

## Activity 1: การจำลอง Scrum - สัปดาห์ที่ 2

รายละเอียดกิจกรรม

ชื่อ: กิจกรรมจำลอง Scrum

ระยะเวลา: 40 นาที

เป้าหมาย: นิสิตจะเข้าใจ Scrum workflow ผ่านการปฏิบัติจริง

จำนวนคน: 3-5 คนต่อกลุ่ม (แบ่งบทบาท)

### ส่วนที่ 1: การเตรียมตัว (5 นาที)

จัดเตรียม

อุปกรณ์:

- ✓ กระดาษโน้ตแบบมีกาว (หลายสี)
- ✓ กระดานไวท์บอร์ด/กระดานพลิก
- ✓ ปากกาเคมี
- ✓ นาฬิกาจับเวลา (สำหรับกำหนดเวลา)
- ✓ การ์ด Planning Poker (หรือตัวเลข 1-2-3-5-8-13...)

บทบาทในทีม:

- Product Owner (PO) จำนวน 1 คน
- Scrum Master (SM) จำนวน 1 คน
- Developers จำนวน 2-3 คน

การจัดห้อง:

- กระดานที่มีคอลัมน์: TODO, IN PROGRESS, REVIEW, DONE
- PO จัดเรียง product backlog
- ทีมนั่งล้อมรอบกระดาน

### Product Backlog (จำลอง)

PO ได้รับความต้องการ (requirements) ดังต่อไปนี้:

เป้าหมาย: สร้างระบบ "Simple Restaurant Ordering" (ระบบสั่งอาหารออนไลน์แบบง่าย)

ความต้องการ (ยังไม่ได้เรียงลำดับความสำคัญ): 10 ข้อ

1. ผู้ใช้สามารถเข้าสู่ระบบด้วย email และ password
2. เรียกดูเมนูและค้นหาตามหมวดหมู่
3. เพิ่มรายการลงในตะกร้าและจัดการจำนวน
4. คำนวณราคารวมและแสดงสรุป
5. ชำระเงินและประมวลผลการชำระเงิน
6. ยืนยันคำสั่งซื้อและส่งอีเมลแจ้งเตือน
7. แดชบอร์ดสำหรับแอดมินจัดการเมนู
8. ผู้ใช้สามารถติดตามสถานะคำสั่งซื้อ
9. รีวิวและให้คะแนนรายการอาหาร
10. รองรับโค้ดส่วนลด

## ส่วนที่ 2: การวางแผน Sprint (10 นาที)

### ขั้นตอนที่ 1: PO นำเสนอและจัดลำดับความสำคัญ (3 นาที)

PO นำเสนอ backlog ที่เรียงลำดับความสำคัญแล้ว:

ลำดับความสำคัญ 1 (MVP - ต้องมี):

User Story 1: ผู้ใช้เข้าสู่ระบบ

"ในฐานะผู้ใช้ ฉันต้องการเข้าสู่ระบบอย่างปลอดภัย"

เกณฑ์การยอมรับ: รูปแบบที่มีการตรวจสอบ email/password

เมื่อสำเร็จ: เห็นหน้าแดชบอร์ด

ประมาณการ: ? (ต้องหารือกัน)

ความสำคัญ: ●●● สำคัญมาก

PO: "นี่คือฟีเจอร์ MVP แรก ผู้ใช้ต้องเข้าสู่ระบบได้"

User Story 2: เรียกดูเมนู

"ในฐานะผู้ใช้ ฉันต้องการเห็นรายการอาหาร"

เกณฑ์การยอมรับ: แสดงรายการพร้อมรูปภาพและราคา

ฟิลเตอร์หมวดหมู่ใช้งานได้

ค้นหาตามชื่อได้

ประมาณการ: ? (ต้องหารือกัน)

ความสำคัญ: ●●● สำคัญมาก

PO: "ผู้ใช้ไม่สามารถสั่งอาหารได้ถ้าไม่เห็นเมนู!"

