ENGSE206 · การกำหนดความต้องการและการออกแบบทางซอฟต์แวร์

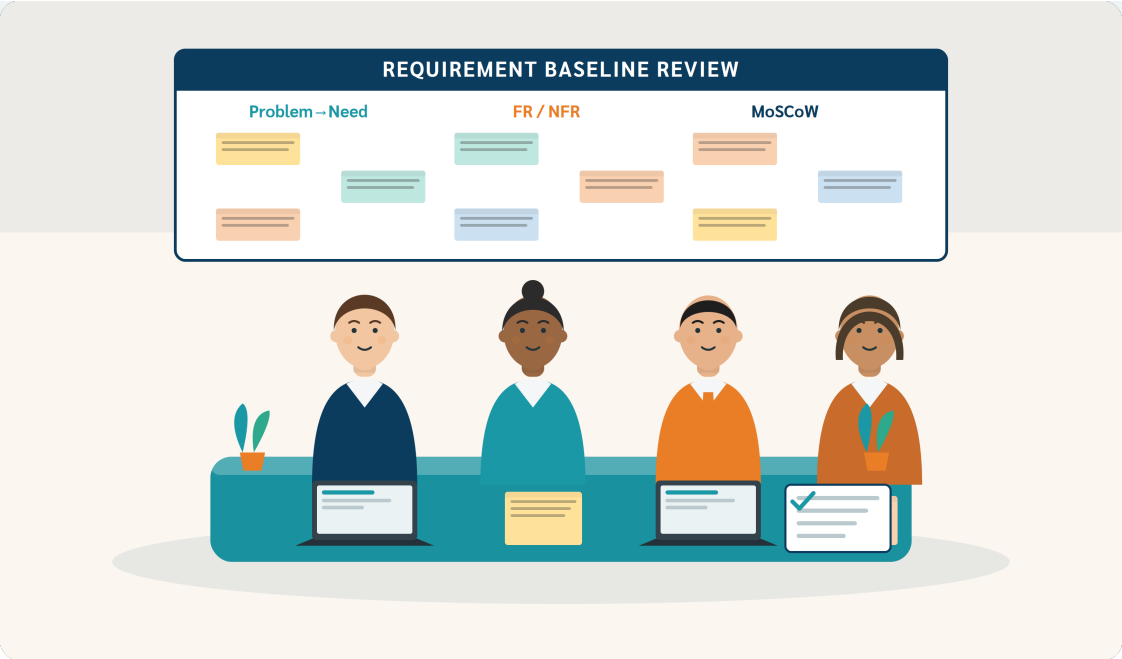
Software Requirements Specification and Design — มทร.ล้านนา (ดอยสะเก็ด)

กิจกรรมกลุ่มแบบทำเอง (Self-Directed Studio)

Requirement Baseline Review

& Readiness Gate

ทบทวน–ตรวจสอบ–ล็อก Baseline ของความต้องการ ก่อนเข้าสู่การทำ Requirement Model ในสัปดาห์ที่ 6

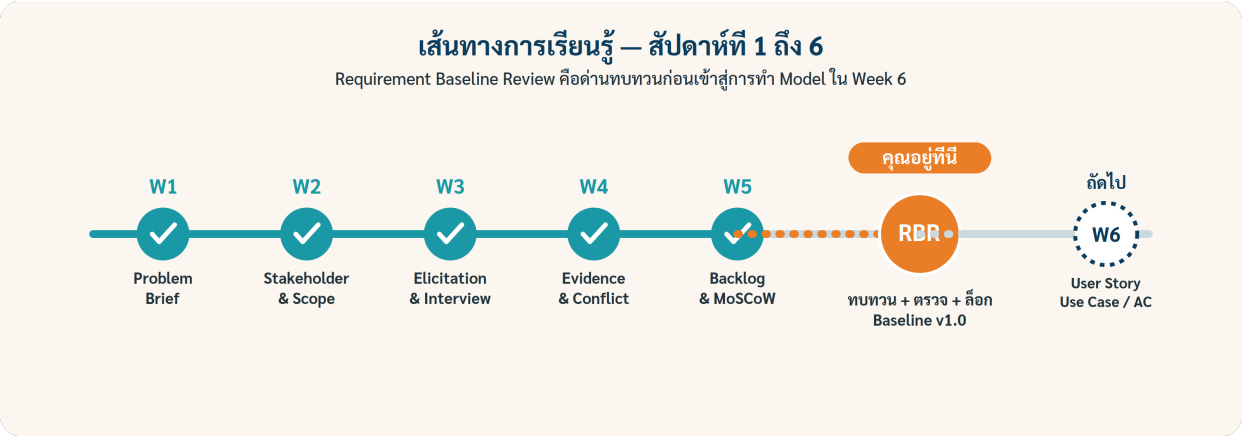


เอกสารนี้ออกแบบให้ทีมเดินเองได้ตั้งแต่ต้นจนจบ — หากติดขัด ดูหัวข้อ 13 “ถ้าติดปัญหา (ไม่มีอาจารย์ในคาบ)”

