

เอกสารฉบับนี้อธิบายเกี่ยวกับ ISO/IEC 25010 ซึ่งเป็นมาตรฐานสากลสำหรับแบบจำลองคุณภาพของซอฟต์แวร์

สารบัญ

- 1. บทนำ
- 2. ISO 25010 คืออะไร
- 3. แบบจำลองคุณภาพของผลิตภัณฑ์ (Product Quality Model)
- 4. แบบจำลองคุณภาพในการใช้งาน (Quality in Use Model)
- 5. ความสัมพันธ์ระหว่างแบบจำลอง
- 6. การนำไปใช้ในการทดสอบ
- 7. ตัวอย่างการประเมิน
- 8. เอกสารอ้างอิง

1. บทนำ

ประวัติศาสตร์

- ISO/IEC 9126 (2001): มาตรฐานเก่า - Software Quality Model
- ISO/IEC 25010 (2011): ปรับปรุง - System and Software Quality Model
- ISO/IEC 25010 (2023): ฉบับล่าสุด - เพิ่มลักษณะใหม่ และปรับปรุงคำจำกัดความ

เหตุผลในการสร้าง ISO 25010

- ให้มีมาตรฐานเดียวกันทั่วโลก
- ทำให้ประเมินคุณภาพได้อย่างเป็นวัตถุประสงค์
- ช่วยให้ผู้ค้นหา bugs ได้อย่างเป็นระบบ
- สนับสนุนการพัฒนาที่มีคุณภาพ

การใช้ในประเทศไทย

มาตรฐาน ผ. 2505 Part 1

- Information technology - System and software quality models
- เป็น National Standard ของประเทศไทย
- มีเนื้อหาเดียวกับ ISO/IEC 25010

2. ISO 25010 คืออะไร

นิยาม

ISO/IEC 25010 เป็นมาตรฐานสากลที่ระบุชุดแบบจำลองสำหรับประเมินคุณภาพของซอฟต์แวร์ โดยประกอบด้วยสอง

แบบจำลองหลัก:

- Product Quality Model - คุณลักษณะของผลิตภัณฑ์ซอฟต์แวร์
- Quality in Use Model - คุณภาพเมื่อผู้ใช้งานนำไปใช้งานจริง

หลักการพื้นฐาน

ด้าน	คำอธิบาย
คุณลักษณะ (Characteristics)	คุณลักษณะหลักของคุณภาพ (8 ด้าน)
คุณลักษณะย่อย (Sub-characteristics)	คุณลักษณะย่อย (20+ ด้าน)
การวัด (Measurement)	วิธีการวัดเพื่อประเมิน
ตัวบ่งชี้ (Indicators)	ตัวบ่งชี้สำหรับการประเมิน

3. แบบจำลองคุณภาพของผลิตภัณฑ์ (Product Quality Model)

ภาพรวม 8 คุณลักษณะหลัก

PRODUCT QUALITY (ISO 25010)

- Functionality - ความสมบูรณ์
- Reliability - ความน่าเชื่อถือ
- Usability - ความง่ายต่อการใช้
- Performance Efficiency - ประสิทธิภาพ
- Security - ความปลอดภัย
- Maintainability - ความดูแลรักษา
- Compatibility - ความเข้ากันได้
- Portability - ความพกพา

3.1 ความสมบูรณ์ (Functionality)

คำนิยาม

ซอฟต์แวร์มีฟีเจอร์และฟังก์ชันครบตามความต้องการหรือไม่

คุณลักษณะย่อย (Sub-characteristics)

- ความครบถ้วน (Completeness) - ครบถ้วนหรือไม่
- ความถูกต้อง (Correctness) - ถูกต้องหรือไม่
- ความเหมาะสม (Appropriateness) - เหมาะสมหรือไม่

วิธีการทดสอบ

- ทดสอบเชิงบวก (Positive testing): ทดสอบกับ input ที่ถูกต้อง
- ทดสอบเชิงลบ (Negative testing): ทดสอบกับ input ที่ผิด
- ทดสอบที่ขอบเขต (Boundary testing): ทดสอบที่ขอบเขตค่า

ตัวอย่าง (Library Management System)

Test Case: Add Book

- Input: Valid book data (ISBN, Title, Author, Year)
- Expected: Book created in database
- Result: PASS ✓

3.2 ความน่าเชื่อถือ (Reliability)

คำนิยาม

ซอฟต์แวร์ทำงานต่อเนื่องโดยไม่เกิดข้อผิดพลาดหรือ crash

คุณลักษณะย่อย (Sub-characteristics)

