

REMEMBER THESE

- **Application scope** isolates application files/data; **ACLs** secure user access.
- **UI Policies and Client Scripts** improve forms; they are not a security boundary.
- Client: **g_form**. Server: **current**, **previous**, **gs**. Always use server logic in **Script Includes**.
- **Before** changes current; **After** acts on related records; **Async** has no previous record (evaluates to null).
- A **flow** has a trigger + actions/logic; a **subflow** has inputs/outputs but no trigger.
- Import path: source → import set → transform map → target. **Coalesce** finds the target record.
- Use Git on non-production; deploy tested apps to production through App Repo or an approved update set.

1 APPLICATION FIT & LIFECYCLE DOMAIN 1

Start with planning: outcome, users, access, tasks, data sources, interactions, process, UI, subscriptions, and whether an existing app can be used or extended.

Phase	Primary decision
Plan	What problem, process, users, data, and security model?
Develop	Add data, experience, logic/automation, and security.
Test	Verify requirements, scripts, flows, ACLs, and integrations in non-prod.
Deploy	Publish a tested version through an approved deployment path.
Maintain	Review behavior, fix defects, and release a new tested version.

GOOD PLATFORM FIT
Structured records, repeatable work, approvals/automation, role-based access, auditability, reporting, or integration. First check the Store/Community and whether extending an existing application is better than duplicating it.

2 DATA MODEL, TABLES & RELATIONSHIPS DOMAIN 1

Model first: forms, lists, flows, reports, and integrations all depend on table and relationship design.

Concept	Meaning
Table	Entity/container for records.
Record	One row representing one item.
Field	Column with a defined data type.
Reference	One-to-many; place the reference on the "many" side.
Many-to-many	Junction table containing references to both tables.
Extension	Child inherits parent fields/functionality and adds its own.

DESIGN CHECKLIST

- Normalize; reference shared data instead of duplicating it.
- Choose type carefully: String, Number, Reference, Choice, Date/Time, Boolean.
- Mandatory only when required for integrity; index frequent search fields.
- Table extension can be selected only while creating the child table.
- Scoped tables use app namespace (**x_company_app_table**); global custom tables use **u_**.

3 APPLICATION SCOPE & CROSS-SCOPE DOMAIN 1 / DOMAIN 3

Scope is a namespace that identifies an application and restricts access to its files, tables, and logic. Private scoped apps can access their own resources by default.

Control	What it governs
Custom app record	Name/version/scope, JS standard, runtime tracking, role, menu, logo, dependencies.
Table access	Whether other scopes can select the table, perform CRUD, or use web services.
Cross-scope privilege	Source scope, target scope/name/type, operation, Requested/Allowed/Denied.
Restricted caller	Allows access to a particular source resource/scope from a protected target.

Table: read/write/create/delete Script Include: execute API access

FAST CONTRAST
Scope protects application resources from other applications. **ACLs** protect data/resources from users.

4 MODULES, FORMS & RECORD PRODUCERS DOMAIN 2

Element	Must know
Application menu	Container for modules; determines menu visibility via roles.
Module	Navigation entry to a link, record, list, page, or URL; can have its own roles.
Form	Displays one record. Layout is tied to a table + view combination.
Form Builder	Add/order fields, sections, annotations, formatters, related lists.
Record Producer	Catalog-style UI that creates a record in its target table.

FORM DESIGN RULES

- Adding a new field to a form creates a column on the underlying table.
- **Do not place the same editable field on multiple form sections** — can cause data loss and policy issues.
- Use different views for different audiences/use cases.
- Record Producer: use for task-based targets; do not use it to create Requested Item records.
- Target table must be in scope or allow cross-scope create access.

5 CLIENT-SIDE FORM LOGIC DOMAIN 2

Tool / type	Use
onLoad	Run when a form finishes loading.
onChange	React when one field value changes.
onSubmit	Validate or stop form submission.
onCellEdit	Validate edits made directly in a list.
UI Policy	Prefer for basic mandatory, visible, and read-only behavior.
Data Policy	Enforces mandatory/read-only beyond the form, including imports and web services.

getValue / setValue setMandatory setReadOnly setDisplay showFieldMsg

SERVER DATA ON THE CLIENT
Known at form load: Display Business Rule → **g_scratchpad**. Needed dynamically: async **GlideAjax** → client-callable Script Include. Avoid client GlideRecord and unnecessary **getReference()**.

List trap: normal Client Scripts and UI Policies apply to forms, not list editing. Use onCellEdit, Data Policy, Business Rule/ACL, or disable list editing.

6 SERVER-SIDE LOGIC & DEBUGGING DOMAIN 2

Business Rule	Best use
Before	Set/validate current before DB write; can abort.
After	Trigger events or update related records after the write.
Async	Longer work after the transaction; order not guaranteed; previous unavailable (null).
Display	Prepare server info for the form, commonly via g_scratchpad .

```
current / previous / gs / g_scratchpad
// Reusable server logic belongs in a Script Include.
// Do not use current.update() in Before – changes save automatically.
```

- Keep rules small, conditioned, and scoped to a table.
- Use Script Includes instead of global Business Rules; they load on request.
- Client-callable Script Includes are invoked through GlideAjax and require access.
- Wrap Business Rule code in a function to avoid global variable conflicts.

