 Major, 9 Minor

Phase 3: System Testing

- Defects found by testers: 15 defects

* Severity: 0 Critical, 8 Major, 7 Minor

Phase 4: User Acceptance Testing (UAT)

- Defects found by users: 8 defects

* Severity: 0 Critical, 3 Major, 5 Minor

Phase 5: Production (First 30 days after release)

- Defects reported by end users: 5 defects

* Severity: 1 Critical, 2 Major, 2 Minor

TOTAL DEFECTS: 68 defects

CODE COVERAGE (from Jest):

- Statement Coverage: 82%

- Branch Coverage: 75%

- Function Coverage: 88%

- Line Coverage: 81%

TIME TRACKING (การติดตามเวลา):

- Total Development Time: 320 hours

- Total Testing Time: 160 hours

- Total Debugging Time: 80 hours

- Total Test Execution Time: 40 hours (spread across 10 working days)

ขั้นตอนที่ 1: คำนวณ Defect Density

สูตร (Formula):

Defect Density = Total Defects / Size (in KLOC)

การคำนวณ (Calculation):

Total Defects = 68

Size = 12,500 LOC = 12.5 KLOC

Defect Density = $68 / 12.5 = \underline{\hspace{2cm}}$ defects/KLOC

การตีความ (Interpretation):

Your Result: _____ defects/KLOC

Industry Benchmark (มาตรฐานอุตสาหกรรม):

- 0-1 = Excellent (ยอดเยี่ยม)
- 1-5 = Good (ดี)
- 5-10 = Average (ปานกลาง)
- 10+ = Poor (แย่)

Your Assessment: _____

ขั้นตอนที่ 2: คำนวณ Test Coverage

สูตร:

Average Coverage = (Statement + Branch + Function + Line) / 4

การคำนวณ:

Statement Coverage: 82%

Branch Coverage: 75%

Function Coverage: 88%

Line Coverage: 81%

Average Coverage = (82 + 75 + 88 + 81) / 4 = _____ %

การตีความ:

Your Result: _____ %

Industry Target (เป้าหมายอุตสาหกรรม):

- 90%+ = Excellent (อาจทดสอบมากเกินไป)
- 80-90% = Very Good (ช่วงที่เหมาะสม)
- 70-80% = Good (ดี)
- 60-70% = Acceptable (ยอมรับได้)
- <60% = Poor (แย่มาก)

Your Assessment: _____

คำถาม (Question to Consider): ทำไม Branch Coverage (75%) ต่ำกว่าประเภทอื่น ๆ? สิ่งนี้บอกเราอะไรเกี่ยวกับคุณภาพของการทดสอบของเรา?

Answer: _

ขั้นตอนที่ 3: คำนวณ Defect Removal Efficiency (DRE)

สูตร:

$$DRE = (\text{Defects Found Before Release} / \text{Total Defects}) \times 100\%$$

การคำนวณ:

$$\text{Defects Before Release} = 18 + 22 + 15 + 8 = \underline{\quad\quad} \text{ defects}$$

$$\text{Defects After Release} = 5 \text{ defects}$$

$$\text{Total Defects} = \underline{\quad\quad} + 5 = \underline{\quad\quad} \text{ defects}$$

$$DRE = (\underline{\quad\quad} / \underline{\quad\quad}) \times 100\% = \underline{\quad\quad} \%$$

การตีความ:

Your Result: %

Industry Benchmark (มาตรฐานอุตสาหกรรม):

95-99% = Excellent (ยอดเยี่ยม)

85-95% = Good (ดี)

70-85% = Acceptable (ยอมรับได้)

<70% = Poor (แย่)

Your Assessment: _____

คำถาม (Question to Consider): เราสามารถปรับปรุง DRE ได้อย่างไร?

