Activity 2: การวิเคราะห์การไหลของงานแบบ **Kanban - สัปดาห์ที่ 2**

รายละเอียดกิจกรรม

ชื่อ: Workshop วิเคราะห์การไหลของงานแบบ Kanban **ระยะเวลา**: 20 นาที

เป้าหมาย: นิสิตเข้าใจข้อจำกัดของ WIP, จุดคอขวด, และการปรับปรุงการไหลของงานใน Kanban

จำนวนคน: 3-5 คนต่อกลุ่ม

ผลลัพธ์: การวิเคราะห์ Kanban board + ข้อเสนอแนะ

ส่วนที่ 1: สถานการณ์

บริษัท XYZ ให้ทีมวิเคราะห์ Kanban board ของพวกเขา

ปัญหา:

❌ Features ใช้เวลานาน (เวลานำ = 3 สัปดาห์)

❌ QA เป็นจุดคอขวด (งานติดค้างใน TESTING column)

❌ Reviewers ทำงานหนักเกินไป

❌ นิสิตต้องวิเคราะห์และแนะนำวิธีปรับปรุง

สภาพ Kanban Board ปัจจุบัน

📊 สถานะปัจจุบันของ Board (วันนี้ 1 ธ.ค.):

BACKLOG (40)	TODO (12)	IN PROG (8/5)	TESTING (7/3)	REVIEW (4/2)	DEPLOY (1/1)✓
[20] Future	[8] ready	▓ [API] Dev: Sara do 3 days	▓▓▓ [QA1] Dev: John busy 2 days waiting	▓▓[Rev1] Review: Mike 1 day	▓ [Dep1] Deploy: Tom (ready)
[15] idea	[2] important high	▓ [Search] Dev: Joe do 1 day	▓▓▓ [QA2] QA: Lisa busy 1 day waiting	▓▓[Rev2] Review: Cam 1 day	
[5] may be	[2] important medium	▓ [UI] Dev: Sara do 4 hrs	▓▓▓ [QA3] QA: John busy 3 hrs waiting		
			▓▓▓ [QA4] QA: Lisa busy 1 day waiting		

ข้อจำกัด WIP ปัจจุบัน:

- BACKLOG: ไม่จำกัด
- TODO: ไม่จำกัด (❌ ควรจำกัด!)
- IN PROGRESS: 5 (⚠️ ถึงขด, มี 8 รายการ! เกินขีดจำกัด!)
- TESTING: 3 (🔴 ถึงขีด, มี 7 รายการ! วิกฤติ!)
- REVIEW: 2 (⚠️ ถึงขีด, มี 4 รายการ! เกินขีดจำกัด!)
- DEPLOY: 1 (✓ โอเค)

การฝ่าฝืนขีดจำกัด:

❌ IN PROGRESS: 8 รายการ (ขีดจำกัด 5) = เกิน 60%

🔴 TESTING: 7 รายการ (ขีดจำกัด 3) = เกิน 133% (รุนแรงมาก!)

❌ REVIEW: 4 รายการ (ขีดจำกัด 2) = เกิน 100%

ส่วนที่ 2: วิเคราะห์สภาพปัจจุบัน (5 นาที)

งานนิสิต: วิเคราะห์ปัญหา

งานที่ 1: ระบุการฝ่าฝืน WIP

คอลัมน์	ขีดจำกัด	จริง	การฝ่าฝืน	ความรุนแรง
BACKLOG	∞	40	(ไม่มี)	✓
TODO	∞	12	(ไม่มี)	✓
IN PROGRESS	5	8	+60%	⚠️ ปานกลาง
TESTING	3	7	+133%	🔴 วิกฤติ
REVIEW	2	4	+100%	⚠️ ปานกลาง
DEPLOY	1	1	0%	✓ โอเค

✅ คำตอบ:

วิกฤติที่สุด: คอลัมน์ TESTING (7 รายการ vs ขีดจำกัด 3)

นี่คือจุดคอขวด! 🔴

งานที่ 2: คำนวณตัวชี้วัด

การวิเคราะห์เวลานำ (Lead Time):

การติดตามเส้นทางรายการ:

API Feature:
- สร้าง: 29 พ.ย. (2 วันใน BACKLOG)
- เริ่ม: 30 พ.ย. (IN PROGRESS)
- พร้อมทดสอบ: 1 ธ.ค. (วันนี้)
- อยู่ในการทดสอบ: วันนี้ (รออยู่... ไม่มี QA ว่าง)
- เวลานำจนถึงตอนนี้: 3 วัน (ยังไม่เสร็จ!)

Search Feature:
- สร้าง: 27 พ.ย. (4 วันใน BACKLOG)
- เริ่ม: 29 พ.ย.
- พร้อมทดสอบ: 30 พ.ย.
- อยู่ในการทดสอบ: 30 พ.ย. ⟶ 1 ธ.ค. (ติด 1 วัน)
- รอ review: 1 ธ.ค.
- เวลานำจนถึงตอนนี้: 4+ วัน

UI Component:
- สร้าง: 30 พ.ย.
- เริ่ม: 30 พ.ย.
- พร้อมทดสอบ: 30 พ.ย.
- อยู่ในการทดสอบ: วันนี้ (ติดอยู่!)
- เวลานำจนถึงตอนนี้: 1 วัน (แต่ติดอยู่!)

Feature ที่ Deploy แล้ว (จากก่อนหน้า):
- สร้าง: 20 พ.ย. (11 วันที่แล้ว)
- เสร็จ: 1 ธ.ค.
- เวลานำ: 11 วัน (นานมาก!)
- สาเหตุ: ติดใน TESTING 5 วัน

✅ การคำนวณ:

เวลานำเฉลี่ย (ประมาณการ):

- Features ก่อนหน้า: 11-15 วัน

- เฉลี่ยปัจจุบัน: ~8 วัน (นานเกินไป!)

- เป้าหมาย: 3-4 วัน

Throughput (สัปดาห์ที่แล้ว):

- รายการที่เสร็จ: 3 รายการ

- Throughput: 3 รายการ/สัปดาห์ = 0.6 รายการ/วัน

- เป้าหมาย: 2 รายการ/วัน

Cycle Time (เวลาทำงานจริงเท่านั้น):

- งานจริง: ~1-2 วันต่อรายการ

- เวลารอ: 5-7 วัน (สูญเปล่า!)

- % เวลารอต่อทั้งหมด: 70% ⚠️ (สูงเกินไป!)

งานที่ 3: ระบุสาเหตุหลัก

แผนผังวิเคราะห์ปัญหา:

🔴 จุดคอขวดการทดสอบ (TESTING BOTTLENECK)

```
    │
    ├─ ทำไม? ทีม QA โอเวอร์โหลด
    │    └─ QA 3 คน แต่มี 7 รายการรออยู่
    │         └─ QA แต่ละคนทดสอบหลายรายการพร้อมกัน
    │              └─ ใช้เวลา 1-2 วันต่อรายการ
    │
    ├─ ทำไม? รายการติดอยู่ใน TESTING
    │    └─ รายการที่เสร็จ: 3/วัน
    │    └─ กำลังการผลิต QA: 1 รายการ/วัน (ช้าเกินไป)
    │    └─ คิวสะสมเพิ่มขึ้น
    │
    └─ ทำไม? มีรายการมากเกินไปในคอลัมน์อื่น
         └─ IN PROGRESS: 8 รายการ (นักพัฒนาทำเสร็จแล้ว)
         └─ REVIEW: 4 รายการ (reviewers ทำไม่ทัน)
         └─ ปลายทางติด → ต้นทางทำต่อ
         └─ WIP ระเบิด
```

สาเหตุหลัก:

⚠️ จุดคอขวดปลายทาง (TESTING) บล็อกทุกอย่าง

⚠️ ขีดจำกัด WIP ไม่ได้บังคับใช้ (รายการเกินขีดจำกัด)

⚠️ กำลังการผลิตทีมไม่สมดุล (นักพัฒนามากกว่า QA)

✅ คำตอบ:

จุดคอขวดหลัก: คอลัมน์ TESTING

- กำลังการผลิต QA ไม่สมดุล (QA 3 คน, รายการมากเกินไป)
- รายการเข้าคิว
- นักพัฒนารออยู่ (ดันงานออกไม่ได้)
- Reviewers รออยู่
- ทุกอย่างสะสม!

