

Agile Methodology

Scrum Framework and Kanban

วัตถุประสงค์การเรียนรู้

- ความเข้าใจ Agile: อธิบายหลักการและคุณค่าของ Agile Manifesto
- ความรู้เกี่ยวกับ Scrum: เข้าใจบทบาท เหตุการณ์ และสิ่งประดิษฐ์ใน Scrum Framework
- Kanban Knowledge: เข้าใจหลักการ Work In Progress (WIP) และ Flow ของ Kanban

Agile Manifesto และหลักการ

ทำไมต้อง Agile

- ปัญหาของ Waterfall
 - ไม่ตอบสนองต่อการเปลี่ยนแปลง
 - ผู้ใช้ต้องรอนาน กว่าที่จะได้เห็นผลิตภัณฑ์
 - ปัญหาพบ late cycle ทำให้แพงและช้า
- สถานการณ์ในปี 2000s
 - Web Development ที่เปลี่ยนแปลงเร็ว
 - Business ต้อง Quick Time-to-Market
 - ต้องวิธีการที่ยืดหยุ่นมากขึ้น

Agile Manifesto (2001)

<https://agilemanifesto.org/>

- 17 ผู้เชี่ยวชาญ ประกาศคำนิยม 4 ประการ บน Snowbird Resort ยูทาห์
- **คนและการมีปฏิสัมพันธ์กัน** มากกว่าการทำตามขั้นตอนและเครื่องมือ
 - คน > Tools
 - Communication > Documentation
- **ซอฟต์แวร์ที่นำไปใช้งานได้จริง** มากกว่าเอกสารที่ครบถ้วนสมบูรณ์
 - Running Code > Comprehensive Documentation
 - "Working Software is the primary measure of progress"
- **ร่วมมือทำงานกับลูกค้า** มากกว่าการต่อรองให้เป็นไปตามสัญญา
 - Customer Collaboration > Contract Negotiation
 - Customer Involvement > Fixed Requirements
- **การตอบรับกับการเปลี่ยนแปลง** มากกว่าการทำตามแผนที่วางไว้
 - Responding to Change > Following a Plan
 - Embrace Change

12 หลักการของ Agile

1. ความพึงพอใจลูกค้าสำคัญสูงสุด ผ่านการส่งซอฟต์แวร์ที่มีคุณค่า
2. ยอมรับการเปลี่ยนแปลงความต้องการ แม้ช่วงปลายของการพัฒนา
3. ส่งมอบซอฟต์แวร์ที่ทำงานได้จริง บ่อยๆ (สัปดาห์ หรือเดือน)
4. ตัวแทนจากฝ่ายธุรกิจและนักพัฒนา ทำงานร่วมกันทุกวัน
5. สมาชิกโครงการเข้าใจและมีจุดมุ่งหมายของโครงการร่วมกัน
6. ใช้การสนทนาแบบหน้าต่อหน้าเพื่อประสิทธิภาพและประสิทธิผลสูงสุด
7. ซอฟต์แวร์ที่ทำงานได้จริง = การวัด progress
8. ส่งเสริมการพัฒนาที่ยั่งยืน สามารถรักษาอัตราเร็วในการทำงานร่วมกันให้คงที่
9. ความเอาใจใส่ต่อความเป็นเลิศด้านเทคนิค
10. ความเรียบง่าย การทำงานอย่างพอเพียง นั้นสำคัญ
11. ทีมที่บริหารตัวเองได้
12. ทีมจะต้องย้อนกลับไปดูสิ่งที่ผ่านมาเพื่อปรับปรุงและเปลี่ยนแปลงพฤติกรรมของทีม

Agile vs Traditional

เจ้าระบบ	Traditional (Waterfall)	Agile
การวางแผน	ล่วงหน้า (ต้นโครงการ)	Progressive (ตามเวลา)
ความเปลี่ยนแปลง	ไม่ต้องการ	ต้อนรับ
ลูกค้า	ที่จุดเริ่มต้นและจุดสิ้นสุด	ตลอดเวลา
Testing	หลัง implementation	ไปพร้อมกับการ implementation
Release	ครั้งเดียว	หลายครั้ง
Feedback	ช้า	เร็ว

Scrum Framework

Scrum Framework

- **นิยาม:** Scrum เป็นกรอบการทำงานที่ช่วยทีมในการพัฒนาผลิตภัณฑ์ที่
 - เน้นการบริหารตัวเองของทีม
 - ทำการสอบเทียบความก้าวหน้า (Inspection) บ่อยๆ
 - ปรับปรุงได้ตลอด (Adaptation)

ที่มาของชื่อ: "Scrum" มาจากกีฬารักบี้ - การรวมตัวของทีมเพื่อแย่งบอล

บทบาทใน Scrum (Scrum Roles)

