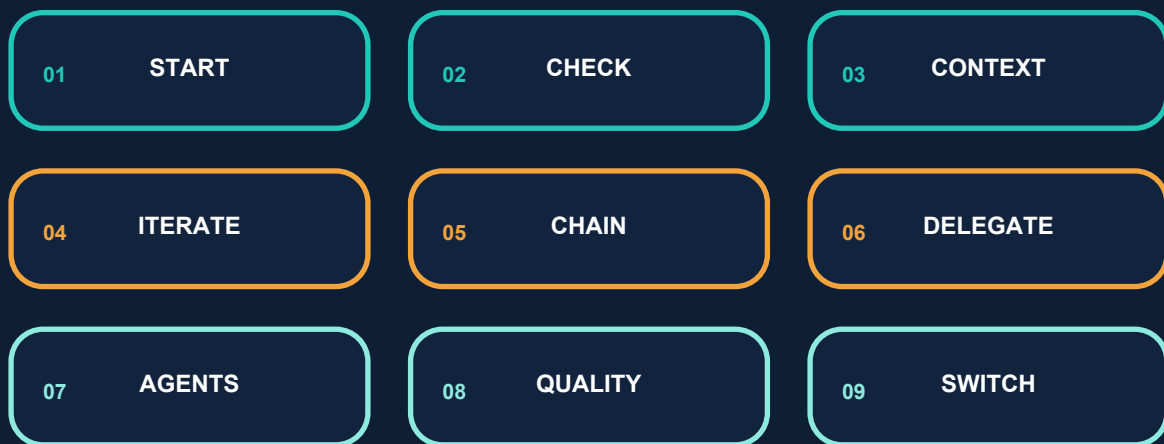
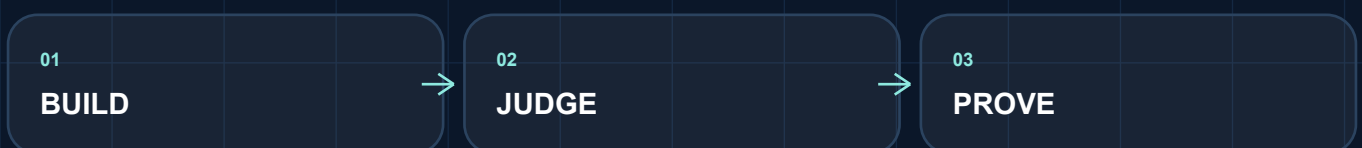


The 9 Skills

The durable habits behind capable AI and agent architects



Nine skill cards adapted from the ReInvent U knowledge base, each with a real-world rep, a visual model, and evidence that can appear on the capability record.



[Accessible web guide: reinventu.work/resources/field-kit/nine-skills](https://reinventu.work/resources/field-kit/nine-skills)

The skill tree at a glance

These skills are earned by use. A builder does not need all nine before starting.

01 BUILD



02 JUDGE



03 PROVE

01

Resourceful start

Use AI to find the next move without giving away the goal.

02

Skeptical checking

Assume fluent is not the same as true.

03

Context loading

Give the material and situation the tool cannot guess.

04

Prompt iteration

Treat the first result as the start of a conversation.

05

Tool chaining

Move work between tools with a purpose and a checkpoint.

06

Delegation thinking

Define what AI can do and what the human must own.

The skill tree at a glance

These skills are earned by use. A builder does not need all nine before starting.

01 BUILD



02 JUDGE



03 PROVE

07

Agent task design

Set a goal, steps, limits, and stop conditions.

08

Quality control

Know what good looks like before approving the output.

09

Learning velocity

Pick up a new tool quickly and transfer the method.

Field note: All nine run through Build-Judge-Prove.

Skill 01 - Resourceful start

When stuck, use AI to reveal routes forward without asking it to own the destination.

01 BUILD



02 JUDGE



03 PROVE

WHAT IT IS**Move from fog to options**

Name what you are trying to do, what you have tried, and the exact place you are stuck.

BUILDER REP**Ask for three next moves**

Request three small options with tradeoffs, then choose one and say why.

EVIDENCE**The choice is yours**

Keep the prompt, the options, and your reason for picking one.

Field note: Model: stuck point -> options -> human choice -> next action.

Skill 02 - Skeptical checking

AI can sound certain when it is wrong. Strong builders design a check before they trust the answer.