ส่วนที่ 3: เสนอแนะการปรับปรุง (8 นาที)

ข้อเสนอแนะที่ 1: ปรับขีดจำกัด WIP

📊 ตัวเลือก A: ลด WIP เพื่อบังคับให้ไหล

ปัจจุบัน:

- IN PROGRESS: 5 → ลดเป็น 3
- TESTING: 3 → ลดเป็น 2
- REVIEW: 2 → คงไว้ที่ 2

ขีดจำกัดใหม่:

BACKLOG	TODO	IN PROG	TESTING	REVIEW	DEPLOY
∞	∞	(3/3)	(2/2)	(2/2)	(1/1)

ผลกระทบ:

1. นักพัฒนาไม่สามารถเริ่มงานใหม่ได้ (บล็อกโดยขีดจำกัด)
   → บังคับให้ช่วย QA
   → บังคับให้จบงานปัจจุบัน

2. QA ไม่โอเวอร์โหลด
   → สามารถมุ่งเน้นคุณภาพ

→ จบเร็วขึ้น

3. Reviewers ทำทัน
→ ป้อนกลับเร็วขึ้น
→ ทำซ้ำน้อยลง

ระยะเวลาในการเคลียร์คิว:
- ปัจจุบัน: 7 รายการในการทดสอบ (ควรมี 3)
- ด้วยกำลังการผลิต QA 2 รายการต่อวัน
- เคลียร์รายการส่วนเกินใน: ~2 วัน

ก่อน: เวลานำ 11 วัน
หลัง: เวลานำ 3-4 วัน
การปรับปรุง: เร็วขึ้น 70%! ✓

✅ **ข้อเสนอแนะที่ 1:**
ลดขีดจำกัด WIP:
- IN PROGRESS: 5 → 3
- TESTING: 3 → 2
คาดหวัง: การไหลเร็วขึ้น, ของเสียน้อยลง

ข้อเสนอแนะที่ 2: บังคับใช้ระบบดึง (Pull System)
📋 ตัวเลือก B: การทำงานแบบดึงที่เข้มงวด

ปัญหาปัจจุบัน:
❌ Dev A เริ่มงานใหม่ (IN PROGRESS เกินขีดจำกัด)
❌ Dev B เริ่มงานใหม่ (ดันรายการลงไป)
❌ QA โอเวอร์โหลด (ดึงไม่ได้, ถูกผลัก)
❌ คิวรายการรออยู่

วิธีที่ดีกว่า: ดึง (PULL)
กฎ: "อย่าเริ่มงานใหม่ถ้าปลายทางบล็อก"

ตัวอย่าง:
Dev A ทำงานเสร็จ ⟶ "TESTING มี 2 รายการ"
Dev A: "Testing ถึงขีด, ดึงงานใหม่ไม่ได้"

Dev A: "จะช่วยอะไรได้บ้าง?"

ตัวเลือกที่ 1: ช่วย QA ทดสอบให้เสร็จ
ตัวเลือกที่ 2: ช่วยแก้บั๊กใน REVIEW
ตัวเลือกที่ 3: Pair programming กับ dev ที่กำลังติดปัญหา

ผลลัพธ์:
✓ คิวเคลียร์เร็วขึ้น
✓ ทีมช่วยกันแก้จุดคอขวด
✓ นักพัฒนาปลดบล็อก
✓ การไหลดีขึ้น

กฎการทำงาน:
IF (รายการ IN_PROGRESS = ขีดจำกัด) THEN
→ ไม่สามารถเริ่มงานใหม่ได้
→ ช่วยคอลัมน์อื่นแทน
ENDIF

✅ **ข้อเสนอแนะที่ 2:**
บังคับใช้วินัยการดึง:
- ไม่สามารถเริ่มงานใหม่ถ้าปลายทางบล็อก
- ช่วยเพื่อนร่วมทีมเคลียร์จุดคอขวดก่อน
คาดหวัง: ทำงานร่วมกันดีขึ้น, จบงานเร็วขึ้น

ข้อเสนอแนะที่ 3: เพิ่มกำลังการผลิต QA
👥 ตัวเลือก C: การปรับสมดุลทีม

ทีมปัจจุบัน:
- นักพัฒนา: 4-5 คน
- QAs: 3 คน
- Reviewers: 2 คน

ปัญหาอัตราส่วน:
- 5 devs : 3 QA = 1.67 devs ต่อ QA (มากเกินไป!)
- Devs ทำเสร็จ 2 รายการ/วัน

- QAs ทดสอบได้ 1 รายการ/วัน
- คิวเพิ่ม 1 รายการ/วัน

วิธีแก้:

แนวทางที่ 1: จ้าง QA เพิ่ม

- เพิ่ม 2 QA (รวม 5 คน)
- อัตราส่วน: 1 dev : 1 QA (สมดุล)
- กำลังการผลิต: 5 รายการ/วัน
- ต้นทุน: สูง 💰

แนวทางที่ 2: ฝึกอบรมข้ามสายงาน

- ฝึก 1-2 นักพัฒนาให้ช่วยงาน QA
- นักพัฒนาทำการทดสอบด้วยตนเอง + อัตโนมัติ
- จำนวนคนเท่าเดิม
- กำลังการผลิตเพิ่มเป็น 4 รายการ/วัน
- ต้นทุน: ต่ำ (เวลาฝึกอบรม) ✓

แนวทางที่ 3: ทดสอบอัตโนมัติ

- เขียนการทดสอบอัตโนมัติ
- QAs เขียน, นักพัฒนาปรับใช้
- วงจรการทดสอบเร็วขึ้น
- กำลังการผลิต: 3-4 รายการ/วัน
- ต้นทุน: ปานกลาง (เครื่องมือ + การเรียนรู้)

แนะนำ: **แนวทางที่ 2 + 3**

- ฝึกอบรมข้ามสายงาน 1 dev สำหรับงาน QA (ทันที)
- ลงทุนในการทดสอบอัตโนมัติ (ระยะยาว)

ระยะเวลา:

- เดือนที่ 1: ฝึกอบรมข้ามสายงาน 1 dev
- เดือนที่ 2: ตั้งค่าการทดสอบอัตโนมัติ
- เดือนที่ 3: ปรับปรุงกำลังการผลิตเต็มรูปแบบ

คาดหวัง:

- ก่อน: throughput 0.6 รายการ/วัน
- หลัง: throughput 2-2.5 รายการ/วัน

- ปรับปรุง 300-400%! 🚀

✅ **ข้อเสนอแนะที่ 3:**
เพิ่มกำลังการผลิต QA:
- ฝึกอบรมข้ามสายงาน 1 นักพัฒนาสำหรับงาน QA
- ลงทุนในการทดสอบอัตโนมัติ
- ปรับสมดุลอัตราส่วนทีม
คาดหวัง: Throughput เร็วขึ้น, จุดคอขวดลดลง

ข้อเสนอแนะที่ 4: การจัดการการไหลรายวัน
🗓️ ตัวเลือก D: กระบวนการการไหลรายวัน

พิธีกรรมใหม่: Daily Flow Standup (5 นาที)

มุ่งเน้น: ไม่ใช่ Sprint (Kanban ไม่มี sprints)
มุ่งเน้น: การไหล & สิ่งกีดขวาง

คำถาม:
1. "มีรายการติดค้าง/บล็อกอยู่ไหม?"
2. "มีการฝ่าฝืนขีดจำกัด WIP ไหม?"
3. "เราช่วยจุดคอขวดได้ไหม?"
4. "มีการกระทำใดที่จะเคลียร์คิว?"

ตัวอย่างการประชุม:

SM: "ตรวจสอบการไหล - มีปัญหาไหม?"

Dev A: "7 รายการใน TESTING (ขีดจำกัด 3)"

QA Lead: "ทีมผมทดสอบได้แค่ 1 รายการต่อวัน"
"คิวสะสมเพิ่มขึ้น"

Dev B: "ผมช่วยได้ ต้องการทดสอบด้วยตนเองไหม?"

QA: "ใช่! กำลังทดสอบ UI features ต้องการทดสอบเบราว์เซอร์"

Dev B: "ผมจะทำการทดสอบเบราว์เซอร์ด้วยตนเองวันนี้"

"ปลดกำลังการผลิตการทดสอบอัตโนมัติ"

Dev C: "ผมจะเขียนสคริปต์การทดสอบอัตโนมัติ"

SM: "ดี การกระทำ:"

1. Dev B: ทดสอบด้วยตนเอง (ปลด QA สำหรับอัตโนมัติ)
2. Dev C: ตั้งค่าการทดสอบอัตโนมัติ
3. Review: อนุมัติเฉพาะรายการพร้อม
4. นักพัฒนา: อย่าเริ่มงานใหม่

ผลลัพธ์:
- คิวลดจาก 7 → 4 ภายในสิ้นวัน
- ทำงานร่วมกันดีขึ้น
- การไหลดีขึ้น

✅ ข้อเสนอแนะที่ 4:
Daily flow standup (5 นาที):
- ตรวจสอบขีดจำกัด WIP
- ระบุและปลดบล็อกจุดคอขวด
- ปรับลำดับความสำคัญรายวัน
คาดหวัง: การจัดการเชิงรุก, การปรับปรุงอย่างต่อเนื่อง

ส่วนที่ 4: สร้างแผนปรับปรุง (3 นาที)
แผนการดำเนินงาน
🎯 แผนงานการปรับปรุง

สัปดาห์ที่ 1: ทันที (สัปดาห์นี้!)