Product Owner (PO)

- **หน้าที่:**

- สร้างและจัดลำดับ Product Backlog
- สื่อสารความต้องการให้ชัดเจน
- ตัดสินใจความสำคัญของ Feature
- ยอมรับการทำงานที่เสร็จ

- **จุดสำคัญ:** PO ไม่ใช่ Manager ของทีม พวกนั้นเป็น Stakeholder Representative

บทบาทใน Scrum (Scrum Roles)

Scrum Master (SM)

- **หน้าที่:**

- ปกป้องทีม (Servant Leader)
- ลบสิ่งกีดขวาง (Impediment Removal)
- ดำเนินการประชุม Scrum
- ช่วยทีมปฏิบัติ Scrum

- **จุดสำคัญ:** Scrum Master ไม่ใช่ Project Manager ไม่ใช่ Boss

บทบาทใน Scrum (Scrum Roles)

Development Team

- **ลักษณะ:**
 - จำนวน 3-9 คน
 - Cross-Functional
 - บริหารตัวเองได้
 - สัญญา (Commitment) ต่อ Sprint Goal
- **จุดสำคัญ:** ไม่มี Sub-teams หรือลำดับชั้น

Scrum Artifacts

1. Product Backlog

Priority	Feature	Story Points	Status
-----	-----	----	----
1	Login System	13 points	To Do
2	User Profile Page	8 points	To Do
3	Dashboard	21 points	To Do
4	Report Generation	13 points	To Do
5	Notification System	5 points	To Do
...			

- **องค์ประกอบ:** User Stories, Bugs, Technical Debts
- **จัดการโดย:** Product Owner
- **ลำดับ:** ตามความสำคัญ (Priority)

Scrum Artifacts

2. Sprint Backlog

Sprint 1 Goal: Create a working Login System

Stories:

- [] Implement authentication (5 pts)
- [] Create UI form (3 pts)
- [] Add password hashing (2 pts)
- [] Write unit tests (3 pts)...

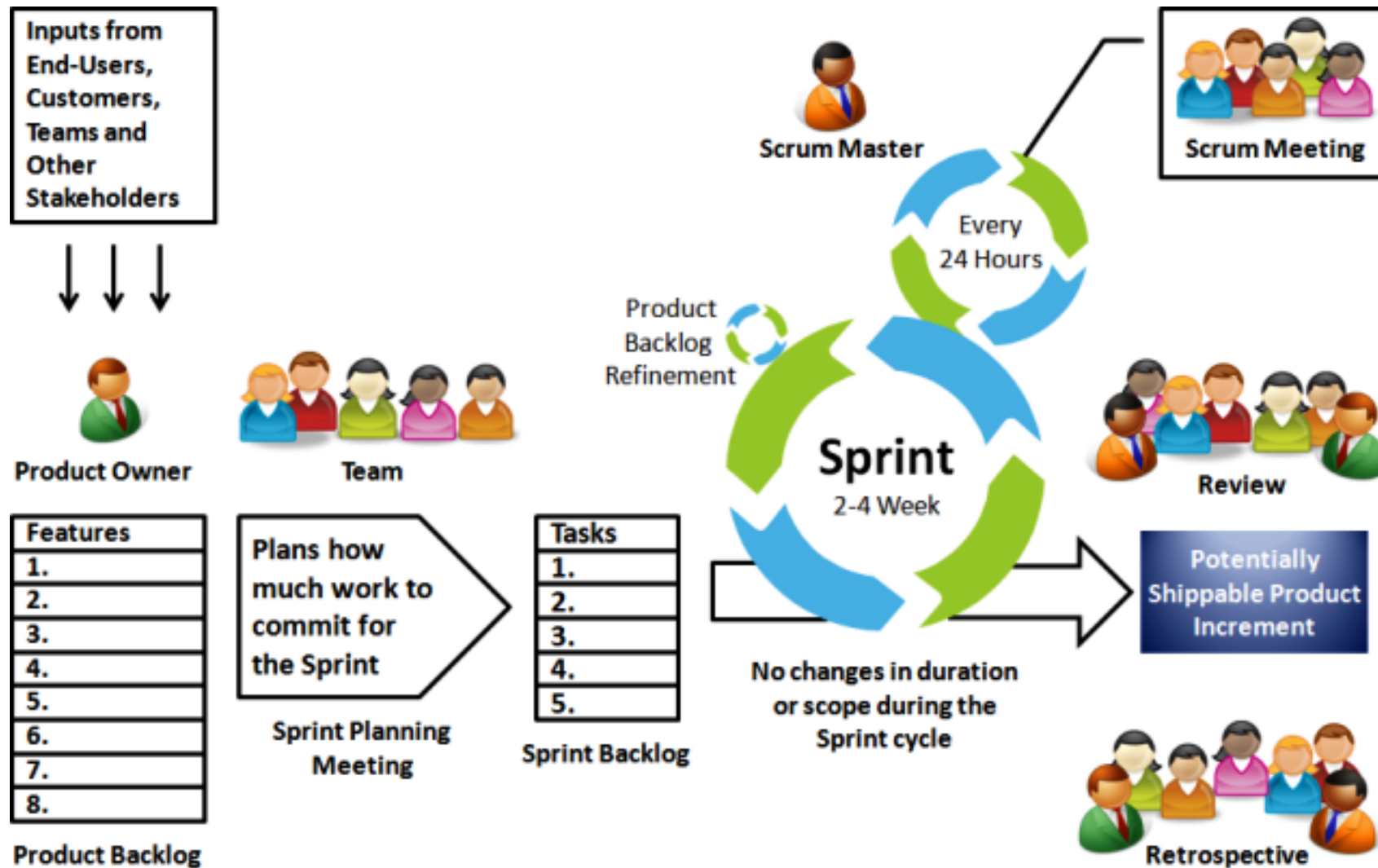
- **องค์ประกอบ:** User Stories ที่เลือกสำหรับ Sprint นี้
- **จัดการโดย:** Development Team
- **ระยะเวลา:** 1 Sprint (1-4 สัปดาห์)