1. ความสมบูรณ์ (Maturity) - ระบบเสถียรหรือไม่
2. ความทนทานต่อข้อผิดพลาด (Fault Tolerance) - ป้องกันความผิดพลาดได้หรือไม่
3. ความสามารถในการฟื้นตัว (Recoverability) - สามารถฟื้นตัวจากข้อผิดพลาดได้หรือไม่

วิธีการทดสอบ

- การทดสอบความเค้น (Stress testing): ให้ระบบทำงานหนัก
- การทดสอบการโหลด (Load testing): ให้ load มากขึ้น
- การทดสอบแบบ Soak (Soak testing): ให้ทำงานต่อเนื่องนานๆ
- การทดสอบการฟื้นตัว (Recovery testing): ทำให้เกิด error แล้วดูการฟื้นตัว

ตัวอย่าง

Test Case: System Stability

- Scenario: 1000 concurrent users login
- Expected: No crash, response time < 5 sec
- Result: System crashes at 500 users X

3.3 ความง่ายต่อการใช้ (Usability)

คำนิยาม

ผู้ใช้สามารถเข้าใจและใช้ระบบได้อย่างมีประสิทธิภาพ

คุณลักษณะย่อย (Sub-characteristics)

1. ความการจำแนก (Recognizability) - ชัดเจนหรือไม่ว่านี่คือระบบอะไร
2. ความสามารถในการเรียนรู้ (Learnability) - เรียนรู้ได้ง่ายหรือไม่
3. ความสมบูรณ์ (Operability) - ใช้งานง่ายหรือไม่
4. การป้องกัน User Error (User Error Protection) - ป้องกัน user error ได้หรือไม่
5. อากศินส่วนติดต่อ (User Interface Aesthetics) - ออกแบบสวยหรือไม่
6. การเข้าถึง (Accessibility) - ผู้พิการสามารถใช้ได้หรือไม่

วิธีการทดสอบ

- การทดสอบผู้ใช้ (User testing): ให้ user ทั่วไปใช้ดู
- การทดสอบความง่ายต่อการใช้ (Usability testing): สังเกตพฤติกรรมการใช้
- การทดสอบเปรียบเทียบ (A/B testing): เปรียบเทียบ UI สองแบบ
- การทดสอบความคมชัด (Accessibility testing): ทดสอบกับ assistive technology

ตัวอย่าง

Test Case: Navigation

- User task: Find and borrow a book
- Expected: Complete in < 2 minutes
- User feedback: "I don't know where to click" X

3.4 ประสิทธิภาพ (Performance Efficiency)

คำนิยาม

ระบบตอบสนองได้รวดเร็วและใช้ทรัพยากรอย่างมีประสิทธิภาพ

คุณลักษณะย่อย (Sub-characteristics)

1. พฤติกรรมการกำหนดเวลา (Time Behavior) - ตอบสนองเร็วไหมหรือไม่
2. การใช้ทรัพยากร (Resource Utilization) - ใช้ CPU/Memory อย่างเหมาะสมหรือไม่

วิธีการทดสอบ

- การวัดเวลาการตอบสนอง (Response time testing): วัดเวลาการตอบสนอง
- การทดสอบหรือทศาคีว (Throughput testing): วัดจำนวน transactions ต่อวินาที
- การกำหนดทรัพยากร (Resource monitoring): ติดตาม CPU, Memory
- Profiling: หา bottleneck ของ code

ตัวอย่าง

Test Case: Search Performance

- Action: Search book with 10,000 records in database
- Expected: Results within 2 seconds
- Result: Takes 15 seconds X
- Root Cause: Missing database index

3.5 ความปลอดภัย (Security)

คำนิยาม

ระบบปกป้องข้อมูลจากการเข้าถึงและการใช้งานโดยไม่ได้รับอนุญาต

Sub-characteristics

1. Confidentiality - ข้อมูลเป็นความลับ
2. Integrity - ข้อมูลไม่ถูกแก้ไขโดยไม่ได้รับอนุญาต
3. Non-repudiation - ไม่สามารถปฏิเสธการกระทำได้
4. Accountability - สามารถติดตามความรับผิดชอบได้
5. Authenticity - ยืนยันตัวตนได้

วิธีการทดสอบ

- Penetration testing: ลองโจมตี system
- SQL Injection testing: ทดสอบ SQL Injection

- XSS testing: ทดสอบ Cross-Site Scripting
- Authentication testing: ทดสอบระบบ login
- Authorization testing: ทดสอบการให้สิทธิ์

ตัวอย่าง

Test Case: SQL Injection Prevention

- Input: admin' OR '1'='1 (SQL Injection)
- Expected: Login rejected, error message
- Result: Login accepted **X**
- Severity: Critical

3.6 ความดูแลรักษา (Maintainability)

คำนิยาม

ซอฟต์แวร์สามารถแก้ไข ปรับปรุง หรือเพิ่มฟีเจอร์ได้ง่าย

Sub-characteristics

1. **Modularity** - โปรแกรมแบ่งเป็น modules ที่อิสระ
2. **Reusability** - สามารถนำ code มาใช้ซ้ำได้
3. **Analysability** - วิเคราะห์หา bugs ได้ง่าย
4. **Modifiability** - แก้ไขได้ง่ายโดยไม่ส่งผลกระทบต่อ
5. **Testability** - ทดสอบได้ง่าย

วิธีการทดสอบ

- Code review: ตรวจสอบคุณภาพ code
- Static analysis: ใช้เครื่องมือตรวจสอบ
- Complexity analysis: วัด cyclomatic complexity
- Documentation review: ตรวจสอบเอกสาร

ตัวอย่าง

Code Review Issue:

- Function: borrow_book() has 500 lines
- Problem: Too long, hard to understand
- Recommendation: Split into smaller functions

3.7 ความเข้ากันได้ (Compatibility)

คำนิยาม

ซอฟต์แวร์ทำงานบน environments ต่างๆ

Sub-characteristics

1. **Co-existence** - ทำงานร่วมกับ software อื่นได้

2. Interoperability - สามารถแลกเปลี่ยนข้อมูลกับระบบอื่น

วิธีการทดสอบ

- Cross-browser testing: ทดสอบบน browsers ต่างๆ
- Cross-platform testing: ทดสอบบน OS ต่างๆ
- API testing: ทดสอบความเข้ากันได้ของ API
- Data format testing: ทดสอบการอ่าน/เขียนข้อมูล

ตัวอย่าง

Test Case: Cross-browser Compatibility

- Test on: Chrome, Firefox, Safari, Edge

- Expected: All functions work

- Result:

- Chrome ✓

- Firefox ✓

- Safari: Button misaligned ✗

3.8 ความพกพา (Portability)

คำนิยาม

ซอฟต์แวร์สามารถย้ายไปยัง environment อื่นได้ง่าย

Sub-characteristics

1. **Adaptability** - ปรับเปลี่ยนได้
2. **Installability** - ติดตั้งได้ง่าย
3. **Replaceability** - แทนที่ได้

วิธีการทดสอบ

- Installation testing: ทดสอบการติดตั้ง
- Uninstallation testing: ทดสอบการถอดถอน
- Migration testing: ทดสอบการย้าย
- Docker testing: ทดสอบ containerization

ตัวอย่าง

Test Case: Docker Deployment

- Build: Docker image successfully

- Run: Container starts without error

- Verify: Application accessible on port 8080

- Result: PASS ✓

4. แบบจำลองคุณภาพในการใช้งาน (Quality in Use Model)

ภาพรวม 5 คุณลักษณะ

QUALITY IN USE (ผลการใช้จริง)

1. ประสิทธิภาพ (Effectiveness) - ประสิทธิภาพ
2. ประสิทธิภาพ (Efficiency) - ประสิทธิภาพ
3. ความพอใจ (Satisfaction) - ความพอใจ
4. การลดความเสี่ยง (Risk Mitigation) - ลดความเสี่ยง
5. ความเข้าถึง (Accessibility) - การเข้าถึง

4.1 ประสิทธิภาพ (Effectiveness)

คำนิยาม

ผู้ใช้บรรลุเป้าหมายได้อย่างถูกต้องและสมบูรณ์

การวัด

- Success rate: กี่เปอร์เซ็นต์ของผู้ใช้ที่สำเร็จ
- Error rate: มี error ไหม
- Completion rate: ทำเสร็จหรือไม่

ตัวอย่าง

Metric: Borrow Book Success Rate

- Target: 95%
- Actual: 87%
- Problem: User couldn't find book
- Action: Improve search function

4.2 ประสิทธิภาพ (Efficiency)

คำนิยาม

ผู้ใช้ทรัพยากร (เวลา, ความพยายาม) เท่าไรในการบรรลุเป้าหมาย

การวัด

- Time to complete task
- Number of steps
- Number of errors made
- Number of times help needed

ตัวอย่าง

Metric: Time to Return Book

- Target: < 2 minutes
- Actual: 4 minutes
- Problem: Process too complicated

- Action: Simplify workflow

4.3 ความพอใจ (Satisfaction)

คำนิยาม

ผู้ใช้พอใจกับการใช้ระบบหรือไม่

การวัด

- User satisfaction survey
- Net Promoter Score (NPS)
- System Usability Scale (SUS)
- Rating scores (1-5 stars)

ตัวอย่าง

Survey Question:

"How satisfied are you with the system?"

- Very satisfied: 40%

- Satisfied: 35%

- Neutral: 15%

- Dissatisfied: 10%

- Result: 75% satisfied

4.4 การลดความเสี่ยง (Risk Mitigation)

คำนิยาม

ระบบลดความเสี่ยงที่ไม่พึงประสงค์ได้หรือไม่

ตัวอย่างความเสี่ยง

- Data loss: หายข้อมูล
- Security breach: รั่วไหลข้อมูลส่วนตัว
- Financial loss: สูญเสียเงิน
- Reputation damage: เสื่อมเสียชื่อเสียง

การวัด

- Backup and recovery: มี backup ใหม่
 - Security measures: มีมาตรการป้องกัน
 - Audit trails: สามารถตรวจสอบได้
 - Incident response: มีแผนรับมือเหตุการณ์
-

4.5 ความคหบแแจก (Accessibility)

คำนิยาม

ผู้คนทั้งหมด (รวมทั้งผู้พิการ) สามารถใช้ระบบได้

ประเภท Disabilities

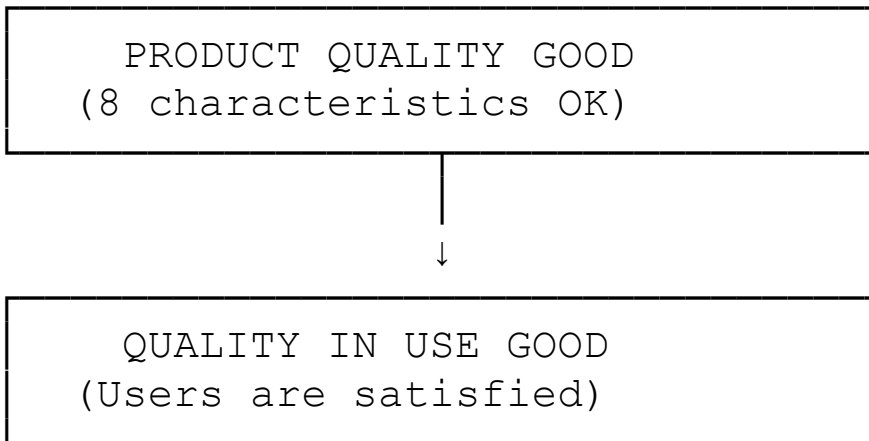
- Visual: ตาบอด
- Hearing: หูหนวก, หูหนวกบาง
- Motor: อัมพาต
- Cognitive: ความบกพร่องทางสติปัญญา

วิธีการทดสอบ

- Screen reader testing
- Keyboard navigation
- Color contrast checking
- Font size testing
- Captions for audio

5. ความสัมพันธ์ระหว่างแบบจำลอง

Product Quality → Quality in Use



ตัวอย่าง

Product Quality	↓	Quality in Use
Functionality OK	↓	Users achieve goals (Effectiveness)
Performance Good	↓	Users complete fast (Efficiency)
Usability OK	↓	Users are happy (Satisfaction)
Security Good	↓	Users feel safe (Risk Mitigation)
Compatibility OK	↓	Works for everyone (Accessibility)

6. การนำไปใช้ในการทดสอบ (Testing Application)

ขั้นตอนการใช้ ISO 25010 ในการทดสอบ

ขั้นที่ 1: วางแผนการทดสอบ (Test Planning)

1. ระบุคุณลักษณะสำคัญ (Priority)

- Critical: Security, Reliability
- Important: Functionality, Performance
- Nice-to-have: Usability

2. ตั้งเป้าหมาย (Goals)

- Security: 0 critical vulnerabilities
- Performance: Response time < 2 sec
- Usability: SUS score > 70

ขั้นที่ 2: ออกแบบ Test Cases

สำหรับแต่ละคุณลักษณะ:

1. เลือก sub-characteristics
2. ตัดสินใจ test approach
3. สร้าง test cases
4. กำหนด acceptance criteria

