

Salesforce Certified Business Analyst

The exam contains 6 domains. This guide walks each one in the order of the most weighted, including information it covers, the terms you are expected to know, and the judgment the questions are really testing.

Where the questions come from — study top to bottom

Domain	Weight	Questions on a 60-question test
Collaboration with Stakeholders	23%	~14 of a 60-question test
Requirements	18%	~11 of a 60-question test
User Stories	18%	~11 of a 60-question test
Customer Discovery	17%	~10 of a 60-question test
Business Process Mapping	12%	~7 of a 60-question test
Development Support & User Acceptance	12%	~7 of a 60-question test

60 scored questions (plus a few unscored)	72% needed to pass	105 minutes
---	------------------------------	-----------------------

How to actually pass this exam

This guide is the a supplement study. It shows you where to aim and what the questions are really after, but reading it will not, on its own, get most people to 72%. These four things are how you actually get there.

1. **Take the practice tests to 80%+, not 72% once.** Sit full 60-question tests under time, then study every miss — not just the right answer, but why the tempting one was wrong. Use your domain breakdown to find weak spots, come back here to read up, and retake until you are clearing 80% across several attempts. A single lucky pass is not ready.
2. **Work Salesforce's official Trailhead path.** The [Business Analyst trailmix](#) and the [certification-prep module](#) cover the material at the depth the exam tests, and turn the terms in this guide into real understanding. Free, and it is what the exam was built against.
3. **Get hands-on in a free org.** Spin up a [Developer Edition](#) or a Trailhead Playground and do the work — write a few stories with acceptance criteria, map a process, open Optimizer on an org. The applied questions get much easier once you have actually touched the tools.
4. **Ground it in real BA situations.** This exam rewards people who have sat through requirements conversations and watched a project move from discovery to UAT. If that is not your day job, the case-study Trailhead modules and treating these practice scenarios as your own project help close the gap.

Collaboration with Stakeholders 23%

largest domain · ~46 questions in this bank

The biggest slice of the exam, and it is about people, not features. You find the right stakeholders, earn their trust, translate between the business and the technical team, pick the right way to draw out what they need, and steer competing priorities toward a decision everyone can live with.

Terms to know

Stakeholder analysis	Working out who is affected by or can influence the project, so the right voices are included.
Power / interest grid	Sorting stakeholders by influence and interest: manage closely, keep satisfied, keep informed, or monitor.
Trusted advisor	The relationship a BA earns by listening and giving honest guidance — not by agreeing to everything.
Liaison	The BA's role bridging business language and technical language so both sides share one understanding.
Elicitation techniques	Interviews, workshops, surveys, observation, document analysis, focus groups, brainstorming, prototyping — each fits a different situation.
RACI	A matrix naming who is Responsible, Accountable, Consulted, and Informed for each decision.
Consensus	Broad support people can commit to — not unanimity, and not a majority overruling the rest.
Register / comms plan	Records of who the stakeholders are and who needs what information, how often, through which channel.

What the questions are really testing

- The BA facilitates a decision; it rarely makes the call, sides with the loudest voice, or defers to the most senior person by default.
- When priorities collide, understand the business value behind each request before choosing — then work toward consensus.
- A stakeholder who was missed gets engaged now, even if it is late and inconvenient. Skipping them to protect the schedule is wrong.
- Match the technique to the need: interviews for candid depth, surveys for scale, observation for what people actually do, workshops to align groups.

Requirements 18%

~36 questions in this bank

Turning what stakeholders say into clear, agreed, testable requirements — and keeping them under control as the project moves. Expect questions on the difference between scope, requirements, and user stories, and on how requirements get validated, prioritized, and version-controlled.

Terms to know

Scope / requirement / story	Broad to specific: scope draws the boundary, a requirement states a needed capability, a user story frames it from a user's view.
Functional vs non-functional	Functional = what the system does; non-functional = qualities like performance, security, and usability.

The four levels	Business, stakeholder, solution, and transition requirements — from the goal down to what the solution does and what is needed to switch over.
Traceability matrix (RTM)	Links each requirement to its source, its design, and its tests. Backward traceability points to its origin.
Baseline & change control	The approved reference set of requirements; later changes go through impact analysis, not a quiet edit.
Validate / analyze / elicit	Elicit = gather, analyze = organize and de-conflict, validate = confirm it reflects the real need.
Gold plating	Adding features nobody asked for — a form of scope creep, not a nice extra.
SMART / testable	'Fast and user-friendly' is not a requirement; 'loads in under two seconds' is.

What the questions are really testing

- The business owns prioritization; the BA facilitates it. Developers and the BA do not set priority alone.
- After the baseline is set, a new requirement is assessed and routed through change control — never just added.
- Favor standard Salesforce functionality over a custom build when it meets the need.
- A good requirement is complete (covers exceptions, not just the happy path) and testable.

User Stories 18%

~36 questions in this bank

Writing and judging user stories, breaking big ideas into stories, and writing acceptance criteria that actually pin down 'done.' A lot of these questions hinge on one definition or one quality of a good story, so the vocabulary here pays off directly.

