
Field Validation Decision Framework

Purpose:

To help product teams decide whether to use **standard required fields** or **custom validation logic with disabled actions** in Salesforce experiences (Service Cloud, Screen Flows, LWCs, etc.), ensuring clarity, accessibility, and user trust.

1. Guiding Principles

Principle	Description
 Human-Centered	Validation should help users complete tasks — not block them.
 Predictable	Validation behavior should be consistent across pages and flows.
 Clear Feedback	Messages must explain <i>what's wrong</i> , <i>why</i> , and <i>how to fix it</i> .
 Maintainable	Use built-in Salesforce validation where possible to reduce tech debt.
 Accessible	All validation messaging must meet WCAG 2.1 AA standards (ARIA role, visible, keyboard-navigable).

♦ 2. Core Decision Criteria

Question	If YES	If NO
Is the field always required for system logic or data integrity (e.g., Case Subject, Contact ID)?	→ Use “ Required ” at the component or field level.	→ Move to next question.
Does the field’s requiredness depend on other user inputs or context (conditional)?	→ Use custom validation (formula, decision element, or disabled button).	→ Move to next question.

Does the user need to see <i>why</i> it's required before filling it in?	→ Use custom inline message and/or disable submission until valid.	→ Use “Required” if it's straightforward.
Is the flow multi-screen or multi-step , where early required checks may frustrate users?	→ Delay validation — use custom validation before submission .	→ Use Required .
Do we need a custom tone or friendly message (e.g., HCD wording, plain language)?	→ Use custom validation .	→ “Required” is sufficient.
Will users enter this data later (e.g., in draft or save-for-later flow)?	→ Use custom validation and disabled button logic until ready.	→ Use Required .

3. Framework Summary Table

Scenario	Recommended Method	Example	Rationale
Always required, universally understood	✅ Required checkbox	Case Subject, Contact Name	Fast, standard validation
Conditional (only required if X)	⚙️ Custom Validation Formula	Phone OR Email required	Prevents unnecessary blocking
Multi-step input flow	⚙️ Custom Validation Before Submit	Loan application wizard	Better progressive disclosure
Needs specific, helpful feedback	⚙️ Custom Validation Message	“Please include a short summary so the next agent can triage correctly.”	Improves clarity
Complex logic across multiple fields	⚙️ Decision Element or Disable Next Button	Service Type requires matching Sub-Type	Ensures business rule accuracy

Prevent accidental submission	⚙️ Disabled Submit/Next Button until valid	“Save” disabled until all key fields are valid	Prevents frustration; supports accessibility
Backend dependency (record-level rules)	⚙️ Apex addError() or Flow Fault Path	Duplicate record prevention	Keeps logic server-side
Dynamic requiredness by role or record type	⚙️ Conditional Required Formula	Required for Agents, optional for Managers	Contextual behavior

♦ 4. UX Pattern: Disabled Button Validation

When using **custom validation + disabled buttons**, follow this UX pattern:

a. Visual Cues

- Display inline helper text near invalid fields.
- Use `aria-describedby` or `aria-invalid="true"` for accessibility.
- Gray out primary action until all required conditions are met.

b. Behavior Pattern

1. User begins entering data.
2. Validation checks run **on change** or **on blur**.
3. Submit button remains disabled until all rules pass.
4. On hover (or focus), explain why the button is disabled:

“Complete all required details before submitting.”

c. Error Message Tone

Tone

Example

Directive “Please enter your contact method before continuing.”

Supportive “We need one way to reach you — email or phone works.”

Contextual “Since you selected ‘Escalated,’ a Priority level is required.”

5. Recommended Governance Workflow

Step	Owner	Output
Identify validation need	Product / Designer	Short summary of logic or requirement
Evaluate decision criteria	Designer + Admin	“Required” or “Custom Validation” choice
Document in design spec	Designer	Validation table with rules, tone, and messages
Build & test	Dev / Admin	Field, Flow, or LWC logic implemented
Review accessibility	QA	Confirm all error states are screen-reader friendly
Track usability	Research / Support	Monitor error rates and user frustration points

6. Example Matrix for Team Reference

Field	When Required	Type	Error Message	Decision Owner
Case Subject	Always	Required checkbox	—	Product
Case Reason	Only if Priority = High	Custom Validation	“Please select a reason for high-priority cases.”	Product
Phone / Email	At least one	Custom Validation	“Provide at least one contact method.”	UX
Internal Comments	On Escalation	Disabled Submit Button	Button disabled until field completed	UX / Dev

SLA Due Date	When Status = In Progress	Required checkbox	—	Admin
--------------	---------------------------	-------------------	---	-------

7. Implementation Examples

Required Field

[Checkbox] Required

Error (auto): "Complete this field."

Custom Validation (Screen Flow Formula)

`ISBLANK({!Email}) && ISBLANK({!Phone})`

Message: "Please enter at least one contact method."

LWC Disabled Button

```
<lightning-button
  label="Submit"
  disabled={isInvalid}
/>
```

where `isInvalid = !email && !phone`

8. Quick Reference: When in Doubt

If the field must always be filled in | → **Use Required**

If the field depends on context, logic, or sequence | → **Use Custom Validation**

If users need clear human-readable messages | → **Use Custom Validation with messaging**

If accessibility or friendliness is the priority | → **Use Custom Validation and disable until valid**