ข้อมูลกิจกรรมโดยย่อ

รายการ	รายละเอียด
ประเภทกิจกรรม	Consolidation Studio — ทบทวนและตรวจสอบงานของตัวเอง (ไม่ใช่บทเรียนใหม่)
ระยะเวลา	4 ชั่วโมง (ยืดหยุ่นได้ 3–5 ชม.) ทำจบภายในคาบเดียว
รูปแบบ	ทีมดำเนินการเอง 100% — อาจารย์ติดอบรม ไม่อยู่ในคาบ เอกสารนี้พาทีมเดินเองได้ตั้งแต่ต้นจนจบ
ความรู้ที่ใช้	สัปดาห์ที่ 1–5: Problem Brief ⟶ Stakeholder/Scope ⟶ Elicitation ⟶ Evidence/Negotiation ⟶ Backlog/MoSCoW
ต่อยอดสู่	สัปดาห์ที่ 6: User Story / Use Case / Acceptance Criteria (อาจารย์สอนคาบถัดไป)
ส่งงานที่	Team Repo (จาก engse206-project-template) — รายละเอียดในหัวข้อ 11 “สิ่งที่ต้องส่ง”

1 ทำไมต้องทำกิจกรรมนี้



ภาพที่ 1 — เส้นทางสัปดาห์ที่ 1–5 (ทำเสร็จแล้ว) สู่ด่านทบทวน RBR และต่อไปยัง Week 6

มาถึงสัปดาห์ที่ 5 ทีมของเราสะสมชิ้นงานไว้มากแล้ว ทั้ง Problem Brief, Stakeholder & Scope, แผนสัมภาษณ์, หลักฐาน (evidence), และ Requirement Backlog ที่จัดลำดับด้วย MoSCoW กิจกรรมวันนี้ไม่ได้เพิ่มเนื้อหาใหม่ แต่เป็นการ “หยุดตรวจตัวเอง” เพื่อให้ทุกคนเข้าใจตรงกัน และมั่นใจว่าของที่ทำมามีคุณภาพ มีที่มา และเชื่อมโยงกันจริง ก่อนจะต่อยอดเป็น Model ในสัปดาห์ที่ 6

ในงานวิศวกรรมซอฟต์แวร์จริง ขั้นตอนนี้เรียกว่า Requirement Baseline Review — ทีมจะ “ล็อก” ชุดความต้องการให้เป็นฐาน (baseline) ที่ตกลงร่วมกัน ก่อนนำไปออกแบบต่อ นี่คือทักษะที่ทีมทำเองได้โดยไม่ต้องมีอาจารย์คอยบอกทีละขั้น

สิ่งที่ทีมจะได้กลับไปเมื่อจบกิจกรรม

- **Baseline v1.0** — ชุด requirement ที่ตรวจแล้วและ commit/tag ไว้ใน repo
- **สาย Traceability ครบ** — ทุก requirement สำคัญลากกลับไปถึงหลักฐานและ stakeholder ได้
- **รายการช่องว่างและคำถามค้าง (Gap & Open Questions)** — สิ่งที่ต้องถามอาจารย์หรือทำต่อใน Week 6

ตรวจความเข้าใจ (Self-check)

ตอบสั้น ๆ ในทีมก่อนไปต่อ:

- ต่างจาก "ทำงานสัปดาห์ใหม่" อย่างไร? (คำใบ้: วันนี้เราตรวจของเดิม ไม่ได้เก็บ requirement เพิ่ม)
- ทำไม "baseline" จึงสำคัญก่อนออกแบบ?

2 เป้าหมายการเรียนรู้ และของที่ต้องเตรียม

เป้าหมายการเรียนรู้ (เมื่อจบกิจกรรม ทีมจะสามารถ...)

#	ความสามารถ	เชื่อมกับสัปดาห์
1	ตรวจว่าเอกสาร requirement ของทีมครบและสอดคล้องกันหรือไม่	W1–W5
2	ไล่สาย traceability จากปัญหา ⟶ หลักฐาน ⟶ ความต้องการ ⟶ ลำดับความสำคัญ	W1, W2, W4, W5
3	ประเมินคุณภาพ FR/NFR ว่า "วัดได้ / ทดสอบได้ / ไม่กำกวม"	W5
4	ให้และรับ feedback ระหว่างทีมอย่างสร้างสรรค์ (peer review)	W3–W4
5	ล็อก baseline และบันทึกการตัดสินใจ/ช่องว่างอย่างเป็นระบบ	W4–W5

เปิดของให้พร้อมก่อนเริ่ม (Prerequisites)

ให้ทีมเปิดไฟล์เหล่านี้ใน Team Repo ค้างไว้ จะได้ไม่เสียเวลาหาระหว่างทำ:

เอกสาร/โฟลเดอร์	ใช้ทำอะไรในกิจกรรมนี้
CASE_CARD.md	อ้างอิงขอบเขตและ constraints เดิม กันการขยาย scope
docs/01-problem-brief-v0.1.md	ต้นทางของ pain point / goal
docs/02-stakeholder-context-scope.md	รายชื่อ stakeholder และ in/out scope
docs/04-evidence-log.md + evidence/week-03,04/	หลักฐานที่ใช้ยืนยัน requirement

เอกสาร/โฟลเดอร์	ใช้ทำอะไรในกิจกรรมนี้
docs/05-requirement-backlog.md	FR/NFR + MoSCoW ที่จะตรวจวันนี้
docs/08-validation-traceability.md	ที่จะบันทึกผลการไล่สาย traceability
project-management/ (worklog, decision-log)	บันทึกบทบาท การตัดสินใจ และ commit

เคล็ดลับ

ยังไม่ต้องแก้อะไรในช่วงเตรียม — แค่เปิดไฟล์และแบ่งบทบาท การแก้จริงจะทำรวมทีเดียวในช่วงที่ 5 เพื่อไม่ให้เอกสารปั่นป่วนระหว่างตรวจ

3 บทบาทในทีม

กำหนดบทบาท 5 อย่างนี้ (หมุนเวียนได้ระหว่างช่วง) ถ้าทีมมี 3 คน ให้ควบบทบาทได้ เช่น Facilitator ควบ Timekeeper และ Scribe ควบ Quality Checker

บทบาท	หน้าที่หลัก
Facilitator	คุมลำดับและเวลา พาทีมทำครบทุกช่วง ตัดสินใจเดินต่อเมื่อถึง checkpoint
Traceability Auditor	ไล่สายเชื่อมโยง Problem ⟶ Evidence ⟶ Need ⟶ FR/NFR ⟶ Priority
Quality Checker	ตรวจว่า FR/NFR แต่ละข้อวัดได้ ทดสอบได้ และไม่กำกวม
Scribe	จดผลตรวจ แก้เอกสาร ทำ commit และอัปเดต decision-log
Timekeeper	เตือนเวลาและ checkpoint นาทีที่ 45 / 100 / 155

4 ภาพรวมแผนการทำงาน 4 ชั่วโมง

ภาพที่ 2 — ลำดับ 7 ช่วง พร้อมเวลาและ checkpoint (ปรับยืดหยุ่นได้ตามคาบ 3–5 ชม.)

Checkpoint สำคัญ (ให้ Timekeeper ประกาศ)

- **นาทีที่ 45** — ต้องเริ่มลงมือ Traceability Audit แล้ว
- **นาทีที่ 100** — ต้องมีตาราง traceability + ผลตรวจคุณภาพเป็นรูปเป็นร่าง
- **นาทีที่ 155** — ต้องแก้เสร็จและพร้อมเข้าด่าน Readiness Gate

ปรับตามเวลา: ถ้ามีแค่ 3 ชม. ให้ลดช่วงที่ 1 และ 4 ลงครึ่งหนึ่ง และตรวจ traceability เฉพาะ requirement ที่เป็น “Must” · ถ้ามี 5 ชม. ให้เพิ่มการไล่สายให้ครบทั้ง Should และเติมตัวอย่าง acceptance criteria ล่วงหน้าเพื่อวอร์มเข้าสู่ Week 6

5 ช่วงที่ 1 · ตรวจคลังชิ้นงาน (Artefact Health Check) — 40 นาที

ไล่ทีละไฟล์ว่า “มีจริง / ครบหัวข้อ / อัปเดตล่าสุด” หรือไม่ ยังไม่ต้องตัดสินคุณภาพลึก แค่เช็กว่ามีของครบก่อน ให้ Scribe ทำเครื่องหมายสถานะในตารางนี้ (ทำสำเนาลงใน evidence/week-05/):

เอกสาร	ต้องมีอะไรอยู่ข้างใน	สถานะ
docs/01 Problem Brief	goal เชิงผลลัพธ์, pain points, stakeholder เริ่มต้น, NFR เริ่มต้น	☐ ครบ ☐ ต้องแก้
docs/02 Stakeholder/Scope	stakeholder map, context diagram, in/out scope, constraints	☐ ครบ ☐ ต้องแก้
docs/03 Elicitation/Interview	objectives, คำถาม 10–12 ข้อ, bias check	☐ ครบ ☐ ต้องแก้
docs/04 Evidence Log	หลักฐานติด tag, conflict + ผลเจรจา, requirement candidates	☐ ครบ ☐ ต้องแก้
docs/05 Backlog	FR/NFR + source + priority + acceptance measure	☐ ครบ ☐ ต้องแก้

ระวัง

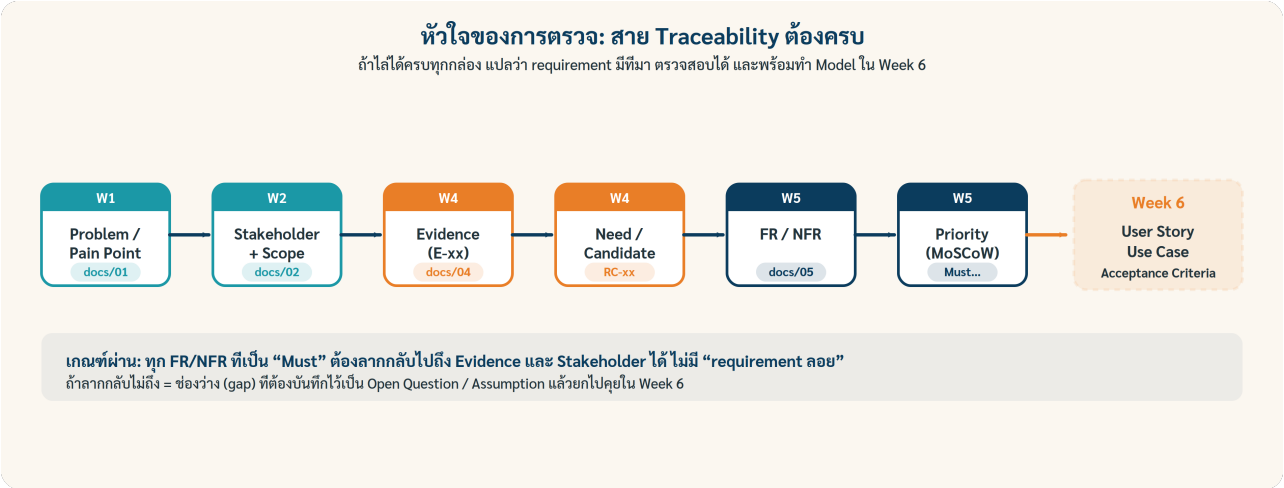
ถ้าเจอไฟล์ไหน “ไม่ครบ” ยังไม่ต้องหยุดแก้ตอนนี้ ให้จดไว้ในรายการ “ต้องแก้” แล้วไปต่อ การแก้ทั้งหมดจะทำรวมในช่วงที่ 5 เพื่อคุมเวลา

