SPEC: G1 Form UX Redesign

สเปก handoff สำหรับ developer — ปรับ UX ฟอร์มกรอกรายการ G1 (บานกระจก ประตู/หน้าต่าง)

วันที่: 2026-06-11 | สถานะ: ดราฟพร้อม handoff | ไฟล์เป้าหมาย: public/calculator/index.html | ดราฟ demo: draft-G1-formux.html

1. ขอบเขตงาน

เปลี่ยนแค่ "วิธีกรอก" (control type) — ไม่แตะ logic/ราคา/ค่าที่ส่งเข้าใบ
ชิปและ stepper ต้อง sync ค่ากลับเข้า element เดิมที่ `readItem()` อ่าน (class .o-* / .i-* เดิม) — ห้ามเปลี่ยน value, class, หรือลำดับ option ใดๆ

กฎเหล็ก: readItem() อ่านค่าจาก `d.querySelector('.i-type').value`, `d.querySelector('.o-bottomrail').value` ฯลฯ — ต้อง element เหล่านี้ยังคงอยู่ใน DOM (hidden input หรือ select ที่ซ่อน) พร้อมค่าที่ถูกต้องเสมอ ก่อนที่ calcQuote() จะรัน

สรุปงาน

1. ย้าย .i-prod (แบบ) ขึ้นมาเป็นลำดับ 2 (ทันทีหลัง .i-group)
2. เปลี่ยน .i-type (ประเภท) จาก select → ชิป 2 ปุ่มใหญ่ (ไม่เกิน 52px สูง)
3. เปลี่ยน select ≤4 ตัวเลือก → ชิป (8 ช่อง รายละเอียดในตาราง)
4. ปุ่ม/ชิปทุกตัวสูง ≥40px (tap target มือถือ)

2. ลำดับช่องใหม่ (addItem template)

Before (ลำดับปัจจุบัน บรรทัด 2101–2127)

กลุ่มงาน → ด้าน/ตำแหน่ง → ~~ประเภท~~ → ~~แบบ (อันดับ 4!)~~ → กว้าง/สูง → บานทั้งหมด/ชุด

After (ลำดับใหม่)

กลุ่มงาน → แบบ (.i-prod) ขึ้น #2 → ประเภท (ชิป) ชิป → ด้าน/ตำแหน่ง → กว้าง/สูง → บล็อกฟ้า (บาน+ราง+ล็อค) → จำนวนชุด

HTML template ที่ต้องแก้ (addItem · บรรทัด 2099–2128)

> แก้เฉพาะลำดับ div ใน `d.innerHTML = ...` — ห้ามเปลี่ยน class ของ element ใดๆ
> ประเภทชิปใช้ `<input type="hidden" class="i-type">` แทน select (หรือซ่อน select แล้ว sync) — ดูแพทเทิร์นในหัวข้อ 4

```
// ลำดับใหม่ใน addItem() template (แก้เฉพาะส่วนนี้)
// ก่อน: group → position → type → prod → w/h → panels/qty
// หลัง: group → prod → type-chip → position → w/h → (slide-block) → qty

d.innerHTML =
  `<div class="top">...</div>` +
  `<div class="chgrid">` +

    // 1. กลุ่มงาน (คงเดิม)
    `<div class="full"><label>กลุ่มงาน</label>
      <select class="i-group">...</select></div>` +

    // 2. แบบ — ย้ายขึ้นมา (เดิมอยู่อันดับ 4)
    `<div class="full"><label>แบบ</label>
      <select class="i-prod">...</select></div>` +

    // 3. ประเภท — เปลี่ยนเป็นชิป (pattern ดูหัวข้อ 4)
    `<div class="full"><label>ประเภท</label>
      <div class="chip-group chip-2up" data-chip="i-type">
        <button type="button" class="chip on" data-val="door">ประตู</button>
        <button type="button" class="chip" data-val="window">หน้าต่าง</button>
      </div>
      <select class="i-type" style="display:none;">
        <option value="door">ประตู</option>
        <option value="window">หน้าต่าง</option>
      </select></div>` +

    // 4. ด้าน/ตำแหน่ง (ย้ายลงมา)
    `<div class="full"><label>ด้าน/ตำแหน่ง...</label>
      <input class="i-position" list="posPreset" ...></div>` +

    // 5. กว้าง/สูง (คงเดิม)
    `<div>...กว้าง...</div><div>...สูง...</div>` +
```

```
    // 6-N. ที่เหลือคงเดิมทุกอย่าง (panels/qty/opts/etc.)
    ...
```