User Story 3: ตะกร้าสินค้า

"ในฐานะผู้ใช้ ฉันต้องการจัดการตะกร้าของฉัน"

เกณฑ์การยอมรับ: เพิ่ม/ลบรายการ เปลี่ยนจำนวน

แสดงราคารวม

ประมาณการ: ? (ต้องหารือกัน)

ความสำคัญ: ●●● สูง

PO: "ต้องมีตะกร้าก่อนชำระเงิน"

User stories อื่นๆ: ชำระเงิน, การชำระเงิน, แอดมิน, การติดตาม ฯลฯ  
(ความสำคัญต่ำกว่า สำหรับ Sprint 2+)

คำถามจากทีม:

Dev A: "การค้นหาเป็นแบบคลุมเครือหรือตรงทุกตัวอักษร?"

PO: "เริ่มจากตรงทุกตัวอักษร ปรับปรุงทีหลังถ้ามีเวลา"

Dev B: "เมนูอัปเดตบ่อยไหม?"

PO: "ทุกวัน แอดมินเป็นคนอัปเดต"

SM: "มีอะไรอีกไหม? พร้อมประมาณการไหม?"

ทีม: "พร้อม!"

**ขั้นตอนที่ 2: Planning Poker - การประมาณการ (5 นาที)**

SM: "มาประมาณการ Story 1: ผู้ใช้เข้าสู่ระบบ"

Story 1: ผู้ใช้เข้าสู่ระบบ (email/password แบบง่าย)

SM: "ทุกคน คิดถึง:

- ความซับซ้อน?
- ความพยายามที่ต้องใช้?
- มีความท้าทายทางเทคนิคไหม?"

การอภิปราย: (1 นาที)

Dev A: "ค่อนข้างตรงไปตรงมา ต้องมีการเข้ารหัสผ่าน"

Dev B: "ต้องมีการตรวจสอบอีเมล"

Dev C: "จำกัดอัตราการพยายามเข้าสู่ระบบเพื่อความปลอดภัย?"

PO: "ดีนะ มาเพิ่มในเกณฑ์การยอมรับ"

การประมาณการ: (Planning Poker)

SM: "3-2-1-เริ่ม! ทุกคนเปิดการ์ด!"

การ์ดที่เปิด:

Dev A: 5 คะแนน

Dev B: 8 คะแนน

Dev C: 5 คะแนน

SM: "Dev B บอก 8 คนอื่นบอก 5 Dev B ทำไม?"

Dev B: "การตรวจสอบอีเมลอาจยุ่งยาก กฎรหัสผ่าน..."

Dev A: "อืม ดีนะ แต่เรามี library ช่วยนะ"

Dev B: "โอเค บางที 5 คะแนนก็ยุติธรรม"

ลงคะแนนใหม่ (ถ้าจำเป็น):

Dev A: 5

Dev B: 5 (เปลี่ยน)

Dev C: 5

เห็นตรงกัน: 5 คะแนน ✓

---

Story 2: เรียกดูเมนู

การประมาณการ:

Dev A: 5

Dev B: 8

Dev C: 8

SM: "ไม่เห็นตรงกัน Dev A บอก 5 คนอื่นบอก 8?"

Dev A: "แค่ดึงจากฐานข้อมูลและแสดงผล"

Dev B: "การค้นหา + ฟิลเตอร์หมวดหมู่เพิ่มความซับซ้อน"

Dev C: "ต้องปรับประสิทธิภาพ แบ่งหน้า"

Dev A: "จุดดี! ใช้ 8 สมเหตุสมผล"

ลงคะแนนใหม่:

Dev A: 8

Dev B: 8

Dev C: 8

เห็นตรงกัน: 8 คะแนน ✓

---

Story 3: ตะกร้าสินค้า

การประมาณการ:

Dev A: 5

Dev B: 3

Dev C: 5

SM: "Dev B บอก 3 คนอื่นบอก 5?"

Dev B: "ตะกร้าคือการจัดการ state แบบง่าย"

Dev C: "แต่การคงอยู่ของ session กรณีพิเศษ?"