ตรวจความเข้าใจ (Self-check)

ตอบสั้น ๆ ในทีมก่อนไปต่อ:

- เอกสารช่วงไหนที่ทีม “ครบน้อยที่สุด”? เพราะอะไร?
- มีหัวข้อใดที่เคยข้ามไปตอนทำจริงไหม?

6 ช่วงที่ 2 · Traceability Audit — 55 นาที



ภาพที่ 3 — สาย traceability ที่ต้องไล่ให้ครบ ตั้งแต่ปัญหาไปจนถึงลำดับความสำคัญ

เลือก requirement ที่เป็น "Must" มา 3 ข้อ แล้วไล่ย้อนกลับทีละข้อว่ามาจากไหน ถ้าลากกลับไปถึง Evidence และ Stakeholder ได้ = ผ่าน ถ้าลากไม่ถึง = พบช่องว่าง (gap) ให้บันทึกเป็น Open Question ทีมที่มีเวลา 5 ชม. ให้ไล่ Should เพิ่ม

ตัวอย่างกลาง (Case X · SE Career Talk — งานจองห้อง/อุปกรณ์สำหรับกิจกรรมพูดคุยอาชีพ)

ใช้เป็นตัวอย่างวิธีคิดเท่านั้น ไม่ใช่คำตอบของกลุ่มใด — แต่ละทีมต้องไล่จากหลักฐานของตัวเอง:

ขั้น	เนื้อหา (Case X)	อ้างอิง	ตรวจ
Problem	ผู้จัดจองห้อง+อุปกรณ์หลายที่ ทำให้จองชนและตกหล่น	docs/01 · PB-02	✓
Stakeholder	ผู้จัดกิจกรรม, เจ้าหน้าที่ห้อง	docs/02 · ST-01/03	✓
Evidence	สัมภาษณ์: "เคยจองห้องชนกับอีกกิจกรรม"	docs/04 · E-07	✓
Need	อยากเห็นสถานะห้อง/อุปกรณ์ว่างในหน้าจอเดียว	RC-04	✓
FR/NFR	FR-05: ระบบต้องแสดงห้อง/อุปกรณ์ว่างตามช่วงเวลา	docs/05	✓
Priority	Must (กันจองชน = คุณค่าหลัก)	docs/05	✓

แบบฟอร์มให้ทีมกรอกเอง (ทำ 3 แถวสำหรับ 3 Must ของทีม)

Req ID	มาจาก Evidence (E-xx)	ผูกกับ Stakeholder	Need/Candidate (RC)	ลากครบ?
FR-__	E-__	ST-__	RC-__	☐ ครบ
FR-__	E-__	ST-__	RC-__	☐ ครบ
NFR-__	E-__	ST-__	RC-__	☐ ครบ

พบ gap? บันทึกลง docs/08-validation-traceability.md เป็นแถว “Open Question / Assumption” พร้อมเหตุผล แล้วยกไปคุยใน Week 6 — อย่าลบ requirement ทิ้งเฉย ๆ

ตรวจความเข้าใจ (Self-check)

ตอบสั้น ๆ ในทีมก่อนไปต่อ:

- มี requirement “Must” ข้อไหนที่ลากกลับไปหาหลักฐานไม่เจอบ้าง?
- ถ้าไม่มีหลักฐานรองรับ ควรลดเป็น Should/Could หรือทำอย่างไร?

7 ช่วงที่ 3 · Quality & MoSCoW Check — 40 นาที

อ่าน FR/NFR ทีละข้อ แล้วตรวจด้วยเกณฑ์คุณภาพ 4 ข้อ + ความสมเหตุสมผลของ MoSCoW + scope ไม่บวมเกิน Case Card

เกณฑ์	ถามตัวเองว่า...
วัด/ทดสอบได้ (Verifiable)	มี “ตัวเลข/เงื่อนไข” ที่ตรวจรับได้ไหม? (เช่น ภายในกี่วินาที, กี่คลิก)
ไม่กำกวม (Unambiguous)	อ่านแล้วตีความได้ทางเดียวไหม? เลี่ยงคำว่า “เร็ว/ง่าย/สะดวก” ลอย ๆ
หนึ่งข้อหนึ่งเรื่อง (Atomic)	มัดหลายเรื่องในข้อเดียวหรือเปล่า? (ถ้าใช่ ให้แยก)
มีที่มา (Traceable)	ผ่านช่วงที่ 2 มาแล้ว — มี source/stakeholder ชัด
MoSCoW สมเหตุสมผล	ทำไมข้อนี้ถึง Must? ถ้าตัดออกแล้วระบบยัง “ใช้ได้จริง” ไหม?
Scope ไม่บวม	เกินสิ่งที่ Case Card อนุญาตไหม? (เช่น เพิ่มระบบจ่ายเงินที่ out of scope)