การกระทำที่ 1: ลดขีดจำกัด WIP
```
├─ IN PROGRESS: 5 → 3
├─ TESTING: 3 → 2
├─ REVIEW: 2 → 2 (คงไว้, เน้นย้ำ)
└─ ดำเนินการ: อัปเดต board วันนี้
```
คาดหวัง: เริ่มเห็นการปรับปรุงใน 2-3 วันข้างหน้า

การกระทำที่ 2: บังคับใช้วินัยการดึง

- ฝึกอบรม: อธิบายการทำงานแบบดึง
- กฎ: "ไม่สามารถเริ่มถ้าปลายทางบล็อก"
- ตรวจสอบรายวัน: ขีดจำกัดบังคับใช้หรือไม่?
- คาดหวัง: นักพัฒนาช่วย QA, คิวเคลียร์

การกระทำที่ 3: Daily Flow Standup

- เวลา: 10:30 น. (หลัง dev standup)
- ระยะเวลา: 5 นาที
- มุ่งเน้น: การไหล, จุดคอขวด, การกระทำ
- การตัดสินใจ: เคลียร์คิวหรือช่วยจุดคอขวด
- คาดหวัง: การจัดการเชิงรุก

สัปดาห์ที่ 2: ระยะสั้น (สัปดาห์หน้า)

การกระทำที่ 4: ฝึกอบรมนักพัฒนาข้ามสายงาน

- เลือก: นักพัฒนาที่แข็งแกร่ง 1 คน
- ฝึกอบรม: กระบวนการ QA, การเขียนการทดสอบ
- เป้าหมาย: ช่วย QA 50% ของเวลา
- ระยะเวลา: ฝึกทั้งสัปดาห์
- คาดหวัง: กำลังการผลิต QA +1 ภายินสิ้นสัปดาห์

การกระทำที่ 5: แผนการทดสอบอัตโนมัติ

- ตรวจสอบ: การทดสอบใดที่สามารถทำอัตโนมัติ?
- ตั้งค่า: กรอบการทดสอบอัตโนมัติ
- ลำดับความสำคัญ: การทดสอบมูลค่าสูงก่อน
- ระยะเวลา: เริ่มตั้งค่า
- คาดหวัง: แผนงานพร้อมสำหรับดำเนินการ

เดือนที่ 1-2: ระยะกลาง

การกระทำที่ 6: ดำเนินการทดสอบอัตโนมัติ

```
├─ ระยะที่ 1: การทดสอบ Login/auth (สัปดาห์ 1-2)
├─ ระยะที่ 2: ฟีเจอร์หลัก (สัปดาห์ 3-4)
├─ ระยะที่ 3: การทดสอบ Integration (สัปดาห์ 5-6)
├─ Coverage: เป้าหมาย 80%
└─ คาดหวัง: กำลังการผลิต QA +100%
```

การกระทำที่ 7: ติดตามตัวชี้วัด

```
├─ ติดตาม: เวลานำ, cycle time, throughput
├─ รายสัปดาห์: ทบทวนแนวโน้ม
├─ ปรับ: ขีดจำกัด WIP ถ้าจำเป็น
└─ คาดหวัง: การปรับปรุงอย่างต่อเนื่อง
```

📊 เกณฑ์ความสำเร็จ

ก่อน:

❌ เวลานำ: 11 วัน

❌ Throughput: 0.6 รายการ/วัน

❌ การฝ่าฝืน WIP: รุนแรง

❌ ขวัญกำลังใจทีม: ต่ำ (หงุดหงิดจากจุดคอขวด)

หลังสัปดาห์ที่ 1:

✓ เวลานำ: 6-8 วัน (ปรับปรุง 30%)

✓ Throughput: 1 รายการ/วัน (ปรับปรุง 50%)

✓ การฝ่าฝืน WIP: ไม่มี (บังคับใช้)