Dev B: "จริง บางที 5 ปลอดภัยกว่า"

ลงคะแนนใหม่:

Dev A: 5

Dev B: 5

Dev C: 5

เห็นตรงกัน: 5 คะแนน ✓

---

ผลการประมาณการ:

Story 1 (เข้าสู่ระบบ): 5 คะแนน ✓

Story 2 (เมนู): 8 คะแนน ✓

Story 3 (ตะกร้า): 5 คะแนน ✓

รวม: 18 คะแนน

ความสามารถของทีม: ~20-25 คะแนน (สำหรับ sprint จำลอง)

การเลือก: ทั้ง 3 stories พอดี! ✓

ขั้นตอนที่ 3: เลือกและยืนยัน (2 นาที)

SM: "ทั้งหมด 18 คะแนนพอดีกับความสามารถของเรา 20-25 คะแนน"

SM: "ทีม พวกคุณยืนยันที่จะทำ 3 stories นี้ไหม?"

Dev A: "ใช่"

Dev B: "ใช่"

Dev C: "ใช่"

SM: "เยี่ยม! Sprint 1 ยืนยันแล้ว:

- ผู้ใช้เข้าสู่ระบบ (5 คะแนน)

- เรียกดูเมนู (8 คะแนน)

- ตะกร้าสินค้า (5 คะแนน)

รวม: 18 คะแนน"

เป้าหมาย Sprint: "ให้ผู้ใช้เข้าสู่ระบบ เรียกดูเมนู จัดการตะกร้าได้"

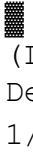
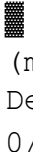
สร้าง SPRINT BACKLOG แล้ว ✓

---

### ส่วนที่ 3: การดำเนินการ Sprint (20 นาที - จำนวน 1 sprint)

วันที่ 1-2: เริ่มพัฒนา (จำนวน 5 นาที = 2 วัน)

สถานะกระดาน SCRUM - วันที่ 1:

TODO	IN PROG	REVIEW	DONE
Story 1 (Login) 5 point	 Story1 (Login) Dev A 1/2 done		
Story 2 (menu) 8 point	 Story2 (menu) Dev B 0/3 done		
Story 3 (cart) 5 point			

Daily Standup - วันที่ 1:

Dev A: "ทำงานเกี่ยวกับฟอร์มเข้าสู่ระบบและ API endpoint"

"เสร็จแล้ว: การตรวจสอบฟอร์ม ✓"

"ต้องทำ: การเข้ารหัสที่ปลอดภัย การทดสอบ"

"อุปสรรค: ไม่มี"

Dev B: "เริ่มออกแบบ API เมนู"

"เสร็จแล้ว: ทบทวน database schema"

"ต้องทำ: ทำ API UI"

"อุปสรรค: ต้องการข้อมูลเมนูตัวอย่าง"

Dev C: "ยังไม่เริ่ม ช่วยงานอื่น"

"ต้องทำ: เริ่มทำตะกร้า"

SM: "ความคืบหน้าดี! Dev B ข้อมูลตัวอย่าง?"

PO: "จะให้ไฟล์ CSV วันนี้"

สถานะ: อยู่ในแผน ✓

สถานะกระดาน SCRUM - วันที่ 2:

TODO	IN PROG	REVIEW	DONE
Story 3 (cart) 5 point	<ul style="list-style-type: none"> <li>■ Story1 (Login) Dev A 2/2 done (ready)</li> <li>■ Story2 (menu) Dev B 2/3 done</li> <li>■ Story3 (cart) Dev C 1/2 done</li> </ul>	<ul style="list-style-type: none"> <li>■ Story1 (Login) Dev B review 1 hr</li> <li>■ Story3 (cart) Dev C tests 1 hr</li> </ul>	

Daily Standup - วันที่ 2:

Dev A: "ทำ login API + tests เสร็จแล้ว ✓"

"ย้ายไป REVIEW"

"ตอนนี้ช่วยทำ menu API"

"อุปสรรค: ไม่มี"

Dev B: "Menu API เกือบเสร็จ"

"เริ่ม UI component"

"ต้องทำ: Category filter"

"อุปสรรค: ไม่มี"

Dev C: "ทำ Cart logic เสร็จ"

"เริ่มทดสอบ"

"ย้ายไป REVIEW"



"อุปสรรค: ไม่มี"

SM: "เยี่ยม! 2 stories อยู่ใน review เคลื่อนไหวเร็ว"