Answer: _

ขั้นตอนที่ 4: คำนวณ Test Pass Rate

สูตร:

Test Pass Rate = (Tests Passed / Total Tests Executed) × 100%

การคำนวณ:

Tests Passed = 135

Total Tests Executed = 150

Test Pass Rate = (135 / 150) × 100% = _____ %

การตีความ:

Your Result: _____ %

Assessment Guide (คู่มือการประเมิน):

- 95-100% = System is very stable, ready for release (ระบบเสถียรมากพร้อมเปิดตัว)
- 85-95% = System is stable, minor issues (ระบบเสถียรมีปัญหาน้อย)
- 70-85% = System has issues, needs work (ระบบมีปัญหาต้องการการแก้ไข)
- <70% = System is unstable, not ready (ระบบไม่เสถียรไม่พร้อม)

Your Assessment: _____

ขั้นตอนที่ 5: คำนวณ Test Execution Rate

สูตร:

Test Execution Rate = Total Tests Executed / Time (in days)

การคำนวณ:

Total Tests Executed = 150

Time = 10 working days

Test Execution Rate = 150 / 10 = _____ tests/day

การตีความ:

Your Result: _____ tests/day

This metric helps with (เมตริกซ์นี้ช่วยในการ):

- Planning future test schedules (วางแผนตารางการทดสอบในอนาคต)

- Estimating project timelines (ประมาณการระยะเวลาโครงการ)
- Identifying process bottlenecks (ระบุคอขวดของกระบวนการ)

If we need to test 300 test cases, we would need (หากเราต้องการทดสอบ 300 กรณีการทดสอบ เราจะต้องการ):
 300 / _____ tests/day = _____ days

ขั้นตอนที่ 6: วิเคราะห์การกระจายข้อบกพร่อง

โดยขั้นตอนการทดสอบ (By Test Phase):

Phase	Defects	Percentage
Development/Unit	18	_____ %
Integration	22	_____ %
System Testing	15	_____ %
UAT	8	_____ %
Production	5	_____ %
TOTAL	68	100%

คำนวณเปอร์เซ็นต์ (Calculate Percentages):

Development: $(18/68) \times 100\% =$ _____ %
 Integration: $(22/68) \times 100\% =$ _____ %
 System Test: $(15/68) \times 100\% =$ _____ %
 UAT: $(8/68) \times 100\% =$ _____ %
 Production: $(5/68) \times 100\% =$ _____ %

ตามระดับความร้ายแรง (By Severity):

Severity	Defects	Percentage
Critical	4	_____ %
Major	33	_____ %
Minor	31	_____ %
TOTAL	68	100%

คำนวณเปอร์เซ็นต์:

Critical: $(4/68) \times 100\% =$ _____ %
 Major: $(33/68) \times 100\% =$ _____ %
 Minor: $(31/68) \times 100\% =$ _____ %

คำถามวิเคราะห์ (Analysis Questions):

1. ขั้นตอนใดค้นหาข้อบกพร่องมากที่สุด? ทำไมถึงเป็นเช่นนี้? Answer: ____
2. การกระจายของข้อบกพร่องข้ามขั้นตอนมีความเป็นสุภาพดีหรือไม่? (Ideally: ตรวจสอบข้อบกพร่องมากกว่าในขั้นตอนแรก น้อยกว่าในขั้นตอนต่อมา) Answer: ____
3. เรามีข้อบกพร่องที่สำคัญ (Critical Defects) ใน Production หรือไม่? สิ่งนี้หมายถึงอะไร? Answer: ____

ขั้นตอนที่ 7: คำนวณ Cost Metrics

สูตร (Formula):

Cost per Defect = Total Testing Cost / Number of Defects Found

ข้อมูลที่กำหนด (Given):

Tester hourly rate: \$40/hour

Total testing time: 160 hours

Total Testing Cost = 160 × \$40 = \$6,400

การคำนวณ:

Defects found in testing (before production) = 63 defects

Cost per Defect = \$6,400 / 63 = \$_____ per defect

การวิเคราะห์ ROI (ROI Analysis):

Cost to find defect in testing: \$_____

Cost to fix defect in production: \$_____ × 10 = \$_____

(Using 1-10-100 rule (กฎ 1-10-100))

Money saved by finding 63 defects in testing:

Savings = 63 × (\$_____ - \$_____) = \$_____

การตีความ: การทดสอบคุ้มค่ากับการลงทุนหรือไม่? ____

กิจกรรมที่ 3: สร้างแดชบอร์ดคุณภาพ

วัตถุประสงค์

สร้างแดชบอร์ด (Dashboard) ที่สรุปเมตริกซ์คุณภาพทั้งหมดด้วยภาพ

คำแนะนำ

ขั้นตอนที่ 1: เปิด Google Sheets หรือ Excel

ขั้นตอนที่ 2: สร้างโครงสร้างแดชบอร์ด

คัดลอกแม่แบบนี้ลงในสเปรดชีต:

LIBRARY MANAGEMENT
QUALITY METRICS DASHBOARD
Version 1.0

PROJECT OVERVIEW

Team Size: 7 (5 dev + 2 QA)
Duration: 8 weeks
Lines of Code: 12,500
Total Test Cases: 150
Total Defects: 68

KEY METRICS

Metric	Value	Target	Status
Defect Density	___/KLOC	<5	___
Test Coverage	___%	>80%	___
DRE	___%	>85%	___
Test Pass Rate	___%	>90%	___
Test Execution Rate	___/day	-	INFO

DEFECT DISTRIBUTION BY PHASE

Phase	Count	Percentage	[Bar]
Development	18	___%	██████████
Integration	22	___%	██████████
System Testing	15	___%	██████████
UAT	8	___%	██████
Production	5	___%	████

DEFECT DISTRIBUTION BY SEVERITY (การกระจายตามระดับความร้ายแรง)

Severity	Count	Percentage	[Bar]
Critical	4	___%	████
Major	33	___%	██████████
Minor	31	___%	██████████

ISO 25010 QUALITY ASSESSMENT (การประเมินคุณภาพตามมาตรฐาน)

Characteristic	Rating (1-5)	[Visual]
Functional Suitability	___	★★★★☆
Performance Efficiency	___	★★★★☆
Usability	___	★★★★☆
Reliability	___	★★★★☆
Security	___	★★★★☆
Maintainability	___	★★★★☆
Average Quality Score	___/5	

OVERALL ASSESSMENT	
Quality Level:	<input type="checkbox"/> Excellent <input type="checkbox"/> Good <input type="checkbox"/> Acceptable <input type="checkbox"/> Poor
Release Ready?	<input type="checkbox"/> YES <input type="checkbox"/> NO <input type="checkbox"/> CONDITIONAL
Confidence Level:	<input type="checkbox"/> High <input type="checkbox"/> Medium <input type="checkbox"/> Low

TOP 3 RISKS
1. _____
2. _____
3. _____

RECOMMENDATIONS
1. _____
2. _____
3. _____
4. _____

ขั้นตอนที่ 3: กรอกค่าที่คำนวณ (Fill in Your Calculated Values)

โอนค่าทั้งหมดที่คำนวณในกิจกรรมที่ 2 ลงในแดชบอร์ด

ขั้นตอนที่ 4: สร้างแผนภูมิ (Create Charts) (ทางเลือก แต่แนะนำ)

สร้างแผนภูมิเหล่านี้ในสเปรดชีต:

- Pie Chart (แผนภูมิวงกลม):** Defect Distribution by Phase
- Bar Chart (แผนภูมิแท่ง):** Defect Distribution by Severity
- Radar Chart (แผนภูมิเรดาร์):** ISO 25010 Quality Assessment
- Line Chart (แผนภูมิเส้น):** Defects Found per Week (ถ้ามีข้อมูลรายสัปดาห์)

ขั้นตอนที่ 5: ฟอแมตเพื่อให้อ่านง่าย (Format for Readability)

- ใช้สี: เขียวสำหรับดี เหลืองสำหรับยอมรับได้ แดงสำหรับแย่
- ใช้ Conditional Formatting สำหรับตัวบ่งชี้สถานะ
- เพิ่มเส้นขอบและการจัดตำแหน่งที่เหมาะสม