ขั้นที่ 3: ดำเนินการทดสอบ (Execution)

1. รันทั้ง manual และ automated tests
2. บันทึกผลลัพธ์
3. บันทึก defects ที่พบ

ขั้นที่ 4: ประเมินผล (Evaluation)

1. วิเคราะห์ผลลัพธ์
2. เปรียบเทียบกับ acceptance criteria
3. สรุปคุณภาพโดยรวม

ตัวอย่าง Test Plan สำหรับ Library Management System

1. Functionality Testing

Characteristic: Functionality

Sub-characteristics:

- Completeness: ทดสอบว่ามีฟีเจอร์ครบหรือไม่
- Correctness: ทดสอบว่าทำงานถูกต้องหรือไม่
- Appropriateness: ทดสอบว่าเหมาะสมหรือไม่

Test Cases:

TC-F001: Add book with valid data → PASS

TC-F002: Search book by ISBN → PASS

TC-F003: Borrow book → PASS

TC-F004: Return book → PASS

Result: 4/4 PASS (100%)

2. Security Testing

Characteristic: Security

Sub-characteristics:

- Confidentiality
- Integrity
- Authenticity

Test Cases:

TC-S001: SQL Injection in login → BLOCKED ✓

TC-S002: XSS in book title → ESCAPED ✓

TC-S003: Access without authentication → REJECTED ✓

TC-S004: Password stored in plain text → HASHED ✓

Result: 4/4 PASS (100%)

3. Performance Testing

Characteristic: Performance Efficiency

Sub-characteristics:

- Time Behavior
- Resource Utilization

Test Cases:

TC-P001: Search with 10k records → 2 sec ✓

TC-P002: 100 concurrent users → No crash ✓

TC-P003: Database backup → No performance impact ✓

Result: 3/3 PASS (100%)

7. ตัวอย่างการประเมินคุณภาพ (Quality Assessment)

Quality Assessment Report

Project: Library Management System

Date: 2025-11-28

Evaluated by: QA Team

QUALITY CHARACTERISTICS EVALUATION			
Characteristic	Tests	Pass %	Status
Functionality	25	100%	PASS
Reliability	15	93%	FAIR
Usability	20	85%	FAIR
Performance	10	100%	PASS
Security	30	100%	PASS
Maintainability	8	80%	FAIR
Compatibility	12	100%	PASS
Portability	5	80%	FAIR
TOTAL	125	94%	PASS

SUMMARY:

- Overall Quality Score: 94/100
- Critical Issues: 0
- Major Issues: 3 (Reliability, Usability)
- Minor Issues: 8

RECOMMENDATION:

- ✓ APPROVED for production release
(with recommendations for improvement)

8. เอกสารอ้างอิง (References)

มาตรฐานอย่างเป็นทางการ

1. ISO/IEC 25010:2023
 - Title: System and software quality models
 - Publisher: International Organization for Standardization
 - Website: <https://www.iso.org/>
2. TIS 2505 Part 1
 - Thai National Standard

- ยืมจาก ISO/IEC 25010

เอกสารเพิ่มเติม

- [ISO/IEC 25040: Quality evaluation process](#)
- [ISO/IEC 25041: Guidance for developers](#)
- [ISO/IEC 25051: Quality for ready-made software](#)
- [ISO/IEC 25020: Measurement reference model](#)
- [ISTOB Glossary](#)
- [ISO Official](#)
- [Software Testing Help](#)
- [Guru99 - Software Quality](#)

สรุป

ประเด็นสำคัญ

1. ISO 25010 ให้มาตรฐานสากล 8 ด้านสำหรับประเมินคุณภาพ
2. Product Quality Model: ประเมินคุณลักษณะของ software
3. Quality in Use Model: ประเมินคุณภาพเมื่อผู้ใช้ใช้จริง
4. ใช้ ISO 25010 ในการวางแผนการทดสอบ
5. ประเมินผลลัพธ์อย่างเป็นวัตถุประสงค์

ขั้นตอนถัดไป

1. อ่านมาตรฐาน ISO/IEC 25010 ฉบับเต็ม
2. สร้าง Quality Metrics สำหรับโปรเจกต์
3. ออกแบบ Test Cases ตามแต่ละคุณลักษณะ
4. บันทึกผลการทดสอบอย่างเป็นระบบ
5. ติดตาม Quality Metrics ตลอดโปรเจกต์

Document Created: November 28, 2025

Version: 1.0

For Course: 88734365 - Software Testing and Evaluation

Language: Thai (ไทย)