สถานะ: เร็วกว่าแผน! ✓

### วันที่ 3: ทบทวนและปรับปรุง (จำลอง 3 นาที)

สถานะกระดาน SCRUM - วันที่ 3:

TODO	IN PROG	REVIEW	DONE
	 Story2 (menu) Dev B 3/3 done (ready)		✓ Story1 (Login) done 5 point
	 edit: Bugs from testing Dev A,C		✓ Story3 (cart) done 5 point
			✓ Story2 (menu) done 8 point

Daily Standup - วันที่ 3:

Dev A: "ย้าย login ไป DONE ✓"

"ตอนนี้ทดสอบ menu API กับ Dev B"

"เจอบั๊ก: category filter ไม่ทำงาน"

"อุปสรรค: ไม่มี"

Dev B: "Menu API เสร็จ ย้ายไป DONE ✓"

"ช่วยแก้บั๊กที่เจอในการทดสอบ"

"อุปสรรค: ไม่มี"

Dev C: "การทดสอบ Cart ผ่าน ✓"

"ย้ายไป DONE"

"พร้อมช่วยใครก็ได้"

"อุปสรรค: ไม่มี"

SM: "ทั้ง 3 stories เสร็จภายในวันที่ 3!"

"Sprint วางแผน 5 วัน แต่เราเสร็จใน 3 วัน"

"ความเร็วดีมาก!"

สถานะ: SPRINT เสร็จสมบูรณ์ ✓

รวมคะแนนที่เสร็จ: 18 คะแนน

จำนวนวันที่ใช้: 3 จาก 5 วัน

ความเร็ว (Velocity): 18 คะแนน (มากกว่าที่ประมาณการ!)

#### ส่วนที่ 4: การทบทวน Sprint (Demo) - 3 นาที

สถิติให้ผู้มีส่วนได้ส่วนเสีย

Sprint Review - สิ้นสุดวันที่ 3

ผู้เข้าร่วม:

- ทีม Dev (A, B, C)

- SM

- PO

- อาจารย์ (เล่นบทผู้มีส่วนได้ส่วนเสีย)

SM: "Sprint 1 Review! มาดูกันว่าเราสร้างอะไรได้บ้าง"

DEMO 1: พีเจอร์เข้าสู่ระบบผู้ใช้

Dev A: "จะแสดงพีเจอร์เข้าสู่ระบบ"

(เปิดแอปในเบราว์เซอร์)

"นี่คือหน้า landing page พร้อมฟอร์มเข้าสู่ระบบ"

(แสดงช่อง email + password)

"ลองใช้ email ไม่ถูกต้องก่อน:"

(พิมพ์ "invalid") → "รูปแบบ email ไม่ถูกต้อง"

"ลองรหัสผ่านสั้น:"

(พิมพ์ "123") → "รหัสผ่านต้องมีอย่างน้อย 8 ตัวอักษร"

"ตอนนี้ใช้ข้อมูลที่ถูกต้อง:"

(พิมพ์ test@test.com / Password123)

(คลิกเข้าสู่ระบบ)

→ สำเร็จ! แสดงหน้าแดชบอร์ด

"Session ผู้ใช้ถูกสร้าง เก็บไว้ในฐานข้อมูล"

PO: "เยี่ยม! ผู้ใช้ออกจากระบบได้ไหม?"

Dev A: "ได้ ปุ่มออกจากระบบอยู่ตรงนี้ (ชี้ที่มุมบนขวา)"

(คลิกออกจากระบบ) → กลับไปที่หน้าเข้าสู่ระบบ

อาจารย์: "รหัสผ่านปลอดภัยแค่ไหน?"