DEBUGGING
Script Debugger: session-only breakpoints, step execution, variables, call stack. **Session Log:** transaction/log output for Business Rules, Script Includes, and custom UI activity.

7 ACL SECURITY & ACCESS CHECKS DOMAIN 3

Record ACL = object (table/field) + operation + requirements. Create ACLs after elevating to **security_admin**.

Operation	Failure result
create	No New action; API insert blocked.
read	Record/field hidden; API retrieval blocked.
write	Field/record becomes read-only; API update blocked.
delete	No Delete action; API delete blocked.
execute	Blocks callable Script Include or REST endpoint execution.

PASS LOGIC
Within one matching ACL, user must satisfy role/security attribute, condition, and script. Field access requires applicable table-level and field-level checks. Empty/invalid ACLs deny by default. Application/menu roles control navigation visibility, not data access.

```
answer = gs.hasRole('x_app.user') && current.active == true;
```

8 FLOW DESIGNER / WORKFLOW STUDIO DOMAIN 4

Trigger → Actions / Flow Logic → Outputs

Component	Meaning
Flow	Triggered process with actions, logic, optional error handler, and data pills.
Subflow	Reusable process with inputs/outputs and no trigger .
Action	Reusable operation with defined inputs and outputs.
Spoke	Scoped app packaging actions/subflows for a product or integration.

Record created Updated Created/updated Scheduled Inbound email

- Flow tests bypass trigger conditions and perform real record changes — test only on non-prod.
- Prefer small single-purpose flows and reusable subflows; use records rather than raw Sys IDs.
- Use wait logic or scheduled triggers, not **gs.sleep()**. Keep loops under 1000 iterations.
- Run As can be System or initiating user; flow roles temporarily grant only required access.
- Legacy Workflow: know properties/activities/transitions, but Workflow Studio is preferred for new automation.

9 EVENTS, JOBS, EMAIL & PROPERTIES DOMAIN 4

Capability	Must know
System event	Register, queue, then respond with notification, script action, or flow.
Scheduled job	Run a report, generate records from a template, or run a script on a schedule.
Notification	Defines when to send, recipients, and message content.
Mail script	Reusable dynamic content: #{mail_script:script_name} .
Inbound email	Forward, reply, or new; inbound-email flows run before inbound actions.

```
gs.eventQueue('x_app.case.opened', current, parm1, parm2);
var value = GlideApplicationProperty.getValue('x_scope.setting', domainSysId);
```

INBOUND EMAIL TIP
A watermark associates an email with an existing task. A reply can update it; an unrecognized new message can create a record. Attachments are added to the first record produced.

10 IMPORT SETS & TRANSFORM MAPS DOMAIN 5

CSV / XLSX → Data Source → Import Set → Transform Map → Target

Component	Purpose
Data source	File, format, retrieval method, and import table.
Import set table	Staging table holding raw rows before transformation.
Transform map	Maps source fields/logic to a target table.
Coalesce	Determines whether a target record matches; no match normally inserts.
Field map	Maps one source column to a target field; can include script logic.

- **Run business rules:** also runs workflows/approvals/auditing; clearing uses setWorkflow(false).
- **Enforce mandatory:** No, Only Mapped Fields, or All Fields.
- **Copy empty fields:** lets empty source values overwrite target values.
- Use **Test Load 20 Records** to validate a data source before full import.

11 REST INTEGRATIONS DOMAIN 5

Verb	Expected intent	Common success
GET	Read/query; must not modify data.	200
POST	Create a new resource.	201
PUT/PATCH	Modify an existing resource.	200
DELETE	Remove a resource.	204

OUTBOUND REST MESSAGE

- Define endpoint, authentication, headers, and HTTP methods; a GET method is created automatically.
- Auth can be inherited from the parent message or overridden per method (Basic, OAuth 2.0).
- Use a MID Server to reach a REST provider on an internal network.
- **executeAsync()** does not wait for the provider response.

```
var r = new sn_ws.RESTMessageV2('Message name', 'get');
r.setStringParameterNoEscape('id', recordId);
var response = r.execute();
```

12 MANAGING APPS + EXAM TRAPS DOMAIN 6

Topic	High-value fact
App Repository	Publish tested scoped-app version → QA/Test → install approved version in Production.
Delegated Development	Lets a non-admin develop/deploy a specific app with app-specific permissions.
Git integration	Use on non-production; each app has its own repository. Commit local / apply remote / branches.
Production	Do not manage production through Git. Use App Repo, approved update set, or supported path.

TROUBLESHOOTING / EXAM TRAPS

Symptom	Likely issue / fix
Module visible, records denied	Navigation role passed; data ACL failed. Debug ACLs.
Rule works on form, not import/list	UI Policy/Client Script scope. Use Data Policy/server control/onCellEdit.
Cross-scope call blocked	Check table access, runtime tracking, and cross-scope privilege status.
Import creates duplicates	Missing/incorrect coalesce or poor source data.
REST returns 401/403	Authentication, API access, ACL, or scope issue; inspect response.
Flow surprises production data	Test flow on non-prod; testing performs real record changes.

Menu roles ≠ ACLs Scope ≠ user access UI Policy ≠ Data Policy Flow ≠ subflow