3. ตารางตัดสิน: ช่องไหนเปลี่ยนเป็นชิป

> กฎ: select ≤4 ตัวเลือก → ชิป · select ≥5 ตัวเลือก → คง dropdown (หรือ dropdown + search)

ช่อง (class)	ตัวเลือก	Control ใหม่	เหตุผล	ตำแหน่งในโค้ด
`.i-type` (ประเภท)	2	**ชิป 2up (52px)**	2 ตัว · tap ได้เลย · แกนหลักของฟอร์ม	addItem L2103
`.o-bottomrail` (ราง)	2	**ชิป sm (44px)**	2 ตัว (กันน้ำ/เตี้ย) · ใช้ทุกงานบานเลื่อน	buildItemOpts L2334
`.o-slidelock` (ชุดล็อค)	2	**ชิป sm (44px)**	2 ตัว (มีกุญแจ/ไม่มี) · เฉพาะ itype=door	buildItemOpts L2345
`.o-cmech` (มือจับ Cmech)	3	**ชิป sm (44px)**	3 ตัว (ไม่ใส่/ฝัง/เมโทร)	buildItemOpts L2290
`.o-mosqcolor` (สีกรอบมุ้ง)	4	**ชิป sm (44px)**	4 ตัว (ตามสีบาน/ซาฮาร่า/KL พิเศษ/KL ลายไม้)	buildItemOpts L2469
`.o-mosqfabric` (ผ้ามุ้ง)	5	คง dropdown	5 ตัว เกินกว่า 4 · บางชื่อยาว (สแตนนิรภัย)	buildItemOpts L2467
`.o-solidlower` (แผ่นทึบล่าง)	3	**ชิป sm (44px)**	3 ตัว (ไม่มี/อลูลูกฟูก/คอมโพสิท) · เห็นทันที	buildItemOpts L2562
`.o-fcsides` (ครอบวงกบ)	3	**ชิป sm (44px)**	3 ตัว (ไม่มี/3 ด้าน/4 ด้าน) + ช่องกรอก .o-fcm ต่อท้าย	buildItemOpts L2565
`.o-beamm` (เสริมคาน)	4	**ชิป sm (44px)**	4 ตัว (ไม่มี/รุ่น1/รุ่น2/รุ่น3) · ราคาต่างชัดเจน	buildItemOpts L2375
รวม: 8 ช่องเปลี่ยนเป็นชิป + 1 ช่องคง dropdown		**ผลรวม 9 ช่อง (G1 scope)**		

ไม่เปลี่ยน (เกิน 4 ตัวเลือก หรือตัวเลือก dynamic): .i-group (7), .i-prod (40+), .i-color (เยอะ), .i-glass (เยอะ), .o-opentype (4 แต่ตัวเลือกยาว), .o-mosq (12+), .o-digi (เยอะ), .o-stainless (เยอะ), .o-mosqfabric (5)

4. HTML Pattern: select → ชิป (วิธีทำ)

กลยุทธ์: ซ่อน select เดิม (display:none) แล้วใส่ div ชิปด้านบน — ชิปแต่ละตัวเมื่อกดจะ set .value ของ select เดิม แล้ว fire Event('change') เพื่อให้ readItem / calcQuote รับรู้โดยไม่ต้องแก้ logic เลย