Dev A: "เข้ารหัสด้วย bcrypt ไม่เก็บในรูปแบบข้อความธรรมดา"

อนุมัติ ✓

DEMO 2: พีเจอร์เรียกดูเมนู

Dev B: "พีเจอร์เรียกดูเมนู"

(คลิกลิงก์เมนูหลังเข้าสู่ระบบ)

"นี่คือหน้าเมนูที่มีรายการทั้งหมด"

(แสดงรายการ: Burger, Pizza, Salad, Sushi ฯลฯ)

(แต่ละรายการมีราคา รูปภาพ คำอธิบาย)

"ค้นหาตามชื่อ:"

(พิมพ์ "pizza") → ฟิลเตอร์แสดงเฉพาะรายการ Pizza

"ฟิลเตอร์หมวดหมู่:"

(คลิกหมวดหมู่ "Italian") → แสดงเฉพาะอาหารอิตาลี

(คลิก "Thai") → แสดงอาหารไทย

"ล้างฟิลเตอร์:"

(คลิก "Clear all") → แสดงรายการทั้งหมดอีกครั้ง

"แต่ละรายการมีรายละเอียด:"

(คลิก "Sushi") → แสดงคำอธิบายแบบเต็ม

→ ปุ่ม "Add to cart" พร้อมใช้งาน

PO: "การค้นหาจับคู่ชื่อบางส่วนไหม?"

Dev B: "ใช่ 'piz' หา Pizza ได้ 'sal' หา Salad ได้"

อาจารย์: "ประสิทธิภาพโอเคกับ 1000 รายการไหม?"

Dev B: "เพิ่มการแบ่งหน้า โหลด 20 รายการต่อครั้ง"

อนุมัติ ✓

DEMO 3: พี่เจอร์ตะกร้าสินค้า

Dev C: "การจัดการตะกร้าสินค้า"

(เพิ่ม pizza ลงตะกร้าจากเมนู)

(เพิ่ม burger)

(เพิ่ม salad)

"คลิกไอคอนตะกร้า (มุมบนขวา)"

(แสดง: Pizza x1 @ 12.99, Burger x1 @ 10.99, Salad x1 @ 8.99)

(รวม: 32.97)

"เปลี่ยนจำนวน:"

(เพิ่ม pizza เป็น x2) → รวมอัปเดต: 45.96

"ลบรายการ:"

(ลบ salad) → อัปเดตเป็น 23.98

"ตะกร้าคงอยู่:"

(ปิดเบราว์เซอร์ เปิดใหม่) → รายการยังอยู่!

(เพราะเก็บในฐานข้อมูล)

PO: "ตะกร้ามีขีดจำกัดไหม?"

Dev C: "ตอนนี้ยังไม่มี แต่สามารถเพิ่มจำนวนสูงสุดต่อรายการได้"

อาจารย์: "ดูดีมาก! พร้อมชำระไหม?"

Dev C: "พี่เจอร์ชำระเงินอยู่ใน Sprint 2"

อนุมัติ ✓

สรุป SPRINT

SM: "สรุป:"

"✓ Story 1: เข้าสู่ระบบ - เสร็จสมบูรณ์"

"✓ Story 2: เมนู - เสร็จสมบูรณ์"

"✓ Story 3: ตะกร้า - เสร็จสมบูรณ์"

"✓ รวม: 18 story points"

"✓ เสร็จใน 3 วัน (เร็วกว่า 2 วัน!)"

PO: "ความคิดเห็น?"

"ทำได้ยอดเยี่ยม! พี่เจอร์ทั้งหมดทำงานได้"

"Sprint ต่อไป: ชำระเงิน การชำระเงิน ยืนยันคำสั่งซื้อ"

อาจารย์: "มีคำถามไหม?"

ทีม: (ไม่มี)

ผลลัพธ์:

- ✓ ผู้มีส่วนได้ส่วนเสียเห็นความคืบหน้า
- ✓ ความคิดเห็นถูกรวมเข้า Sprint ต่อไป
- ✓ ทีมรู้สึกสำเร็จ

---

## ส่วนที่ 5: การทบทวนย้อนหลัง Sprint (2 นาที)

การสะท้อนคิดของทีม

Retrospective - หลังจาก Review ทันที

เฉพาะทีม (ไม่มี PO/อาจารย์)

SM: "อะไรที่ดีใน sprint นี้?"