Scrum Artifacts

3. Increment (Product Increment)

- ผลิตภัณฑ์ที่เสร็จแล้ว ใช้งานได้ (Done)
- สามารถ Release ได้
- Potentially Shippable

เหตุการณ์ใน Scrum (Scrum Events)



เหตุการณ์ใน Scrum (Scrum Events)

Sprint Planning (4 ชั่วโมง)

- **ผู้เข้าร่วม:** PO, Scrum Master, Development Team
- **วัตถุประสงค์:**
 - เลือก User Stories จาก Backlog
 - สร้าง Sprint Backlog
 - กำหนด Sprint Goal
 - Estimate effort (Story Points)
- **ผลลัพธ์:** Sprint Backlog พร้อม Sprint Goal ที่ชัดเจน

ตัวอย่าง Sprint Planning:

Sprint Goal: "Enable users to create and manage their profile"

Selected Stories:

1. As a user, I want to view my profile information

Estimate: 5 points

Tasks: UI Design, Backend API, Validation

2. As a user, I want to edit my profile

Estimate: 8 points

Tasks: Edit form, Update API, Testing

3. As admin, I want to see user statistics

Estimate: 8 points

Tasks: Dashboard design, Query optimization

เหตุการณ์ใน Scrum (Scrum Events)

Daily Standup (15 นาที)

- **ผู้เข้าร่วม:** Development Team (PO, SM optional)
- **คำถาม 3 ข้อ (ของแต่ละคน):**
 - What did I do yesterday?
 - What will I do today?
 - What obstacles are in my way?
- **จุดสำคัญ:** ไม่ใช่ Status Report ต่อ Manager

ตัวอย่าง

Developer 1:

- Yesterday: Implemented login API
- Today: Write unit tests for login
- Blocker: Need DB access from team lead

Developer 2:

- Yesterday: Design UI mockup
- Today: Start Frontend components
- Blocker: None

Developer 3:

- Yesterday: Set up testing environment
- Today: Create test cases for login
- Blocker: Wait for API docs

เหตุการณ์ใน Scrum (Scrum Events)

Sprint Review (4 ชั่วโมง)

- **ผู้เข้าร่วม:** PO, SM, Team, Stakeholders
- **กิจกรรม:**
 - Demo สิ่งเสร็จแล้ว
 - รับ Feedback จาก Stakeholder
 - อัปเดต Product Backlog
 - Celebrate success
- **ผลลัพธ์:** Feedback, Updated Backlog

เหตุการณ์ใน Scrum (Scrum Events)

Sprint Retrospective (1.5 ชั่วโมง)

- **ผู้เข้าร่วม:** SM, Development Team
- **คำถามหลัก:**
 - What went well? (ทำได้ดี)
 - What didn't go well? (ทำไม่ได้)
 - What should we improve? (ปรับปรุงอะไร)
- **ผลลัพธ์:** Action Items สำหรับ Sprint ถัดไป

ตัวอย่าง

What went well:

- Good communication within team
- Finished all stories on time
- Test coverage improved

What didn't go well:

- Testing started too late
- API documentation unclear
- Tool setup took too long

Improvement Actions (for next sprint):

- Define API spec before development
- Start testing earlier
- Prepare environment setup in advance