01 BUILD



02 JUDGE



03 PROVE

WHAT IT IS**Trust evidence, not tone**

Separate claims, calculations, links, and assumptions so each can be tested.

BUILDER REP**Catch one error**

Compare a claim with a primary source, run a calculation another way, or test the output with a known case.

EVIDENCE**The check record**

Record what you checked, how you checked it, what failed, and what changed.

Field note: Model: claim -> test -> result -> correction.

Skill 03 - Context loading

A tool cannot see the goal, audience, files, tone, or limits unless the builder provides them.

01 BUILD



02

JUDGE



03

PROVE

WHAT IT IS

Give the missing world

Supply the real material, intended user, useful examples, constraints, and privacy boundaries.

BUILDER REP

Run before and after

Try a vague prompt, then a context-rich prompt. Compare what became more useful.

EVIDENCE

A context pack

Keep the brief and a note naming which context changed the result most.

Field note: Model: material + audience + goal + constraints = useful context.

Skill 04 - Prompt iteration

The first prompt is a draft. Architects question, refine, push back, and narrow.

01 BUILD



02

JUDGE



03

PROVE

WHAT IT IS

Direct through dialogue

Inspect the first result and name the strongest part, weakest part, and next change.

BUILDER REP

Three-pass improvement

Run a shape pass, a stress-test pass, and a final fit pass. Change one variable at a time.

EVIDENCE

A visible trail

Keep the original, the prompts that changed it, and the final version.

Field note: Model: draft -> inspect -> direct -> compare -> keep.

Skill 05 - Tool chaining

One tool rarely does every job best. A chain makes handoffs deliberate instead of messy.

01 BUILD



02 JUDGE



03 PROVE

WHAT IT IS**Give each tool one job**

For example: collect input, organise it, draft an output, design it, then publish it.

BUILDER REP**Map a two-tool chain**

Name the input and output for each tool and place a human check between them.

EVIDENCE**The handoff map**

Show the chain, the checkpoint, and one problem the checkpoint caught.

Field note: Model: input -> tool -> human check -> tool -> ship.

Skill 06 - Delegation thinking

Good delegation is not 'do everything.' It is a clear transfer with ownership left in the right place.

01 BUILD



02 JUDGE



03 PROVE

WHAT IT IS

Split the work honestly

AI may explore, sort, draft, or transform. The builder owns the goal, facts, judgment, permission, and final approval.

BUILDER REP

Draw the ownership line

List tasks under AI can do, human must do, and human must check.

EVIDENCE

The disclosure block

State what the builder directed and what AI produced in plain language.

Field note: Model: delegate the repeatable; keep the consequential.

Skill 07 - Agent task design

An agent needs more than a prompt. It needs a goal, tools, a loop, limits, and a safe way to stop.

01 BUILD



02 JUDGE



03 PROVE

WHAT IT IS**Design bounded autonomy**

Define allowed actions, forbidden actions, required approval, retry limits, and a clear finish condition.

BUILDER REP**Run a small agent**

Use a low-risk job such as organising public information or proposing a family plan that an adult approves.

EVIDENCE**The agent card**

Keep the goal, tool list, boundaries, test cases, run log, and stop control.

Field note: Model: goal -> observe -> choose -> act -> check -> stop.

Skill 08 - Output quality control

Checking truth is essential. Quality control also asks whether the result is useful, clear, safe, and right for its user.

01 BUILD



02 JUDGE



03 PROVE

WHAT IT IS**Define good before review**

Write three to five observable success checks before asking AI to produce or improve anything.

BUILDER REP**Review with a rubric**

Use no numerical rating. Mark each check as passes, needs work, or cannot yet verify, then fix the highest-impact gap.

EVIDENCE**The approval note**

Keep the checks, result, change made, and reason it is now ready.

Field note: Model: define good -> inspect -> repair -> approve.

Skill 09 - Learning velocity

Tools will change. The durable skill is becoming useful with the next one without losing the method.

01 BUILD



02 JUDGE



03 PROVE

WHAT IT IS**Transfer the pattern**

Find the new tool's input, output, limits, sharing controls, and fastest safe practice task.

BUILDER REP**One-hour tool switch**

Use an unfamiliar approved tool to complete a small job, keeping notes on surprises and equivalents.

EVIDENCE**The transfer map**

Show old move, new move, what stayed the same, and what needed relearning.

Field note: Model: orient -> small rep -> compare -> transfer -> ship.