Dev A: "Planning poker ช่วยให้เราประมาณการได้ดีขึ้น"

Dev B: "ทำงานเป็นทีมดี ช่วยเหลือกัน"

Dev C: "เสร็จเร็ว ไม่เครียด"

กระดานไวท์บอร์ด - "ดี":

- ✓ Planning poker ตกงกันได้
- ✓ เป้าหมาย sprint ชัดเจน
- ✓ ทำงานเป็นทีมและสื่อสาร
- ✓ เสร็จเร็วกว่า 2 วัน
- ✓ ไม่มีอุปสรรค
- ✓ การทดสอบดี

SM: "อะไรที่ไม่ดี?"

Dev A: "Standup อาจสั้นกว่านี้ได้"

Dev B: "Schema เริ่มต้นไม่สมบูรณ์ ต้องออกแบบใหม่"

Dev C: "ควร pair-program มากขึ้นเพื่อเรียนรู้"

กระดานไวท์บอร์ด - "ไม่ดี":

X Standup ใช้เวลา 10 นาที (เป้าหมาย 5)

X ออกแบบ database schema ใหม่ (เสียเวลา)

X Pair programming จำกัด

X ไม่มีข้อมูลทดสอบในตอนแรก

SM: "เราจะปรับปรุงอะไร?"

Dev A: "กำหนดเวลา standup ไม่เกิน 5 นาที"

Dev B: "วางแผน database schema ดีวก่อน coding"

Dev C: "Pair program ในงานที่ซับซ้อน"

กระดานไวท์บอร์ด - "การปรับปรุง":

→ กำหนดเวลา Standup: 5 นาที

→ ทบทวน Schema ก่อน coding

→ Pair programming เพื่อการเรียนรู้

→ เตรียมข้อมูลทดสอบล่วงหน้า

SM: "แผนการดำเนินการสำหรับ sprint ต่อไป?"

1. Standup master บังคับ 5 นาที

2. ทบทวน Database ในการวางแผน (เพิ่มงาน)

3. มอบหมายการจับคู่สำหรับ 1-2 stories

4. เตรียมข้อมูลทดสอบล่วงหน้า

ความมุ่งมั่น:

ทุกคน: "ใช่ เราจะทำใน Sprint 2"

---

ตัวชี้วัดที่บันทึก:

### Sprint 1:

- ✓ Velocity: 18 คะแนน (เสร็จสมบูรณ์)
- ✓ ความสามารถของทีม: 20-25 คะแนน
- ✓ ประสิทธิภาพ: 18, เสร็จ: 18 (100% ✓)
- ✓ วันที่ใช้: 3 จาก 5 วัน
- ✓ Lead time: เฉลี่ย 3 วัน
- ✓ ความสุขของทีม: สูง ✓

### ส่วนที่ 6: บทเรียนและการสรุป (จำลองเสร็จสมบูรณ์)

#### สิ่งที่ได้เรียนรู้

สิ่งที่ทีมได้เรียนรู้:

#### 1. การวางแผน SPRINT มีความสำคัญ

- ✓ Planning poker ช่วยประมาณการที่สมจริง
- ✓ การอภิปรายเปิดเผยความซับซ้อน
- ✓ ความเห็นพ้องของทีม → ความมุ่งมั่น
- ✓ เป้าหมาย sprint รักษาไฟกัส

#### 2. DAILY STANDUP ทำให้สิ่งต่างๆ เดินหน้าต่อ

- ✓ Standup 15 นาทีทำให้ทีมสอดคล้องกัน
- ✓ ระบุอุปสรรคแต่เนิ่นๆ
- ✓ ช่วยเหลือกัน
- ✓ ป้องกันการสูญเปล่า

#### 3. การไหลอย่างต่อเนื่องและการมองเห็น

- ✓ กระดาน Kanban แสดงความคืบหน้าจริง
- ✓ ขีดจำกัด WIP ป้องกันภาระเกิน
- ✓ การย้ายการ์ด → แรงจูงใจ
- ✓ ทุกคนรู้สถานะ

#### 4. ติดตามความต้องการ (ไม่ใช่ผลักดัน)

- ✓ นักพัฒนาทำงานเมื่อพร้อม
- ✓ ไม่ใช่ผู้จัดการมอบหมาย
- ✓ มีอิสระมากขึ้น
- ✓ ขวัญกำลังใจดีขึ้น