✓ ขวัญกำลังใจทีม: ดีขึ้น (กำลังดำเนินการ)

หลังเดือนที่ 1:

✓ เวลานำ: 3-4 วัน (ปรับปรุง 70%)

✓ Throughput: 2 รายการ/วัน (ปรับปรุง 200%)

✓ การฝ่าฝืน WIP: ไม่เคย

✓ ขวัญกำลังใจทีม: สูง (การไหลราบรื่น)

หลังเดือนที่ 2:

✓ เวลานำ: 2-3 วัน (ปรับปรุง 80%)

✓ Throughput: 2.5+ รายการ/วัน (ปรับปรุง 300%)

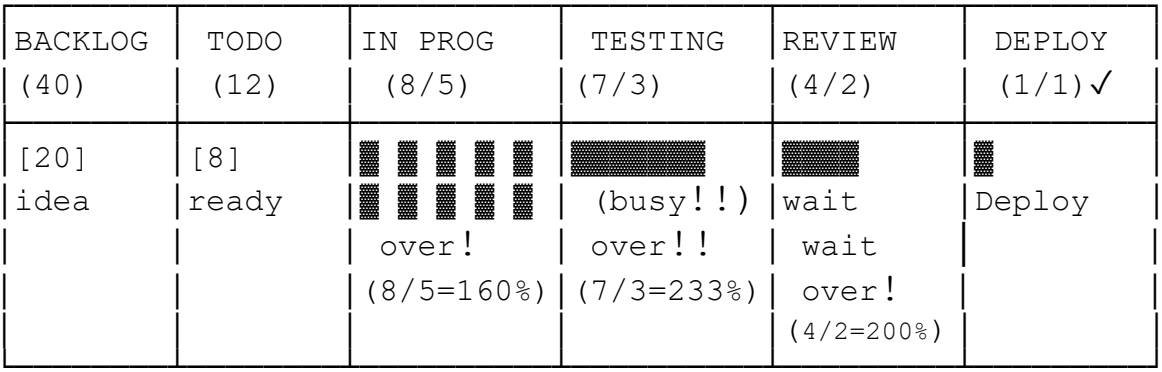
✓ ยั่งยืน: กระบวนการมั่นคง

✓ ทีมมีความสุข: การส่งมอบที่คาดการณ์ได้

ส่วนที่ 5: แสดงภาพสภาพที่ปรับปรุงแล้ว (2 นาที)

ก่อน vs หลัง

ก่อน (สภาพปัจจุบัน):



ปัญหา:

❌ คิวมหึมาใน TESTING (7 รายการ!)

❌ นักพัฒนานั่งรอหรือเริ่มงานใหม่

❌ Reviewers โอเวอร์โหลด (4 รายการ)

❌ จุดคอขวดทำให้เกิดผลกระทบแบบ cascade

❌ เวลานำ: 11 วัน 😞

❌ ทีมหงุดหงิด

หลังสัปดาห์ที่ 1 (ดำเนินการ)

BACKLOG (35)	TODO (8)	IN PROG (3/3)✓	TESTING (2/2)✓	REVIEW (2/2)✓	DEPLOY (1/1)✓
[35] wait	[8] ready work	▓ ▓ ▓ focus work (ถึงขีด)	▓ ▓ focus test (ถึงขีด)	▓ ▓ focus Review (ถึงขีด)	▓ Deploy

การปรับปรุง:

✓ ไม่มีการฝ่าฝืน WIP (ทั้งหมดถึงขีด, ไม่เกิน)

✓ นักพัฒนาช่วย QA (ลดคิว)

✓ คิวลดจาก 7 → 2

✓ รายการไหลราบรื่น

✓ Reviewers ทำทัน

✓ เวลานำ: 6-8 วัน (ดีขึ้น 30%) 🎉

✓ ทีมมีส่วนร่วมและช่วยเหลือ

หลังเดือนที่ 1 (ยั่งยืน):

BACKLOG (60)	TODO (10)	IN PROG (3/3)✓	TESTING (2/2)✓	REVIEW (2/2)✓	DEPLOY (1/1)✓
[60] idea + prio update	[10] ready work	▓ ▓ ▓ focus flow	✓ ✓ testing efficiency auto +yourself)	✓ ✓ smooth Review (fast)	✓ Deploy Deploy daily

ประโยชน์:

✓ ไม่มีจุดคอขวด (กำลังการผลิตสมดุล)

✓ การไหลคงที่ (คาดการณ์ได้)

✓ เวลานำ: 3-4 วัน (ดีขึ้น 70%!) 🚀

✓ Throughput: 2+ รายการ/วัน (ดีขึ้น 300%!)