Terms to know

Story format	As a [role], I want [goal], so that [benefit]. The 'so that' carries the value — a story without it is weak.
INVEST	Independent, Negotiable, Valuable, Estimable, Small, Testable — the qualities of a good story.
Epic / theme / story / task	Epic = too big for one sprint, split into stories; theme = related work under a goal; task = a step to build a story.
Acceptance criteria	The conditions a story must meet to be accepted, often written Given / When / Then. Specific to that one story.
Definition of done	A shared quality bar that applies to every story — different from acceptance criteria.
Definition of ready	What makes a story clear, estimated, and small enough to pull into a sprint.
Backlog refinement	Clarifying, estimating, and ordering upcoming stories — where the BA often helps most.
Product owner	Owns and prioritizes the backlog; the BA supports rather than replaces them.

What the questions are really testing

- A bare technical task ('build an object with ten fields') is not a user story — a story names a role and a benefit.
- Acceptance criteria must be testable. 'Should feel fast' fails; a measurable condition passes.
- Acceptance criteria belong to one story; the definition of done spans all of them.
- A story too big for a sprint is an epic to split; a story you cannot estimate needs more clarity, not a guess.

Customer Discovery 17%

~34 questions in this bank

The front of the project: understanding the business, its goals, and how things work today before anyone designs a solution. It covers the BA's role in the implementation lifecycle, sizing up an existing Salesforce org, and the discovery techniques that surface the real picture.

Terms to know

Current state (as-is)	How a process actually works today — captured before any redesign.
Gap analysis	Comparing today's state to the desired future state to see what has to change.
Business objective	A measurable outcome the organization wants, like cutting case-close time by 20%.
Project charter	The document stating a project's purpose, objectives, and scope.
Salesforce Optimizer	A report on how an existing org is set up and where features go unused.
5 Whys	Asking 'why' repeatedly to get past symptoms to the root cause.
Demonstrating capabilities	Showing standard Salesforce features so stakeholders can picture what is possible and refine their needs.

What the questions are really testing

- Understand the underlying need before jumping to a specific solution, even when a stakeholder asks for one by name.
- Discovery produces understanding, not a built solution — deploying or configuring is a later phase.
- With multiple orgs or an unknown environment, assess first; consolidating or building comes after.
- Involve end users, not just executives — they know how the work is really done.

Business Process Mapping 12%

~24 questions in this bank

Drawing how work flows — today and in the redesigned future. Smaller domain, but symbol-and-notation heavy: know what the shapes mean, when a swimlane earns its keep, and how to capture the as-is honestly before designing the to-be.

Terms to know

Flowchart symbols	Oval = start/end, rectangle = task, diamond = decision, arrow = sequence flow (the order of steps).
Swimlane diagram	Lanes show which role or department does each step — the go-to when work hands off between people.
BPMN	Business Process Model and Notation. A pool = a participant/organization; lanes divide it by role.
Value stream mapping	Highlights which steps add value and which are waste.
SIPOC	Suppliers, Inputs, Process, Outputs, Customers — a high-level snapshot of a process.
As-is vs to-be	The current-state map versus the improved future-state design.
Architect Diagrams	Salesforce's own framework of diagram templates and notation.

What the questions are really testing

- 'Map the process' means diagram it — not build a Flow or write validation rules.
- Multiple roles and handoffs call for a swimlane; a branch in the path needs a decision point.
- The as-is map reflects how work is really done today, not the ideal or the future design.
- In the to-be, remove the redundancy and waste the current-state map exposed.

Development Support & User Acceptance **12%**

~24 questions in this bank

The back half of delivery: supporting the build team, then running user acceptance testing so the business confirms the solution meets its needs before go-live. Know who does UAT, where it fits, and how it differs from the other kinds of testing.

Terms to know

UAT	User acceptance testing — business/end users confirm the solution works for their real work.
Test cases from criteria	UAT scenarios come from the user stories and their acceptance criteria.
UAT vs unit/system/regression	Unit checks code, system checks the whole build technically, regression checks nothing broke, UAT checks it meets the need.
Entry / exit criteria	What must be true before UAT starts and before it is considered finished.
Sign-off & go/no-go	The business accepting the solution as ready, and the launch decision that rests on UAT results.
Hypercare	The stretch of heightened support right after go-live.
Backlog refinement (support)	The BA clarifying upcoming stories and criteria so the team can build without guessing.

What the questions are really testing

- UAT is run by business users, not developers, and it validates needs — not line-by-line code logic.
- It runs in a production-like test environment before go-live, never straight in production.
- The business owns the go/no-go decision, informed by the UAT results.
- UAT feedback gets captured and weighed against scope and value — not auto-built and not dismissed.

This is a companion to the practice tests. Use it to understand the material and find your weak spots. For registration, fees, and the official outline, see the Salesforce Certified Business Analyst exam guide on Trailhead.