## 5. วงรอบการทบทวนและข้อเสนอแนะ

- ✓ สถิติงานจริง (ไม่ใช่สไลด์)
- ✓ ผู้มีส่วนได้ส่วนเสียเห็นความคืบหน้า
- ✓ ข้อเสนอแนะแต่เนิ่น (ไม่ใช่ตอนท้าย)
- ✓ ปรับ sprint ต่อไป

## 6. RETROSPECTIVE = การปรับปรุง

- ✓ ทีมสะท้อนคิดอย่างสร้างสรรค์
- ✓ ระบุสิ่งที่ต้องปรับปรุง
- ✓ ทำความมุ่งมั่น
- ✓ การพัฒนาอย่างต่อเนื่อง

## 7. VELOCITY คาดการณ์ได้

- ✓ 18 คะแนนใน Sprint 1
- ✓ สามารถคาดการณื: 18 คะแนน/sprint
- ✓ วางแผนใหม่ไฉนจากข้อมูล
- ✓ เชื่อถือได้ (ไม่ใช่แค่เดา)

## เปรียบเทียบ: ก่อนจำลอง กับ หลังจำลอง

ก่อนการจำลอง:

"Agile? Scrum? Sprint planning? ฟังดูซับซ้อน..."

"เราประมาณการยังไง? แค่เดาชั่วโมง?"

"ทำไมไม่มีประชุมเยอะจัง?"

"ติดตามความคืบหน้ายังไง?"

หลังการจำลอง:

"โอ้! Planning poker ทำให้การประมาณการโปร่งใส"

"Daily standup ทำให้เราโฟกัส (และสั้น!)"

"การแสดงผลด้วยกระดานทรงพลัง"

"Sprint review แสดงคุณค่าที่เป็นจริง"

"Retro ช่วยให้เราปรับปรุงกระบวนการ"

"เราเสร็จเร็วกว่า 2 วัน!"

"Velocity ช่วยคาดการณืใหม่ไฉน"

การเปลี่ยนแปลงความคิด:

ก่อน: คิดแบบ Waterfall (คาดการณืทุกอย่างล่วงหน้า)

หลัง: คิดแบบ Agile (วางแผน ทำ เรียนรู้ ปรับ)

## 🎓 รายการตรวจสอบผลการเรียนรู้

### นิสิตสามารถ:

#### เข้าใจกระบวนการ Sprint Planning

- ✓ Product backlog → จัดลำดับความสำคัญ
- ✓ การประมาณการด้วย Planning poker
- ✓ ความมุ่งมั่นของทีมต่อ sprint

#### เข้าร่วม daily standups

- ✓ สิ่งที่ทำเมื่อวาน
- ✓ สิ่งที่จะทำวันนี้
- ✓ มีอุปสรรคไหม?

#### ใช้กระดาน Kanban อย่างมีประสิทธิภาพ

- ✓ ย้ายการ์ดตามความคืบหน้า
- ✓ ติดตามขีดจำกัด WIP
- ✓ เห็นการไหลของงาน

#### ปฏิบัติงานเป็นทีม Scrum

- ✓ จัดองค์กรตนเอง
- ✓ ทำงานร่วมกัน
- ✓ เน้นที่เป้าหมาย sprint

#### นำเสนอใน Sprint Review

- ✓ สาธิตฟีเจอร์ที่ทำงานได้
- ✓ ตอบคำถาม
- ✓ รวบรวมข้อเสนอแนะ

#### สะท้อนคิดใน Retrospective

- ✓ อะไรที่ดี?
- ✓ อะไรที่ต้องปรับปรุง?
- ✓ ทำความมุ่งมั่น

#### เข้าใจตัวชี้วัด Scrum

- ✓ Velocity (คะแนนที่เสร็จสมบูรณ์)
- ✓ Lead time และ cycle time
- ✓ Throughput

#### ชื่นชมกรอบความคิด Agile

- ✓ การปรับปรุงอย่างต่อเนื่อง

- ✓ ยอมรับการเปลี่ยนแปลง
  - ✓ เน้นที่คุณค่า
  - ✓ ทีมมากกว่าบุคคล
-

ให้นักนิสิตเขียน

ชื่อ: \_\_\_\_\_

วันที่: \_\_\_\_\_

1. Scrum Planning คืออะไร? อธิบายตามความเข้าใจของคุณ

\_\_\_\_\_

[พื้นที่สำหรับคำตอบ]

2. ทำไม Planning Poker ดีกว่าการประมาณการชั่วโมง?