✓ คุณภาพ: การทดสอบอัตโนมัติรับประกันความสม่ำเสมอ

✓ ทีม: มีความสุข, ก้าวที่ยั่งยืน

✓ การ Deploy: รายวัน (continuous delivery!)

ส่วนที่ 6: การอภิปรายของทีมและถาม-ตอบ (2 นาที)

ประเด็นอภิปราย

คำถามผู้อำนวยความสะดวก:

Q1: "ทำไม TESTING ถึงเป็นจุดคอขวด?"

A: "รายการมาถึงมากกว่าที่ QA ประมวลผลได้"

"ขีดจำกัด WIP (3) เกิน (7 รายการ)"

"สร้างคิวที่บล็อกทุกอย่าง"

Q2: "ระบบดึงช่วยอย่างไร?"

A: "นักพัฒนาหยุดดันงานเมื่อถึงขีดจำกัด"

"พวกเขาช่วยจบงานปัจจุบันแทน"

"เคลียร์คิวจากล่างขึ้นบน"

"การไหลราบรื่น"

Q3: "ทำไมต้องลดขีดจำกัด WIP?"

A: "บังคับให้ทีมมุ่งเน้นที่การจบงาน"

"ป้องกันโอเวอร์โหลดที่จุดคอขวด"

"แสดงภาพปัญหาทันที"

"สลับบริบทน้อยลง"

Q4: "วัดการปรับปรุงอย่างไร?"

A: "เวลานำ: เวลาจากสร้างถึง deploy"

"Cycle time: เวลาจากเริ่มถึงเสร็จ"

"Throughput: รายการที่เสร็จต่อวัน"

"ติดตามแนวโน้มตลอดสัปดาห์"

Q5: "ถ้าลด WIP ไม่ได้?"

A: "เพิ่มกำลังการผลิต (จ้าง/ฝึก)"

"ทำอัตโนมัติ (ทดสอบ, deployments)"

"ปรับปรุงกระบวนการ (ลดของเสีย)"

"หรือยอมรับเวลานำที่นานขึ้น"

Q6: "นานแค่ไหนถึงเห็นการปรับปรุง?"

A: "วันที่ 1-2: คิวเริ่มเคลียร์"

"สัปดาห์ที่ 1: การปรับปรุง 30% มองเห็นได้"

"เดือนที่ 1: การปรับปรุงที่ยั่งยืน 70-80%"

"เดือนที่ 2: ปรับให้เหมาะสมและยั่งยืน"

ข้อผิดพลาดทั่วไปที่ควรหลีกเลี่ยง

❌ ข้อผิดพลาดที่ 1: เพิ่มขีดจำกัด WIP

ผิด: "บล็อกมากเกินไป, เพิ่มขีดจำกัด!"

ถูก: "บล็อก? หาจุดคอขวด, ช่วยเคลียร์"

❌ ข้อผิดพลาดที่ 2: ไม่สนใจตัวชี้วัด

ผิด: "ดูเหมือนโอเค, ไม่ต้องวัด"

ถูก: "วัดเวลานำ/throughput, ใช้ข้อมูล"

❌ ข้อผิดพลาดที่ 3: ดันงานมากขึ้น

ผิด: "คิวสะสม? ดันรายการเพิ่ม!"

ถูก: "จบงานปัจจุบัน, แล้วค่อยดึงงานใหม่"

❌ ข้อผิดพลาดที่ 4: ตำหนิบุคคล

ผิด: "QA ช้า, เปลี่ยนพวกเขา"

ถูก: "ระบบมีจุดคอขวด, แก้กระบวนการ"

❌ ข้อผิดพลาดที่ 5: ไม่มีการสื่อสาร

ผิด: "แค่ลดขีดจำกัด, บังคับอย่างเข้มงวด"

ถูก: "อธิบายว่าทำไม, ฝึกทีม, ทำแบบร่วมมือ"

✅ หลักการสำคัญ:

Kanban = การปรับปรุงระบบ, ไม่ใช่ตำหนิบุคคล

มุ่งเน้นที่การไหล, ไม่ใช่ความพยายาม

ผลการเรียนรู้

นิสิตสามารถ:

☑ เข้าใจขีดจำกัด WIP

- ✓ รู้ว่าทำไมขีดจำกัด WIP สำคัญ
- ✓ ระบุการฝ่าฝืน
- ✓ คำนวณผลกระทบ

☑ ระบุจุดคอขวด

- ✓ หาคอลัมน์ที่ถูกจำกัดมากที่สุด
- ✓ ติดตามสาเหตุหลัก
- ✓ เข้าใจผลกระทบแบบ cascade

☑ คำนวณตัวชี้วัด Kanban

- ✓ เวลานำ (เวลาที่ใช้เสร็จ)
- ✓ Cycle time (เวลาทำงานเท่านั้น)
- ✓ Throughput (รายการต่อวัน)

☑ เสนอแนะการปรับปรุง

- ✓ แนะนำหลายแนวทาง
- ✓ จัดลำดับความสำคัญการกระทำ
- ✓ ประมาณผลกระทบ

☑ สร้างแผนการดำเนินงาน

- ✓ ระยะเวลาสำหรับการเปลี่ยนแปลง
- ✓ ตัวชี้วัดความสำเร็จ
- ✓ ความรับผิดชอบของทีม

☑ ติดตามการปรับปรุงการไหล

- ✓ ติดตามตัวชี้วัดตลอดเวลา
- ✓ ปรับขีดจำกัด WIP
- ✓ การปรับปรุงอย่างต่อเนื่อง

☑ ชื่นชมประโยชน์ Kanban

- ✓ ขั้นตอนการทำงานที่มองเห็นได้
- ✓ การมองเห็นจุดคอขวด
- ✓ การส่งมอบอย่างต่อเนื่อง

✓ การจัดทีมให้สอดคล้องกัน

📚 **เอกสารอ้างอิง**

สำหรับศึกษาเพิ่มเติม

📖 ไฟล์ที่เกี่ยวข้อง:

- KANBAN.md (พื้นฐาน Kanban)
- SCRUM-agile-ceremonies.md (Daily standup)
- SCRUM-velocity.md (ตัวชี้วัดและ throughput)

📊 เครื่องมือสำหรับฝึกฝน:

- Draw.io (สร้างไดอะแกรม Kanban board)
- Trello (ฝึก Kanban board จริง)
- Excel (ติดตามตัวชี้วัดตลอดเวลา)

🎬 วิดีโอ:

- "Kanban WIP Limits Explained" (5 นาที)
- "Bottleneck Analysis" (8 นาที)
- "Flow Metrics" (6 นาที)

📝 โลกจริง:

- ลอง Kanban ในโครงการถัดไปของคุณ
- ติดตามตัวชี้วัดรายสัปดาห์
- ปรับขีดจำกัด WIP ตามข้อมูล

สรุป

Activity 2: การวิเคราะห์การไหลของงาน Kanban

✅ การวิเคราะห์เสร็จสมบูรณ์:

- ระบุการฝ่าฝืน WIP (7/3 ใน TESTING!)
- พบสาเหตุหลัก (จุดคอขวด QA)
- คำนวณตัวชี้วัด (เวลานำ 11 วัน)

✅ ข้อเสนอแนะที่ทำ:

1. ลดขีดจำกัด WIP (บังคับให้มุ่งเน้น)
2. บังคับใช้วินัยการดึง (ช่วยจุดคอขวด)
3. เพิ่มกำลังการผลิต QA (ฝึกอบรมข้ามสายงาน)
4. การจัดการการไหลรายวัน (เชิงรุก)

✅ แผนการปรับปรุง:

- สัปดาห์ที่ 1: ปรับปรุง 30% (เวลานำ: 11→8 วัน)
- เดือนที่ 1: ปรับปรุง 70% (เวลานำ: 11→3-4 วัน)
- เดือนที่ 2: การส่งมอบที่ยั่งยืน

✅ การติดตามตัวชี้วัด:

- เวลานำ, cycle time, throughput
- การติดตามรายสัปดาห์
- การตัดสินใจที่ขับเคลื่อนด้วยข้อมูล

✅ การเรียนรู้ที่สำคัญ:

- ขีดจำกัด WIP มีผลต่อการทำงาน
- จุดคอขวดแบบ cascade (ต้องแก้)
- การไหล > ความพยายาม
- Kanban แสดงปัญหาด้วยภาพ