ตัวอย่างแก้จากกำกวม ⟶ วัดได้ (Case X · กลาง)

ก่อน: *“ระบบต้องแสดงสถานะห้องได้เร็ว”*

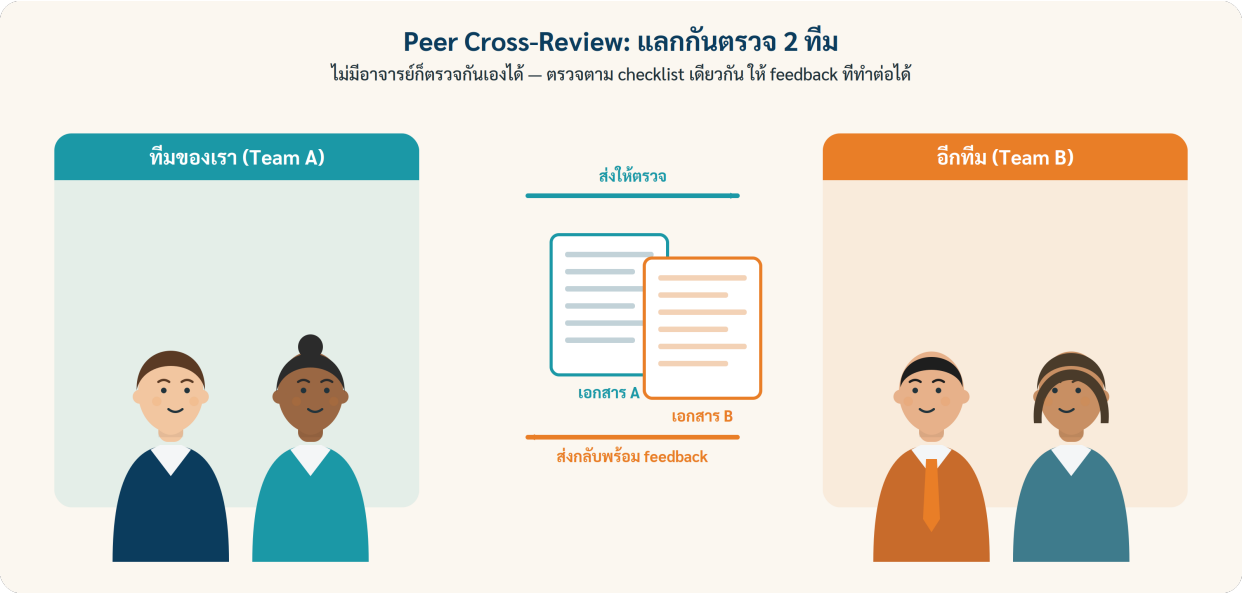
หลัง: “ระบบต้องแสดงสถานะว่าง/ไม่ว่างของห้องภายใน 3 วินาที หลังผู้ใช้เลือกช่วงเวลา”

ตรวจความเข้าใจ (Self-check)

ตอบสั้น ๆ ในทีมก่อนไปต่อ:

- FR/NFR ข้อไหนของทีมยัง “วัดไม่ได้” และจะแก้ถ้อยคำอย่างไร?
- มี Must ข้อไหนที่จริง ๆ ควรเป็น Should?

8 ช่วงที่ 4 · Peer Cross-Review — 35 นาที



จับคู่กับทีมข้าง ๆ แลกกันเปิด docs/05 + ตาราง traceability แล้วตรวจให้กันโดยใช้ checklist เดียวกัน (คนละ 15 นาที) เขียน feedback ที่ "ทำต่อได้จริง" และอ้าง ID เสมอ ถ้าไม่มีทีมข้าง ๆ ให้แบ่งกลุ่มตัวเองเป็น 2 ฝั่งสลับกันตรวจ

ใบตรวจข้ามทีม (Cross-Review Form)

สิ่งที่ตรวจ	ผ่าน?	ข้อเสนอแก้ (อ้าง ID)
ทุก Must มีสาย traceable ครบ	☐ ผ่าน ☐ ไม่	เช่น FR-05: เพิ่ม E-xx
FR/NFR วัด/ทดสอบได้	☐ ผ่าน ☐ ไม่	
ไม่มี requirement กำกวม/ซ้ำ	☐ ผ่าน ☐ ไม่	
Scope ตรงกับ Case Card	☐ ผ่าน ☐ ไม่	
MoSCoW มีเหตุผลรองรับ	☐ ผ่าน ☐ ไม่	

กติกาการให้ feedback ที่ดี

- อ้าง ID เสมอ (เช่น "FR-07 กำกวม") ไม่พูดลอย ๆ ว่า "อ่านยาก"
- เสนอทางแก้ที่ทำต่อได้ ไม่ใช่แค่บอกว่าผิด
- รับ feedback แบบไม่แก้ต่างทันที — จดไว้ก่อน แล้วค่อยตัดสินใจในช่วงที่ 5

ตรวจความเข้าใจ (Self-check)

ตอบสั้น ๆ ในทีมก่อนไปต่อ:

- feedback จากอีกทีมข้อไหน "ตรงจุด" ที่สุด?

สิ่งที่ตรวจ	ผ่าน?	ข้อเสนอแก้ (อ้าง ID)
• มีข้อไหนที่ทีมเราไม่เห็นด้วย และจะบันทึกเหตุผลไว้อย่างไร?		