Pattern A: ชิป 2 ปุ่ม (ประเภท .i-type)

```
<!-- ใน addItem template — แทน select class="i-type" เดิม -->
<div class="full">
  <label class="field-label">ประเภท</label>

  <!-- ชิป UI -->
  <div class="chip-group chip-2up">
    <button type="button" class="chip on"
      data-val="door"
      onclick="chSetChip(this,'i-type')">ประตู</button>
    <button type="button" class="chip"
      data-val="window"
      onclick="chSetChip(this,'i-type')">หน้าต่าง</button>
  </div>

  <!-- select เดิม — ซ่อน · readItem ยังอ่านจากตรงนี้ -->
  <select class="i-type" style="display:none;">
    <option value="door">ประตู</option>
    <option value="window">หน้าต่าง</option>
  </select>
</div>
```

Pattern B: ชิปเล็ก (ราง, ล็อค, Cmech ฯลฯ)

```
<!-- ตัวอย่าง o-bottomrail (ใน buildItemOpts · L2334) -->
<!-- แทน: h += '<label class="opt">ราง <select class="o-bottomrail"...>...' -->

h += `<label class="opt" style="width:100%;">
  ราง
```

```
  <div class="chip-group chip-sm">
    <button type="button" class="chip ${ _brDef==='กันน้ำ'?'on':'' }"
      data-val="กันน้ำ" onclick="chSetChip(this,'o-bottomrail')">
      รางกันน้ำ
    </button>
    <button type="button" class="chip ${ _brDef==='เตี้ย'?'on':'' }"
      data-val="เตี้ย" onclick="chSetChip(this,'o-bottomrail')">
      รางเตี้ย 7 มม.
    </button>
  </div>
  <select class="o-bottomrail" style="display:none;" onchange="calcQuote();">
    <option value="กันน้ำ" ${ _brDef==='กันน้ำ'?'selected':'' }>รางล่าง (รุ่นกันน้ำ)</option>
    <option value="เตี้ย" ${ _brDef==='เตี้ย'?'selected':'' }>รางล่าง (รุ่นรางเตี้ย 7 มม.)</option>
  </select>
</label>`;
```

Helper function: chSetChip (เพิ่มใน JS global)

```
// เพิ่ม 1 function นี้ใน script หลัก (ทำงานกับทุกชิปในระบบ)
function chSetChip(btn, selectClass) {
  // หา .ch ที่ครอบ (scope ถูกต้อง ไม่ข้ามรายการ)
  const ch = btn.closest('.ch');
  if (!ch) return;
  const val = btn.dataset.val;

  // toggle active chip
  ch.querySelectorAll(`.chip[onclick*="${selectClass}"]`).forEach(function(c) {
    c.classList.remove('on');
  });
  btn.classList.add('on');

  // sync ค่าเข้า select เดิม + fire change → calcQuote รับรู้
  const sel = ch.querySelector('.' + selectClass);
  if (sel) {
    sel.value = val;
    sel.dispatchEvent(new Event('change', { bubbles: true }));
  }
}
```

ข้อควรระวัง selector: ชิปใน buildItemOpts จะอยู่ใน .i-opts ซึ่ง innerHTML ถูก rebuild ทุกครั้ง ที่ i-group หรือ i-prod เปลี่ยน — ชิปจะถูก regenerate อัตโนมัติ ไม่ต้อง hook พิเศษ แต่ต้อง set ค่า default ของชิปให้ตรงกับ _brDef /

_itypeNow ตอน build เสมอ

5. CSS ที่ต้องเพิ่มใน <style>

```
/* ===== CHIP CONTROLS (G1 UX Redesign) ===== */
.chip-group {
  display: flex;
  gap: 8px;
  flex-wrap: wrap;
  margin-top: 4px;
}
.chip {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  min-height: 44px;       /* tap target มือถือ */
  padding: 8px 14px;
  border: 1.5px solid var(--line);
  border-radius: 10px;
  background: #fff;
  color: var(--ink);
  font-size: 14px;
  font-family: inherit;
  font-weight: 600;
  cursor: pointer;
  line-height: 1.3;
  text-align: center;
  transition: background .12s, border-color .12s;
  -webkit-tap-highlight-color: transparent;
}
.chip:active { background: var(--redbg); }
.chip.on {
  background: var(--red-dark);
  border-color: var(--red-dark);
  color: #fff;
}

/* ประเภท — 2 ปุ่มใหญ่เต็มแถว */
.chip-2up .chip {
  flex: 1;
```

```
  font-size: 16px;
  min-height: 52px;
}

/* ชิปเล็ก (option ใน .i-opts) */
.chip-sm .chip {
  min-height: 40px;
  padding: 6px 12px;
  font-size: 13px;
}
```

> เพิ่ม CSS ต่อท้าย `<style>` ที่มีอยู่แล้ว (ไม่ทับ rule เดิม)
> ตัวแปร `--red-dark` มีอยู่แล้วในโค้ด (#B3151D) — ใช้ได้เลย

6. จุดโค้ดที่ต้องแก้ (พร้อม line reference)

จุด	บรรทัด	งาน	ความเสี่ยง
addItem template	2099–2128	1. เรียงลำดับ div ใหม่ (prod ขึ้นก่อน type) 2. เปลี่ยน .i-type select → chip + hidden select 3. เพิ่ม data-chip attribute บน wrapper	กลาง อย่าลืมซ่อน select เดิม ห้ามลบ
buildItemOpts — ราง	2334	เปลี่ยน o-bottomrail select → chip (Pattern B)	สูง _brDef ต้องตรง default ชิป
buildItemOpts — ชุดล็อค	2345, 2353	เปลี่ยน o-slidelock select → chip · มี 2 ที่ (banan_luen + PC Door)	สูง มี 2 branch แก้ให้ครบ
buildItemOpts — Cmech	2290	เปลี่ยน o-cmech select → chip (3 ตัว) · o-cmechcolor คง dropdown (7 ตัว)	กลาง
buildItemOpts — สีกรอบมุ้ง	2469	เปลี่ยน o-mosqcolor select → chip (4 ตัว)	กลาง
buildItemOpts — แผ่นทึบล่าง	2562	เปลี่ยน o-solidlower select → chip (3 ตัว) · logic show/hide .o-solidlower-wrap ยังอยู่ใน onchange ของ select ซ่อน	สูง show/hide sub-fields ต้อง trigger จาก chip onclick ด้วย

จุด	บรรทัด	งาน	ความเสี่ยง
buildItemOpts — ครอบวงกบ	2565	เปลี่ยน o-fcsides select → chip (3 ตัว) · ช่อง o-fcm (กรอกเอง) ยังคงอยู่ต่อท้าย	กลาง
buildItemOpts — เสริมคาน	2375	เปลี่ยน o-beamm select → chip (4 ตัว) · input o-beammlen ยังคงอยู่ต่อท้าย	กลาง
script global	ใหม่ (เพิ่ม)	เพิ่ม function chSetChip(btn, selectClass) — ดูหัวข้อ 4	ต่ำ function ใหม่ ไม่กระทบของเดิม

กรณีพิเศษ: o-solidlower — select เดิมมี inline onchange ที่ show/hide .o-solidlower-wrap และ .o-sl-colorwrap เมื่อชิปกด ต้อง dispatchEvent('change') จาก chSetChip เพื่อ trigger logic นี้ด้วย (chSetChip ที่เขียนไว้ในหัวข้อ 4 ทำสิ่งนี้แล้วผ่าน `sel.dispatchEvent(new Event('change', {bubbles:true}))`)

7. Before / After สรุป

Before (ปัจจุบัน)

- "แบบ" อยู่อันดับ 4 — เลื่อนหา
- "ประเภท" = dropdown popup 2 ตัว
- ราง/ล็อค/Cmech = dropdown แม้ตัวเลือกน้อย
- ทุก option อยู่ในระดับเดียวกัน (ยาว)
- tap target เล็ก (select height:32px)
- ไม่มี stepper → พิมพ์ตัวเลขอย่างเดียว