\_\_\_\_\_

[พื้นที่สำหรับคำตอบ]

3. บทบาทใดในทีมที่คุณชอบมากที่สุด? เพราะเหตุใด?

\_\_\_\_\_

[พื้นที่สำหรับคำตอบ]

4. อะไรยากที่สุดในการจำลอง Scrum?

\_\_\_\_\_

[พื้นที่สำหรับคำตอบ]

5. คุณจะนำแนวคิด Scrum ไปใช้ใน project จริงได้อย่างไร?

\_\_\_\_\_

[พื้นที่สำหรับคำตอบ]

6. ให้คะแนนความเข้าใจของคุณ (1-5):

ก่อน:  1  2  3  4  5

หลัง:  1  2  3  4  5

คะแนน: \_\_\_/5 คะแนน

---

## จำลอง Sprint 2 (ขยายเวลาตามต้องการ) ถ้าเวลายังเหลือ

ดำเนินการต่อด้วย Sprint 2:

PO: "ลำดับความสำคัญ Sprint 2:"

1. ภาระงานการชำระเงิน (8 คะแนน)
2. การรวมระบบชำระเงิน (13 คะแนน)
3. อีเมลยืนยันคำสั่งซื้อ (5 คะแนน)
4. แก้บั๊กจาก Sprint 1 (2 คะแนน)

รวม: 28 คะแนน

ทีม: "เยอะเกินไป เราทำได้ 18 คะแนน"

"Velocity บอกว่า ~18"

"ทำ 28 ไม่ได้"

PO: "โอเค ลดขอบเขต:"

1. ภาระงานการชำระเงิน (8 คะแนน)
2. ยืนยันคำสั่งซื้อ (5 คะแนน)
3. แก้บั๊ก (2 คะแนน)

รวม: 15 คะแนน

ทีม: "นี่สบายดี"

SM: "Sprint 2: 15 คะแนน, 3 stories"

"เป้าหมาย Sprint: ทำการสั่งซื้อให้สมบูรณ์"

(ทำซ้ำรอบเดิม: Planning → Daily → Demo → Retro)

---

## เกณฑ์ความสำเร็จ

### กิจกรรมประสบความสำเร็จเมื่อ:

- ✓ ทีมทำการวางแผน sprint เสร็จสมบูรณ์ (10 นาที)
  - ✓ สมาชิกทีมทุกคนเข้าใจเป้าหมาย sprint
  - ✓ การ์ดเคลื่อนผ่านกระดานระหว่างการดำเนินการ
  - ✓ Daily standups เกิดขึ้น (แม้จะเป็นการจำลอง)
  - ✓ Sprint Review สานิตทุก stories ที่เสร็จสมบูรณ์
  - ✓ Retrospective สร้างแผนการปรับปรุง
  - ✓ ทีมเข้าใจ Scrum workflow
  - ✓ คำนวณตัวชี้วัด Velocity ได้
  - ✓ ทีมมีส่วนร่วมและเรียนรู้
  - ✓ กำหนดเวลาอย่างเหมาะสม (รวม 40 นาที)
- 

## เอกสารอ้างอิงและการเรียนรู้เพิ่มเติม

### นิติตสามารถศึกษาเพิ่มเติม

#### ไฟล์ที่ควรอ่าน:

- SCRUM-sprint-planning.md (วิธีวางแผน Sprint)
- SCRUM-agile-ceremonies.md (Scrum events)
- SCRUM-velocity.md (Velocity และ metrics)

#### เครื่องมือที่ควรลอง:

- Trello (กระดาน Kanban ฟรี)
- Jira (เครื่องมือ Scrum มืออาชีพ)
- Miro (กระดานไวท์บอร์ดเสมือนสำหรับทำงานระยะไกล)

#### วิดีโอ:

- "Scrum Framework Explained" (5 นาที)
- "Planning Poker Demo" (8 นาที)
- "Scrum in 10 minutes" (YouTube)