9 ช่วงที่ 5 · แก้ และล็อก Baseline v1.0 — 30 นาที

รวมรายการ "ต้องแก้" ทั้งหมดจากช่วง 1–4 เลือกแก้ 3 อันดับแรกที่กระทบคุณภาพมากสุดให้เสร็จจริง ที่เหลือบันทึกเป็นงานค้าง (backlog) แล้วปิดท้ายด้วยการ commit + tag

ลำดับการปิดงาน

- แก้ถ้อยคำ FR/NFR ที่กำกวมใน docs/05 ให้วัดได้
- เติมสาย traceability ที่ขาดลง docs/08 (หรือใส่ Open Question ถ้ายังไม่มีหลักฐาน)
- บันทึกการตัดสินใจสำคัญลง project-management/decision-log.md (เลือกอะไร เพราะอะไร)
- อัปเดต team-worklog.md ว่าใครทำอะไร

ตัวอย่าง commit และ tag

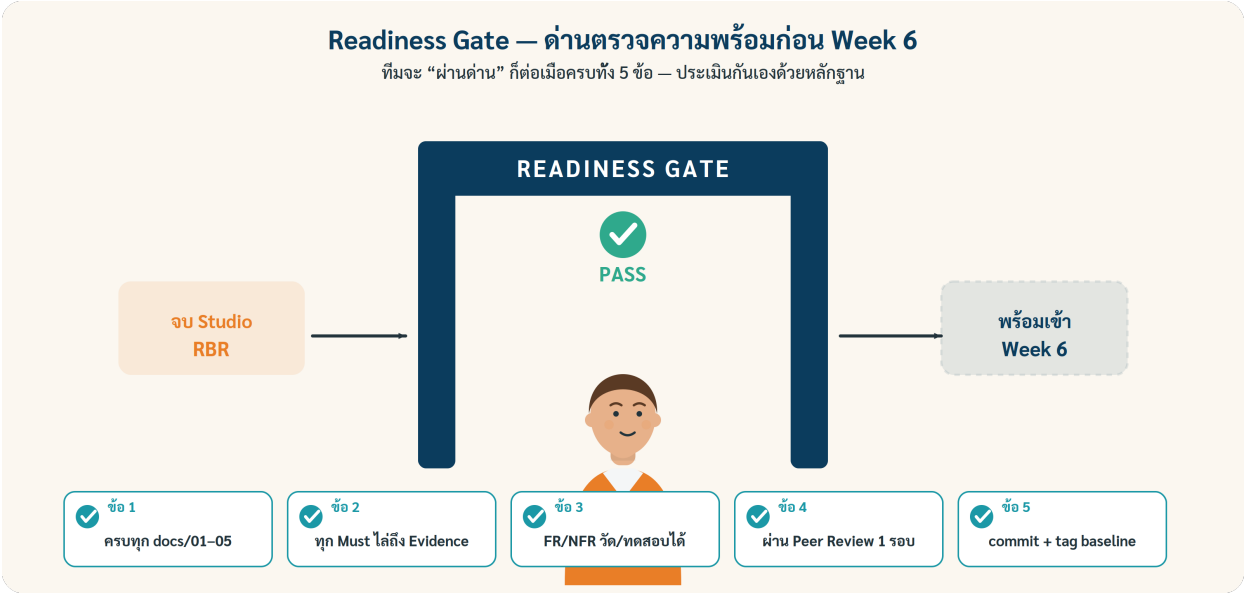
```
docs: revise FR-05/FR-07 to measurable criteria
docs: add traceability rows + open questions (docs/08)
git tag baseline-v1.0   # ล็อกฐาน requirement ก่อนทำ Model
```

ตรวจความเข้าใจ (Self-check)

ตอบสั้น ๆ ในทีมก่อนไปต่อ:

- 3 อันดับที่เลือกแก้ กระทบคุณภาพอย่างไร?
- งานค้างที่เหลือ ใครรับผิดชอบต่อ?

10 ช่วงที่ 6 · Readiness Gate และ Reflection — 25 นาที



ประเมินกันเองด้วยหลักฐานว่า "ผ่านด่าน" ครบทั้ง 5 ข้อหรือยัง ทำเครื่องหมายและแนบลิงก์/commit ที่พิสูจน์ได้

#	เกณฑ์ผ่านด่าน (Readiness Gate)	หลักฐาน
1	เอกสาร docs/01–05 ครบและอัปเดตล่าสุด	☐ (ลิงก์/commit)
2	ทุก requirement ที่เป็น Must ไล่ถึง Evidence + Stakeholder ได้	☐
3	FR/NFR ทุกข้อวัด/ทดสอบได้ ไม่กำกวม	☐
4	ผ่าน Peer Cross-Review อย่างน้อย 1 รอบ และแก้ตามที่ตกลง	☐
5	commit + tag baseline-v1.0 และอัปเดต worklog/decision-log	☐

Reflection รายคน (คนละ 3–4 บรรทัด เขียนลง feedback/15-individual-reflection.md)

- วันนี้ฉันเข้าใจอะไรชัดขึ้นเกี่ยวกับ requirement ของทีม?
- จุดที่ทีมเรายังอ่อนที่สุดคืออะไร และจะทำอย่างไรต่อ?
- คำถามที่อยากถามอาจารย์ในคาบหน้า (Week 6) คือ...