After (ดราฟใหม่)

- "แบบ" ขึ้นอันดับ 2 ทันที
- "ประเภท" = 2 ปุ่มใหญ่ tap ได้เลย (52px)
- 8 ช่อง (≤4 ตัวเลือก) เปลี่ยนเป็นชิป
- option พับใน details · ฟอร์มหลักสั้นลง
- ทุกชิปสูง ≥40px (มาตรฐาน mobile)
- จำนวนบาน/ชุด มี stepper ±

ผลลัพธ์คาดหวัง: เซลล์กรอกฟอร์ม 1 รายการ G1 ปกติ (บานเลื่อน ราคา+มุ้ง) บนมือถือ ลดจาก ~14 tap/swipe เหลือ ~8 tap — ลดเวลากรอกประมาณ 40%

8. ข้อควรระวัง / QA Checklist

⚠️ **readItem ต้องอ่านได้ครบ** — หลังแก้ให้ทดสอบ genQuote 1 รายการ G1 ตรวจว่า optSel มี bottomrail / slidelock / cmech / mosqcolor ถูกต้อง

⚠️ **i-type.value ต้องอัปเดตก่อน refreshItype()** — d.querySelector('.i-type').addEventListener('change') ยังทำงานปกติ เพราะ select ยังอยู่ใน DOM (แค่ซ่อน) + chip fire change event ผ่าน chSetChip

⚠️ **buildItemOpts rebuild ทุกครั้ง** — chip ที่อยู่ใน .i-opts จะถูก regenerate เมื่อ i-prod หรือ i-group เปลี่ยน → ต้อง set default class "on" ให้ถูกต้องในขณะ build HTML string (h += ...)

⚠️ **addItem chip (i-type) ไม่ถูก rebuild** — i-type chip อยู่ใน addItem template (ไม่ใช่ buildItemOpts) จึงถูกสร้างครั้งเดียวตอนเพิ่มรายการ และ default = "door" ถูกต้องแล้ว

✅ **ทดสอบ collapseOthers / toggleCh** — ฟังก์ชัน collapse ทำงานกับ .ch ทั้งหมด chip ที่เพิ่มเข้าไปไม่กระทบ เพราะอยู่ภายใน DOM เดิมอยู่แล้ว

✅ **ทดสอบ global default color/glass** — apply global defaults (L2133) อ่าน .i-color/.i-glass ไม่ใช่ chip ใหม่ ไม่กระทบ

✅ **o-solidlower show/hide** — dispatchEvent change จาก chip → select.onchange inline จะ trigger .o-solidlower-wrap และ .o-sl-colorwrap ได้ปกติ

9. แนวทางขยาย Pattern ไปกลุ่มอื่น (G2–G7)

> **กฎ generic ที่ทำซ้ำได้:** select ≤4 ตัวเลือก → chip · select ≥5 → คง dropdown
> ใช้ function `chSetChip` เดิม (global) ไม่ต้องเขียนใหม่ต่อ group

ตัวอย่าง G2 (ระแนง/รั้ว) ที่คาดว่า chip ได้:

- .o-hmode (ปิด/ต่อ) — 2 ตัว → chip การ์ดแบบ before/after (มีอยู่แล้วใน hmode-block)
- .o-bgdir (ทิศระแนง) — 2 ตัว (ตั้ง/นอน) → chip
- .o-bocdir (ทิศระแนงเปิดปิด) — 2 ตัว → chip
- .o-roofbatten (แป) — 2 ตัว (เดี่ยว/คู่) → chip [G3]
- .o-roofframe (โครงสร้าง) — 2 ตัว → chip [G3]

> **ขั้นต่อไป:** เมื่อ G1 ผ่าน QA แล้ว ให้ใช้ SPEC นี้เป็น template สำหรับ ทำ SPEC-G2, SPEC-G3 ฯลฯ โดยเปลี่ยนเฉพาะตาราง "ช่องที่เปลี่ยน" และ line reference