Velocity และ Capacity Planning

- **Velocity** = จำนวน Story Points ที่ทีมสำเร็จต่อ Sprint

Sprint 1: 21 points completed

Sprint 2: 23 points completed

Sprint 3: 20 points completed

Sprint 4: 24 points completed

Sprint 5: 22 points completed

Average Velocity = $(21 + 23 + 20 + 24 + 22) / 5 = 22$ points/sprint

- **ใช้ประโยชน์:**
 - วางแผนวันที่ Release
 - ประมาณจำนวน Backlog Items ที่เหลือ
 - ตรวจสอบเสถียรภาพของทีม

ตัวอย่าง Release Planning

- Backlog Items: 100 story points
- Velocity: 22 points/sprint
- Number of sprints needed: $100 / 22 \approx 4.5$ sprints (5 sprints)
- Timeline: 5 sprints \times 2 weeks = 10 weeks = ~2.5 months

Kanban Method

Kanban คืออะไร

- **ประวัติ:** มาจากระบบการจัดการสิ่งของของ Toyota (Lean Manufacturing)
- **นิยาม:** Kanban คือวิธีการจัดการการไหลของงาน ด้วยการจำกัด Work In Progress (WIP)
- **คำว่า "Kanban":** มาจากภาษาญี่ปุ่น = "kan" (สัญญาณ) + "ban" (การ์ด)

หลักการพื้นฐาน 3 ประการ

Visualize Workflow (มองเห็นการไหลของงาน)

Backlog	To Do (WIP:3)	In Dev (WIP:2)	Testing (WIP:2)	Done
Feature1 Feature5	Feature2 Feature6	Feature3	Feature5	Feature4

- ใช้ Kanban Board (ฟลิ๊กล หรือดีจิทล)
- บัตรแสดงแต่ละ Task/User Story
- ชัดเจน มองเห็น สถานะของงาน

หลักการพื้นฐาน 3 ประการ

Limit Work In Progress (WIP) (จำกัดงานที่อยู่ระหว่างการทำ)

• ประโยชน์ของ WIP Limit:

- ลดความผิดพลาด (Defects)
- ลดเวลาทำงาน (Lead Time)
- โฟกัสงาน (Focus)
- เพิ่มสัญญาณเตือน Bottleneck

โดยไม่มี WIP Limit:

Task	Status
1	Done
2	Done
3	In Dev
4	In Dev
5	In Dev
6	To Do
7	To Do
8	To Do

Context Switching สูง
Lead Time ยาว
Quality ต่ำ

เมื่อใช้ WIP Limit:

Task	Status
1	Done
2	In Dev
3	In Dev
4	To Do (รอ)
5	To Do (รอ)

Context Switching ต่ำ
Lead Time สั้น
Quality สูง

หลักการพื้นฐาน 3 ประการ

Manage Flow (จัดการการไหล)

Throughput = Work completed per unit time

Lead Time = Time from start to finish

Cycle Time = Time in progress

- ติดตามเวลาที่ Task ค้างในแต่ละขั้น
- ค้นหา Bottleneck
- ปรับปรุง Process

Kanban vs Scrum

มิตี	Kanban	Scrum
ลำดับเวลา	Continuous Flow	Fixed Sprint (1-4 weeks)
WIP Limit	ต้องมี	ไม่บังคับ
Role	ไม่มี Fixed Role	Role ชัดเจน
Commitment	ต่อ Flow	ต่อ Sprint Goal
Change	ได้ตลอดเวลา	ไม่ได้ในระหว่าง Sprint
Retrospective	ต่อเนื่อง	ครั้งละ Sprint
เหมาะสำหรับ	Continuous delivery	Project-based work
Team Size	ยืดหยุ่น	3-9 คน

ตัวอย่าง Kanban Board ในการทำงาน

Monday 09:00 AM:

Backlog (WIP:∞)	To Do (WIP:3)	In Dev (WIP:2)	Done
F1	F2 F5 F7	F3 F6	F4

Monday 14:00 (Developer X แล้วเสร็จ F3):

Backlog	To Do (WIP:2)	In Dev (WIP:1)	Done
F1	F2 F5	F6	F3 F4

(F5 ย้ายจาก To Do ไป In Dev โดยอัตโนมัติ)

Activity 1: Scrum Simulation (40 นาที)

- **วัตถุประสงค์:** สัมผัสการทำ Scrum
- **แบ่งกลุ่ม:** 3-4 กลุ่ม ทีมละ 4-5 คน
- **บทบาท:**
 - 1 คน = Product Owner (กลุ่มอื่น)
 - 1 คน = Scrum Master
 - 2-3 คน = Development Team
- **อ่าน [เอกสารการจำลอง Scrum](#)**

Activity 2: Kanban Flow Analysis

- **วัตถุประสงค์:** เข้าใจข้อจำกัดของ WIP, จุดคอขวด, และการปรับปรุงการไหลของงานใน Kanban
- **อ่าน การวิเคราะห์การไหลของงานแบบ Kanban**

HW: