

Business Analyst Starter Kit

2026

Eleven interview-ready templates, cheat sheets and frameworks - the practical work you'll actually do in your first year as a Business Analyst.

11

RESOURCES

16

INTERVIEW Qs

40+

PAGES

Inside the kit.

A quick map so you know where to jump.

- 01** BA Interview Kit
- 02** ATS-Friendly BA Resume Checklist
- 03** BRD Template
- 04** User Story Template
- 05** AI Prompts for Business Analysts
- 06** BA Career Roadmap 2026
- 07** Market Research Framework
- 08** Wireframing & Mockup Guide
- 09** Scrum & Agile Cheat Sheet
- 10** Requirement Gathering Framework
- 11** BRD vs FRD vs User Stories Guide

HOW TO USE THIS KIT

Read the interview prep sections back-to-back. Bookmark the templates - you'll re-open them the day you start your first project. Everything is written to be edited, not just read.

BA Interview Kit

16 high-signal questions with structured answers you can adapt to your own experience.

Q. Tell me about yourself.

Structure a 90-second pitch: (1) current role and years in industry, (2) 2-3 outcomes that map to what a BA does-discovery, documentation, stakeholder alignment, delivery, (3) why you're moving now and why this role/company. Avoid a chronological resume readout. End with a hook that invites a follow-up question.

Q. What is your process for gathering requirements?

Move stakeholders through five stages: (1) discovery interviews to capture the as-is, (2) elicitation workshops for the to-be, (3) documentation in a BRD or a story map, (4) validation with a walkthrough, (5) sign-off with acceptance criteria. Call out the specific techniques you use in each stage (see Resource 10).

Q. Walk me through a project you led end-to-end.

Use CAR: Context (business problem, KPI at stake, team shape), Action (what you did as the BA specifically-not what the team did), Result (metric moved, launch date, follow-on). Keep it under 3 minutes. Prepare 2-3 stories that let you swap the ending based on what the interviewer probes.

Q. How do you handle conflicting requirements from two stakeholders?

Never mediate in the moment. Document both positions in writing, map each to a business outcome, and quantify the trade-off (cost, risk, time, revenue). Bring both stakeholders back with the data and let the sponsor decide. Show the interviewer you separate advocacy from analysis.

Q. What's the difference between a BRD, an FRD and a User Story?

BRD: business intent + scope + success criteria, no how. FRD: functional behavior of the system, screen by screen, translated for developers. User Story: a small unit of value written from the user's perspective, with acceptance criteria, sized to fit inside a sprint. See Resource 11 for the comparison table.

Q. How do you write good acceptance criteria?

Use Given/When/Then. Each criterion is one testable behavior. Cover the happy path, at least two edge cases, and any non-functional constraints (latency, permissions, empty states). If a QA can't turn your criterion into a test, rewrite it.

Q. Estimate the number of ATMs in Bengaluru.

Show your reasoning aloud. Anchor on population (~13M), estimate banked adults (~60%), estimate ATM users per machine per day (~150), estimate transactions per user per week (~2), then solve. Sanity-check against known data. Interviewers score your structure and honesty about assumptions, not the number.

Q. A stakeholder asks for a feature you think won't help. What do you do?

Turn the ask into a hypothesis: what outcome do they expect, how will we measure it, what's the smallest test that would falsify it. Propose the smallest experiment rather than debating the feature. If they still insist, document the risk and let the decision stand on record.

Q. How do you prioritize a backlog?

Combine two lenses: value (RICE or a weighted score) and cost (T-shirt sizing with dev leads). Cut anything below a value/cost threshold. For everything above, sequence by dependency and risk-put learning-critical items first.

Q. Tell me about a time you failed.

Pick a real failure with a clear metric miss. Own it in one sentence. Spend the rest of the answer on what you changed in your process, and evidence that the change stuck. Interviewers use this to filter for self-awareness, not for perfection.

Q. How do you keep engineers unblocked?

Ship clarity daily. Every story leaves your desk with acceptance criteria, sample data, wireframes and edge cases. Hold a 15-minute daily office hour. When a question does surface, log the answer in the story so the next person doesn't ask again.

Q. A user story is too big for one sprint. What do you do?

Split by workflow step, by data type, or by user segment-never by layer (frontend/backend). Each slice must deliver observable value on its own. If you can't find a value-delivering slice, the story is a task, not a story.

Q. What tools do you use day-to-day?

Name your stack and one specific way you use each: JIRA for backlog + sprint reporting, Confluence for BRDs and decision logs, Figma for wireframes and stakeholder walkthroughs, Excel/Google Sheets for RTM and prioritisation matrices, SQL for self-serve data pulls. Call out any AI tools you rely on-see Resource 5.

Q. How do you measure your own success as a BA?

Two categories: outcome (did the shipped work move the intended KPI) and process (cycle time from idea to signed-off requirement, defects traced back to unclear requirements, stakeholder NPS). If you can't name a metric you own, the interviewer will assume you don't own outcomes.

Q. Why are you leaving your current company?

Frame forward, not away. What's true about your next chapter that isn't true today: scale of product, industry, ownership of a domain, learning slope. Never criticise a manager or company by name.

Q. Do you have any questions for us?

Ask three: (1) what does great look like in this role at the six-month mark, (2) what's the hardest problem the team is unsure how to solve, (3) how do product decisions actually get made here. These signal you think about outcomes, humility and ways of working.

RESOURCE 02

ATS-Friendly BA Resume Checklist

40+ checkpoints across formatting, content and keywords. Print it, tick as you go.

Format & ATS

- ☐ Single column layout, 10.5-11pt sans-serif (Inter, Calibri, Arial). No text boxes, no tables, no images - ATS parsers drop them.
- ☐ PDF, not .docx. File name: firstname-lastname-BA-resume.pdf.
- ☐ One page if under 8 years of experience. Two pages max.
- ☐ Consistent date format (Mmm YYYY - Mmm YYYY). Right-align dates.
- ☐ Section headings match ATS-friendly labels: Summary, Experience, Skills, Education, Certifications.

Header

- ☐ Name at 20-24pt bold. City + country (no full address). One phone, one email, one LinkedIn URL.
- ☐ Portfolio / GitHub / case study URL only if it's real and current.
- ☐ No photo, no date of birth, no marital status.

Summary (3 lines)

- ☐ Line 1: role + years + industry (e.g. 'Business Analyst with 4 years across fintech and lending').
- ☐ Line 2: 2-3 outcome-oriented capabilities.
- ☐ Line 3: what you're looking for next - makes recruiter routing effortless.

Experience bullets

- ☐ Every bullet starts with a strong verb (Led, Shipped, Reduced, Migrated, Automated). No 'Responsible for'.
- ☐ Every bullet has a metric where honest (% , hours saved, users, revenue, defect rate).
- ☐ 3-5 bullets per role. Most recent role gets the most bullets.
- ☐ Mix outcome bullets (what changed) with method bullets (how you worked).
- ☐ Include at least one bullet about stakeholder management for every senior-facing role.

Skills

- ☐ Group into: Analysis (SQL, Excel, RTM), Product (JIRA, Confluence, Figma), Delivery (Scrum, Kanban), Data (Power BI, Looker), AI (prompting, GPT integration).
- ☐ Only list a skill if you can defend it in a 5-minute conversation.

Keywords

- ☐ Copy the job description into a word cloud tool. Any noun that appears ≥ 3 times and matches your real experience should appear verbatim in your resume.
- ☐ Common BA keywords to check: requirement gathering, stakeholder management, BRD, FRD, user stories, backlog grooming, sprint planning, UAT, SQL, wireframing.

Proof

- ☐ Add a 'Selected Projects' or 'Case Studies' section with 2-3 one-line entries linking to Notion / PDF / portfolio.
- ☐ For freshers, replace 'Experience' with 'Projects' - lead with course capstones, live client projects, hackathons.

Final polish

- ☐ Read aloud once. Anything that sounds like corporate jargon - delete or rewrite.
- ☐ Run through Grammarly or a peer review. Zero typos.
- ☐ Ask 3 people to name your 'top 3 strengths' from the resume in 30 seconds. If they can't, revise.

RESOURCE 03

BRD Template

A 15-section skeleton for a Business Requirements Document. Fill each section in order; you'll produce a defensible BRD in half a day.

1. Document Control

Version, author, reviewers, sign-off date, revision history. Every BRD change increments the version.

2. Executive Summary

3-5 sentences a busy sponsor can read in 30 seconds: what problem, who it hurts, what we're doing, expected outcome, when.

3. Business Objectives

Bullet the outcomes the business is buying with this project. Each objective must be observable and time-bound. Example: 'Reduce onboarding drop-off from 42% to <30% by end of Q2.'

4. Success Criteria & KPIs

The specific numbers we'll measure at launch and at 90 days post-launch. Include baseline value, target value, measurement method and owner for each.

5. Scope

In-scope: capabilities, user groups, channels, data domains. Out-of-scope: everything a reasonable reader might assume is in scope but isn't. Being explicit here prevents 60% of change requests.

6. Stakeholders & RACI

Table with Responsible, Accountable, Consulted, Informed for each major workstream. Include stakeholder names - not just role titles.

7. Current State (As-Is)

Process map + list of pain points, tied to KPIs where possible. Screenshot the current experience. Include volume metrics (transactions/day, users, revenue).

8. Future State (To-Be)

Process map + narrative for each user journey. Callouts for what changes and why. Include a before-vs-after view.

9. Assumptions & Constraints

Assumptions: things we're taking as true without proof. Constraints: hard limits (budget, tech, regulatory, timeline). Every assumption should have a validation plan.

10. Risks & Mitigations

Table: risk description, probability (H/M/L), impact (H/M/L), owner, mitigation, trigger to escalate.

11. Dependencies

External teams, third-party vendors, upstream/downstream systems, regulatory approvals. Include the SLA each dependency owes us.

12. High-Level Requirements

Numbered list, business language, not technical. Each requirement traces to at least one objective. This section feeds the FRD.

13. Non-Functional Requirements

Performance targets, security posture, compliance obligations, availability SLA, accessibility (WCAG level), i18n, browser/device support.

14. Timeline & Milestones

Bar chart or milestone list with committed dates. Highlight critical path items.

15. Sign-Off

Explicit approval matrix. No development starts until every named approver has signed.

TIP

Version-control your BRD from day one. Every change should be commented so future you can reconstruct why a decision was made.

RESOURCE 04

User Story Template

Everything you need to write, split and sign off user stories that engineers can build without a follow-up meeting.

Format

As a [persona], I want to [capability], so that [outcome].

Example

As a first-time borrower, I want to save my loan application and resume it on any device, so that I don't lose my inputs when I need to fetch documents.

INVEST criteria

- * Independent - can ship without waiting for another story.
- * Negotiable - the details are a conversation, not a contract.
- * Valuable - delivers something a user or stakeholder can name.
- * Estimable - the team can size it with reasonable confidence.
- * Small - finishes inside one sprint, ideally in 2-4 days of dev time.
- * Testable - acceptance criteria are unambiguous and demoable.

Acceptance criteria - Given/When/Then

- * Given [starting state]
- * When [user action]
- * Then [observable outcome]

Worked example

- * GIVEN a borrower has entered valid personal details on step 1, WHEN they navigate away without submitting, THEN their inputs persist for 30 days.
- * GIVEN a saved application older than 30 days, WHEN the borrower returns, THEN the form starts fresh and a message explains why.
- * GIVEN a mobile-web session, WHEN the borrower resumes on desktop with the same account, THEN the last-saved state loads within 2 seconds.

Definition of Done - default

- * All acceptance criteria pass in staging.
- * Unit tests written; overall coverage doesn't drop.
- * UX approved via Figma walkthrough.
- * Analytics events wired and validated in the debug console.
- * Documentation updated (user-facing help + internal runbook).
- * PO / BA has signed off after a 5-minute demo.

RESOURCE 05

AI Prompts for Business Analysts

25 ready-to-use prompts for the tasks BAs do most: elicitation, drafting, prioritisation, communication and analysis. Copy, edit the [placeholders], paste.

01. Requirement Elicitation

You are an experienced business analyst preparing for a discovery workshop with [department] stakeholders about [problem area]. Draft 12 open-ended questions that surface (a) current process, (b) pain points and workarounds, (c) volumes and edge cases, (d) success metrics. Group by category.

02. BRD from meeting notes

Given these raw meeting notes: [paste notes]. Extract into a BRD skeleton with the following sections: Executive Summary, Objectives, In-Scope, Out-of-Scope, Assumptions, Open Questions. Flag any missing information with [NEEDS INPUT].

03. User story splitter

This user story is too large for one sprint: [paste story]. Suggest 3 ways to split it that each deliver independent user value. For each split, list the acceptance criteria and estimated relative size.

04. Acceptance criteria generator

For this user story: [paste story], write acceptance criteria using Given/When/Then format. Cover the happy path, 3 edge cases, and 2 non-functional constraints (performance, security or accessibility).

05. Stakeholder communication

Draft a 4-sentence status update for [senior stakeholder] about [initiative]. Include: what we shipped this week, one risk we're actively managing, what we need from them, and next milestone with date.

06. Meeting summary

Summarise this meeting transcript into: (1) 3 decisions made, (2) action items with owner and due date, (3) open questions for follow-up. Transcript: [paste].

07. Process mapping

Turn this narrative into a numbered process flow with actors, decision points, and system interactions: [paste narrative]. Format the output as a mermaid.js flowchart.

08. Competitor teardown

Compare how [Product A] and [Product B] handle [feature area]. Cover: user flow, information architecture, monetisation, and one differentiator each. Write it as a table.

09. Prioritisation matrix

Score these features on RICE (Reach, Impact, Confidence, Effort) using 1-10 for each dimension. Give a one-line rationale per score. Features: [paste list].

10. SQL question drafter

I'm interviewing a BA candidate. Write 5 SQL questions of increasing difficulty about an e-commerce schema (customers, orders, order_items, products, refunds). Include the ideal answer for each.

11. Interview prep coach

You are an interview coach. I'm applying for a BA role at [company] focusing on [domain]. Ask me 6 behavioural questions using the STAR framework, one at a time. After each answer, give feedback on structure, specificity and outcome clarity.

12. Test case generator

For this acceptance criterion: [paste], write 8 test cases covering happy path, boundary, negative, and integration scenarios. Format as a table with columns: Test ID, Precondition, Steps, Expected Result, Priority.

13. Risk register drafter

Given this project brief: [paste], draft a risk register with 10 rows. Columns: Risk, Probability (H/M/L), Impact (H/M/L), Category (technical/business/regulatory), Owner, Mitigation.

14. Change request assessor

A stakeholder has requested this change mid-sprint: [paste request]. Draft an assessment covering: scope impact, timeline impact, cost impact, risk of not doing it, and a recommendation with rationale.

15. Root cause analysis

A KPI dropped: [describe drop]. Run a 5-whys analysis. For each 'why' propose 2 hypotheses and the smallest experiment to test each.

16. Wireframe brief

Write a design brief for a designer to wireframe a [screen name] that lets a [persona] accomplish [task]. Include: primary action, secondary actions, information hierarchy, empty state, error state, and success state.

17. Persona builder

Build a user persona for [role/segment]. Include: goals (top 3), pain points (top 3), a day-in-the-life narrative, tools they use, and one quote in their own voice.

18. Release notes writer

Given this list of shipped features: [paste JIRA export], write release notes for (a) end-users in plain language, (b) internal go-to-market team in specific language.

19. A/B test design

Design an A/B test for [hypothesis]. Cover: metric, minimum detectable effect, sample size assumption, exclusion criteria, guardrail metrics, and a decision framework for shipping vs. iterating.

20. Data dictionary

For this database table: [paste DDL], write a data dictionary. Columns: field name, data type, description in business language, sample value, source of truth, PII flag.

21. Documentation refactor

This BRD is 25 pages and hard to follow: [paste]. Restructure it so a busy VP can grasp the intent in 3 minutes and a developer can find implementation-level detail in 30 seconds.

22. Stakeholder empathy

I'm about to present a plan to a stakeholder who is: [describe their pressures]. Predict their top 3 objections and draft a one-sentence response to each.

23. Discovery interview prep

I'm about to interview [role] to understand their [workflow]. Draft an interview guide with: (1) opener that builds rapport, (2) 8 open-ended questions in logical order, (3) 3 backup questions if they're brief, (4) closing that leaves door open for follow-up.

24. Feature deprecation memo

Draft a memo to internal teams explaining that [feature] will be deprecated on [date]. Cover: rationale, migration path for affected users, timeline of comms to customers, support cadence during the transition.

25. Post-mortem template

Write a blameless post-mortem template for a [type of incident]. Include: timeline of events, contributing factors (not root cause), what went well, what didn't, action items with owners and due dates, and a section on how we'll prevent recurrence.

RESOURCE 06

BA Career Roadmap 2026

A month-by-month plan from first learning to Principal BA. Adjust the timeline to your reality - keep the sequence.

Month 1-3

- * Learn the fundamentals: SDLC, Agile & Scrum, BRD vs FRD, user stories, acceptance criteria.
- * Pick one tool per category and get functional: JIRA (backlog), Confluence (docs), Figma (wireframes), Excel/Sheets (RTM & prioritisation).
- * Practise SQL to intermediate (joins, aggregations, window functions). Aim for solving 50 problems on any platform.
- * Build a mock BRD + user story set for a public product you use every day. Post it publicly on LinkedIn.

Month 4-6

- * Land your first BA role - apply broadly, don't over-optimize for domain fit.
- * In the first 30 days: shadow 5 stakeholders, document one process end-to-end, write your first user stories with acceptance criteria.
- * Attend every ceremony (standup, planning, retro, refinement). Take notes and share back.
- * Build your first case study: one initiative you owned, KPI moved, artefacts.

Month 7-12

- * Own a feature end-to-end: discovery -> requirements -> launch -> measurement.
- * Learn one adjacent skill deeply - pick from: product analytics, UX research, or basic Python/pandas.
- * Start a portfolio site with 2-3 case studies (redact confidential specifics).
- * Speak up in retros. Propose one process improvement per quarter.

Month 13-24

- * Move into a Senior BA or Product Owner role - the biggest salary and scope jump happens here.
- * Lead a workstream with 2+ engineering squads. Own OKRs and quarterly business reviews.
- * Mentor one junior BA. Teaching sharpens your fundamentals.
- * Pick a specialisation: payments, lending, insurance, e-commerce, healthtech, edtech. Depth beats breadth after year 2.

Month 25-36

- * Move to Principal BA or Group Product Owner. Own domain strategy, not just delivery.
- * Lead cross-org initiatives. Practise saying no to well-liked but low-value work.
- * Build a public reputation - write, speak, or open-source. This is the biggest lever for recruiter inbound.
- * Prepare for the fork: management (people + process) or IC/expert (deep specialism). Both are valid - pick with intent.

RESOURCE 07

Market Research Framework

Six steps from question to decision. Skip a step and you'll produce a deck no one uses.

- 01** 1. Frame the question. Write the decision the research will inform in one sentence. If you can't name a decision, the research isn't worth doing.
- 02** 2. Choose the method to the question. Quant answers 'how many'; qual answers 'why'. Desk research answers 'what's public'; primary research answers 'what's specific to us'. Mix at least two.
- 03** 3. Recruit or pull the sample. For interviews, 5-8 users per persona is enough to spot patterns. For surveys, aim for ≥ 100 responses per segment. Never mix employees with real users.
- 04** 4. Instrument and run. Interviews: semi-structured guide, one moderator, one note-taker, recording with consent. Surveys: pilot with 5 people first, keep under 10 minutes, mix Likert with one open text.
- 05** 5. Synthesize. Affinity-map raw quotes into themes. Weight themes by frequency and by impact (a rare-but-critical pain outranks a common-but-mild one). Deliver 3-5 insights, not 50 quotes.
- 06** 6. Convert to action. Every insight ends with a proposed decision or hypothesis to test. If it doesn't, it's trivia.

RESOURCE 08

Wireframing & Mockup Guide

How to communicate product intent visually - without pretending to be a designer.

Principles

- * Wireframes solve information architecture, not visual design. Stay grayscale until the flow is agreed.
- * One primary action per screen, obvious in <5 seconds.
- * Show the empty state, error state and success state - not just the happy path.
- * Content shapes the layout, not the reverse. Use real (or realistic) copy.
- * Annotate: label every interactive element with what happens on click and what data it needs.

Fidelity levels

- * Sketch (paper / whiteboard) - use in the first 24 hours of a problem to align on structure.
- * Low-fi (Figma / Whimsical, grayscale) - use for stakeholder walkthroughs and dev handoff.
- * Hi-fi (Figma with real styles) - designer owns, not the BA. Involve them once the low-fi is agreed.

Tools by budget

- * Free: Figma (community), Whimsical, Excalidraw, Balsamiq (cloud trial).
- * Paid: Figma Pro, Sketch, Axure (for complex interactive prototypes).
- * For non-visual BAs: start with Whimsical - the fastest for boxes-and-arrows thinking.

5-step walkthrough

- * 1. Map the user's task as a flowchart before opening the wireframe tool.
- * 2. Draft each screen in the flow with only headers, primary action and content blocks.
- * 3. Add empty, loading, error and success states for every screen.
- * 4. Add annotations for interactions, data requirements, edge cases.
- * 5. Walk 3 real users through the flow. Rewrite based on where they hesitated.

RESOURCE 09

Scrum & Agile Cheat Sheet

One printable reference for roles, ceremonies, artefacts and the traps to avoid.

Roles

Product Owner	Owns the what and the why. Prioritises the backlog. Says no more than yes. Single voice of the customer.
Scrum Master	Owns the how. Removes blockers. Coaches the team on process. Protects the sprint from mid-sprint scope changes.
Developers	Own the how at the delivery level. Estimate, build, test, ship. Self-organizing within the sprint.
BA (embedded)	Optional role in Scrum but common in enterprises. Owns requirements clarity, stakeholder liaison, and cross-team dependencies.

Events

Sprint Planning	Start of sprint. Team picks stories from the top of the backlog that they can commit to. Duration: 2 hours per week of sprint.
Daily Standup	15 minutes, same time daily. Focus on impediments and coordination, not status reports.
Backlog Refinement	1-2 hours mid-sprint. PO/BA walks the team through upcoming stories; team estimates and asks clarifying questions.
Sprint Review	End of sprint. Demo of shipped increments to stakeholders. Gather feedback for the backlog.
Retrospective	End of sprint after review. Team-only. Identify 1-2 process changes to try next sprint. Actually try them.

Artefacts

Product Backlog	Prioritised list of everything that might be built. Only the top is refined; everything below is a placeholder.
Sprint Backlog	The subset picked for this sprint. Frozen at the start; changes require negotiation.
Increment	The working software at the end of the sprint. Must meet the Definition of Done.

Common anti-patterns

- * Standups turn into status reports for the manager. Fix: standup is for the team, not the manager.
- * Sprint reviews become internal-only. Fix: invite one real stakeholder every sprint.
- * Retros become gripe sessions. Fix: end every retro with 1-2 changes and revisit them next sprint.
- * Stories carry over sprint after sprint. Fix: enforce a max story size; split anything larger.
- * BA writes stories in isolation and hands them to the team. Fix: 3-amigos (BA, dev, QA) refine every story together.

RESOURCE 10

Requirement Gathering Framework

Eight techniques and when to reach for each. Never rely on interviews alone.

Interviews

1:1 with a stakeholder, 45-60 min, semi-structured. Best for depth on complex processes.

Workshops

Group session, 2-4 hours, facilitated. Best for alignment across stakeholders and rapid trade-off decisions.

Observation

Silent shadowing of a user doing their real work. Best for surfacing gaps between what people say and what they do.

Document analysis

Existing SOPs, spreadsheets, tickets, past BRDs. Best for the as-is baseline before you ever meet a stakeholder.

Surveys

Broad reach, low-depth. Best for volume-based validation of hypotheses formed in interviews.

Prototyping

Show a low-fi mock and let the stakeholder react. Best when stakeholders struggle to articulate needs abstractly.

Data mining

Query the source systems yourself. Best for understanding volumes and patterns without asking anyone.

Reverse engineering

For legacy systems where docs are lost - walk through the working system with an engineer, capture behavior.

RESOURCE 11

BRD vs FRD vs User Stories

Which document, for which audience, at which point of a project. Print and pin.

Dimension	BRD	FRD	User Stories
Audience	Business sponsors, PMO, senior stakeholders	Development team, technical architects	QA, Development team, PO, BA
Written by	Business Analyst, in business language	Business Analyst / Systems Analyst, in system language	PO or BA, from the user's perspective
Level of detail	High-level intent, scope, KPIs	Field-by-field, screen-by-screen system behavior	Small, testable slice with acceptance criteria
Signed off by	Business sponsor, project steering committee	Tech lead, QA lead, BA	PO acceptance in the sprint review
Lifecycle	One per initiative; heavy revision control	One per module / feature; versioned	Living - refined every sprint; retired at close
When to use	Initiation - before build starts	Between BRD and dev handoff for complex systems	Every sprint of an Agile team
Length	20-50 pages typical	50-200 pages depending on scope	Half a page each
Fails when	Written after dev started; used as a shield instead of a shared plan	Diverges from the built system and nobody updates it	Skipped 'because we're Agile'; team ends up building on assumption

YOUR NEXT STEP

Templates take you to the interview.

A structured practice takes you into the job.

If this kit was useful, the Practical Business Analyst Course is the natural next step - a full curriculum with real project docs, live mentoring, and a portfolio-ready capstone.

pmcodex.com/business-analyst-course

Explore the course, syllabus and student outcomes.

t.me/boost/Officialpmcodex

Join our Telegram community for weekly BA prompts, jobs and Q&A.