> **พรีวิว Week 6 (อาจารย์จะสอนคาบถัดไป)**
>
> นำ Baseline v1.0 นี้ไปแปลงเป็น **User Story, Use Case และ Acceptance Criteria** — ยิ่ง baseline วันนี้สะอาดและมี traceability ครบ งาน Week 6 จะยิ่งง่ายและเร็ว

11 สิ่งที่ต้องส่ง (Deliverables) และวิธีแนบลิงก์หลักฐาน

ไม่ต้องสร้างสัปดาห์ใหม่ — แก้ไฟล์เดิมใน Team Repo แล้ว **รวมลิงก์หลักฐานทั้งหมดไว้ในไฟล์เดียว** คือ `submissions/baseline-review-submission.md` (คนละไฟล์กับ week-05-submission.md ปกติ) เพื่อให้อาจารย์เปิดไฟล์เดียว คลิกตรวจได้ทุกชิ้น

> **หลักการเดียวที่ทำให้ตรวจง่าย: pin ลิงก์กับ tag baseline-v1.0**
>
> อย่าแนบลิงก์ที่ชี้ .../blob/main/... เพราะจะเลื่อนเมื่อทีมแก้ไฟล์ทีหลัง ทำให้อาจารย์เห็นคนละเวอร์ชันกับตอนส่ง ให้ชี้ .../blob/baseline-v1.0/... ซึ่งเป็นสแนปช็อตที่ทีมล็อกไว้ในช่วงที่ 5 — เปิดเมื่อไหร่ก็เห็นของ ณ ตอนส่งเสมอ อยากชี้ตรงหลักฐานเป๊ะ ๆ (เช่น requirement บรรทัดนั้น) เติม line anchor เช่น #L40-L52 ต่อท้าย URL ได้

ลำดับที่ทำให้ลิงก์นิ่ง (ทำตามนี้เท่านั้น)

```
# 1) กรอกไฟล์ submission ให้ครบก่อน (พิมพ์ …/blob/baseline-v1.0/… ได้เลย)
git add -A
git commit -m "submit(w05): lock requirement baseline v1.0"
git tag baseline-v1.0
git push
git push origin baseline-v1.0
```

เพราะ commit ไฟล์ submission **ก่อน** สร้าง tag ⟶ tag จะรวมไฟล์ submission ไว้ด้วย ลิงก์ทุกอัน (รวมลิงก์ที่วางใน Google Sheet) จึงชี้ที่ `baseline-v1.0` และเปิดเห็นของ ณ ตอนส่งได้ครบ

ในตารางด้านล่าง **...** แทน https://github.com/<org>/ENGSE206-GroupXX-ProjectName (ใส่ URL repo ของกลุ่ม)

ชิ้นงาน	ลิงก์ที่ต้องแนบ (ชี้ที่ tag baseline-v1.0)
ไฟล์ส่งงาน (วางลิงก์นี้ใน Google Sheet)	…/blob/baseline-v1.0/submissions/baseline-review-submission.md
Backlog ที่แก้แล้ว	…/blob/baseline-v1.0/docs/05-requirement-backlog.md
ตาราง Traceability + Gap	…/blob/baseline-v1.0/docs/08-validation-traceability.md
ผล Health Check + ใบ Cross-Review	…/tree/baseline-v1.0/evidence/week-05/baseline-review/
บันทึกการตัดสินใจ	…/blob/baseline-v1.0/project-management/decision-log.md
Worklog บทบาทรายคน	…/blob/baseline-v1.0/project-management/team-worklog.md
Reflection รายคน	…/blob/baseline-v1.0/feedback/15-individual-reflection.md

ชิ้นงาน	ลิงก์ที่ต้องแนบ (ชี้ที่ tag baseline-v1.0)

วิธีก็อปลิงก์ให้ได้ permalink (ระดับเริ่มต้น)

ลิงก์ทั้งไฟล์: เปิดไฟล์บน GitHub ตอนที่เลือก tag baseline-v1.0 อยู่ แล้วก็อป URL จากแถบที่อยู่ได้เลย

ลิงก์เจาะจงบรรทัด: คลิกเลขบรรทัดด้านซ้าย (กด Shift ค้างแล้วคลิกอีกบรรทัดเพื่อเลือกช่วง) ⟶ กดปุ่ม … ⟶ Copy permalink (GitHub จะ pin commit ให้เองและใส่ #Lxx-Lyy ให้)

วางลงชีต: คลิกเซลล์คอลัมน์ “Requirement Baseline Review & Readiness Gate” แถว Case ของกลุ่ม ⟶ Insert > Link (หรือ Ctrl/⌘+K) ⟶ วาง URL ไฟล์ส่งงาน ⟶ พิมพ์คำว่า “ส่งงานที่นี่” ⟶ Apply

ตรวจเองก่อนส่ง: เปิดลิงก์ในหน้าต่าง Incognito ถ้าเปิดได้ = ใช้ได้ (ถ้า repo เป็น Private อาจารย์ต้องถูกเชิญเป็น collaborator ก่อน จึงเปิดลิงก์ได้)

Definition of Done

- ผ่าน Readiness Gate ครบ 5 ข้อ พร้อมหลักฐาน
- commit ไฟล์ submission แล้วจึง tag baseline-v1.0 และ push ทั้ง code + tag
- ทุกลิงก์ใน submissions/baseline-review-submission.md ชี้ที่ baseline-v1.0 และเปิดได้จริง
- วางลิงก์ไฟล์ส่งงานในเซลล์ Google Sheet (คอลัมน์ RBR) แถว Case ของกลุ่มแล้ว
- ทุกคนมี worklog/commit หรือบทบาทที่ตรวจสอบได้ + Gap/Open Questions ครบ

12 ผลที่คาดว่าจะได้รับ

ระดับ	ผลลัพธ์
ตัวนักศึกษา	เข้าใจภาพรวม requirement ของทีมอย่างแท้จริง ฝึกอ่านงานอย่างมีวิจารณญาณ และให้/รับ feedback อย่างมืออาชีพ
ทีม	ได้ Baseline v1.0 ที่สะอาด มี traceability ครบ ลดงานแก้ย้อนหลังในสัปดาห์ถัด ๆ ไป
ความพร้อม Week 6	เข้าสู่การทำ User Story/Use Case/AC ได้ทันที เพราะ requirement นิ่งแล้ว
อาจารย์ (async)	ตรวจร่องรอยได้จาก commit, decision-log, reflection และ tag baseline — เห็นว่าทีมเข้าใจจริงแม้ไม่ได้อยู่ในคาบ

13 ถ้าติดปัญหา (ไม่มีอาจารย์ในคาบ)

สถานการณ์	ทำอย่างไร
เถียงกันไม่จบ	บันทึก 2 ทางเลือกใน decision-log เลือกทางที่ดีสุด ณ ตอนนี้เป็นการชั่วคราว ใส่เป็น Open Question ไว้ถามอาจารย์
เอกสารช่วงก่อนไม่ครบ/หาย	ทำฉบับย่อ (minimal) พอให้ไล่ต่อได้ แล้ว note ว่าจะเติมภายหลัง อย่าหยุดทั้งกิจกรรมเพราะไฟล์เดียว
ไม่มีทีมข้าง ๆ ให้ review	แบ่งกลุ่มตัวเองเป็น 2 ฝั่ง สลับกันตรวจด้วย checklist เดียวกัน
สงสัยเนื้อหา/นิยาม	อย่าเดา — จดเป็นคำถามรวมไว้ ส่งอาจารย์แบบ async และถามในคาบ Week 6
เวลาไม่พอ	โฟกัสเฉพาะ requirement ที่เป็น Must ให้ผ่าน Gate ก่อน แล้วค่อยขยายภายหลัง

ช่องทางถามอาจารย์ (async)

รวบรวมคำถามทั้งหมดไว้ที่ท้ายไฟล์ evidence/week-05/baseline-review/open-questions.md

แล้วส่งลิงก์ตามช่องทางที่อาจารย์กำหนด อาจารย์จะตอบและใช้เปิดคาบ Week 6

ภาคผนวก A Case X (กลาง) — ตัวอย่างสาย Traceability แบบเต็ม

Case X = “SE Career Talk: ระบบจองห้องและอุปกรณ์สำหรับกิจกรรมพูดคุยอาชีพ” ใช้สาธิตวิธีคิดเท่านั้น ไม่ใช่คำตอบของกลุ่มใด ทีมต้องสร้างจากหลักฐานของตนเอง

Req	ข้อความ requirement	Priority	Stakeholder	Evidence ⟶ Need
FR-01	ผู้จัดสร้างคำขอจองห้องพร้อมช่วงเวลาได้	Must	ผู้จัดกิจกรรม	E-02 ⟶ RC-01
FR-05	ระบบแสดงห้อง/อุปกรณ์ว่างตามช่วงเวลาภายใน 3 วินาที	Must	ผู้จัด/เจ้าหน้าที่	E-07 ⟶ RC-04
FR-08	ระบบแจ้งเตือนเมื่อมีคำขอจองชนช่วงเวลาเดียวกัน	Must	เจ้าหน้าที่ห้อง	E-07 ⟶ RC-05
NFR-02	ผู้ใช้ใหม่ทำรายการจองสำเร็จได้ภายใน 5 คลิก	Should	ผู้จัดกิจกรรม	E-11 ⟶ RC-09
NFR-04	ระบบเก็บ log การจอง/ยกเลิกเพื่อตรวจสอบย้อนหลัง	Should	ผู้ดูแลระบบ	E-13 ⟶ RC-12

อ่านแนวทแยง: Evidence (มีคนพูด/สังเกตเห็น) ⟶ Need (ผู้ใช้อยากได้อะไร) ⟶ Requirement (ระบบต้องทำอะไร วัดได้) ⟶ Priority (สำคัญแค่ไหน) ทุกแถวลากกลับได้ = traceable

ภาคผนวก B ใบตรวจสำหรับพิมพ์ (Checklists)

B.1 Cross-Review Checklist

- ทุก Must มีสาย traceable ครบ (Problem⟶Evidence⟶Need⟶FR/NFR⟶Priority)
- FR/NFR ทุกข้อวัด/ทดสอบได้ และไม่กำกวม
- ไม่มี requirement ซ้ำ/ขัดกันเอง
- Scope ตรงกับ Case Card ไม่บวมเกิน
- MoSCoW ทุกข้อมีเหตุผลรองรับ

B.2 Readiness Gate Checklist

- docs/01–05 ครบและอัปเดตล่าสุด
- ทุก Must ไล่ถึง Evidence + Stakeholder
- FR/NFR วัด/ทดสอบได้ ไม่กำกวม
- ผ่าน Peer Cross-Review ≥ 1 รอบ และแก้ตามที่ตกลง
- commit + tag baseline-v1.0 + อัปเดต worklog/decision-log

จบเอกสารกิจกรรม — ขอให้ทีมสนุกกับการตรวจงานของตัวเอง และพร้อมลุย Week 6