

MASTERING CLAUDE

for **BUSINESS ANALYSTS**

RITESH ARORA



BlueRoseONE^{.com}
S t o r i e s M a t t e r

New Delhi • London

BLUEROSE PUBLISHERS

India | U.K.

Copyright © Ritesh Arora 2026

All rights reserved by author. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the author. Although every precaution has been taken to verify the accuracy of the information contained herein, the publisher assumes no responsibility for any errors or omissions. No liability is assumed for damages that may result from the use of information contained within.

BlueRose Publishers takes no responsibility for any damages, losses, or liabilities that may arise from the use or misuse of the information, products, or services provided in this publication.



For permissions requests or inquiries regarding this publication,
please contact:

BLUEROSE PUBLISHERS

www.BlueRoseONE.com

info@bluerosepublishers.com

+91 8882 898 898

+4407342408967

ISBN: 978-93-7825-176-4

Cover design: Sharvi

Typesetting: Simran Maurya

First Edition: May 2026

Acknowledgements

To my heroes in life

This book is dedicated to both my parents (Late Shri Om Prakash Arora and Late Smt Saroj Arora) who would have been proud to see this day

To my wife Monica Arora, who has been with me, in thick and thins of life

To my sister Jasmin Sehgal, who has been my sounding board in my life

To my genius computer wizard teen son, Aayush Arora, whose tech prowess amazes me everyday

To Deepa Soni, my sister, who has been a continuous source of encouragement and inspiration to me

To Vikrant Kapoor, my brother-in-law, who has been partner in crime for endless weekend tech discussion, for ages

To Rajkumar Udani, my mentor and brother, Global head of Enterprise Integration and process orchestration, Cognizant technologies.

To Dr. Surya Uday Singh, a brother and a friend, who saved the lives of countless friends of mine in Covid times

To Vijay Kumar, a former colleague in Coforge, who is now more of family.

About the Author

Ritesh Arora is a seasoned IT leader focusing on enterprise business process automation and AI, currently working for a Big Four Consulting firm. He is an author of few Amazon best sellers' fiction stories as well. He is an IIM Calcutta Alumnus and B Tech from Productiin engineering (Rank 1) from Punjab Technical University. He has a big library at home, and is ready for a game of table tennis any time.



2026 EDITION

Mastering Claude for Business Analysts

A comprehensive, hands-on book for mid-senior Business Analysts to integrate Claude AI into daily workflows —
from requirements elicitation to stakeholder
communication, data analysis, and process design.

Content

Module 1: Foundations – How Claude Thinks.....	1
1.1 What Claude Can and Can't Do for a BA	1
1.2 Conversation vs. Completion Mental Model	3
1.3 Prompt Anatomy for BA Work	5
1.4 Responsible Use and Data Governance	8
Module 2: Requirements Elicitation with Claude	11
2.1 Stakeholder Interview Preparation	11
2.2 Extracting Requirements from Transcripts.....	13
2.3 Ambiguity Detection and Gap Analysis.....	15
2.4 Generating User Stories and Acceptance Criteria	17
Module 3: Documentation & Artifacts.....	20
3.1 BRD Generation from Requirements	20
3.2 Process Documentation and BPMN Narratives	22
3.3 Decision Logs and Trade-off Matrices	24
3.4 Editing, Tone-Shifting, and Audience Adaptation ..	27
Module 4: Data Analysis & Insight Generation	30
4.1 Exploratory Data Analysis with Claude	30
4.2 SQL and Formula Generation.....	32
4.3 Interpreting and Narrating Data	34
4.4 Validating Claude's Analysis	36
Module 5: Stakeholder Communication.....	39
5.1 Executive Status Updates	39
5.2 Difficult Conversations and Pushback Emails	41
5.3 Presentation Outline and Talking Points.....	43
5.4 Cross-Functional Translation	46

Module 6: Process Improvement & As-Is/To-Be.....	49
6.1 As-Is Process Extraction.....	49
6.2 Bottleneck and Waste Identification.....	51
6.3 To-Be Process Design	54
6.4 Impact Assessment and Change Readiness.....	56
Module 7: Prompt Engineering Patterns for BAs.....	60
7.1 Chain-of-Thought for Complex Analysis	60
7.2 Few-Shot Examples and Templates.....	62
7.3 Structured Output: JSON, Tables, XML.....	64
7.4 Iterative Refinement and Critique Loops.....	66
Module 8: Capstone – End-to-End BA Workflow	70
8.1 Project Kickoff: Context Gathering and Scoping	70
8.2 Requirements and Documentation Sprint	72
8.3 Stakeholder Review Simulation.....	75
8.4 Retrospective: Measuring Your AI-Augmented Efficiency	77

Module 1

Foundations — How Claude Thinks

Understand Claude's capabilities, limitations, and reasoning to set realistic expectations.

1.1 What Claude Can and Can't Do for a BA

Claude is a large language model that excels at processing, generating, and transforming text. For a Business Analyst, this translates into a powerful set of capabilities — but only if you understand the boundaries.

What Claude does well for BAs: Claude can summarise long documents in seconds, extract structured data from unstructured text (meeting notes, emails, transcripts), generate first drafts of BRDs, PRDs, user stories, and status reports, write SQL queries and Excel formulas from plain English descriptions, rewrite content for different audiences (exec vs. technical vs. end-user), and identify gaps, contradictions, and ambiguities in requirements.

What Claude cannot do: Claude has no access to your company's live systems, databases, or real-time data unless you provide it. It cannot browse the internet during a conversation (unless web search is enabled). It doesn't remember previous conversations by default. It can "hallucinate" — generating plausible-sounding but incorrect facts, especially with numbers and statistics. It

cannot make decisions for you; it's a thinking partner, not a decision-maker.

The mental model shift: Think of Claude not as a search engine or a database, but as an extremely well-read junior analyst who works at superhuman speed but needs clear direction, verification, and judgement from you.

KEY TAKEAWAYS

- Claude excels at text transformation: summarisation, extraction, generation, and reformatting
- Claude has no access to live systems — you are the data pipeline
- Always verify Claude's outputs, especially numbers, statistics, and named entities
- Think of Claude as a fast junior analyst who needs your direction and judgement

PROMPT TEMPLATE: CAPABILITY ASSESSMENT

You are helping a senior Business Analyst assess which of their tasks can be augmented by an AI assistant.

Here is a list of my typical weekly BA tasks: [PASTE YOUR TASK LIST HERE]

For each task, classify it as:

- AI-AUTOMATED: Claude can do this end-to-end with minimal review
- AI-AUGMENTED: Claude can draft/accelerate but I need to review and refine
- HUMAN-ONLY: Requires live system access, relationship judgement, or organisational context Claude can't have

Format as a table with columns: Task | Classification | Rationale | Time Saved Estimate

SPEAKER NOTES

Open with a live demo: paste a messy meeting note into Claude and extract action items in real time.

Spend 5 minutes on capabilities — participants will be eager to see them.

Spend equal time on limitations — managing expectations early prevents frustration.

The hallucination point is critical: show a real example where Claude generates a plausible but wrong statistic.

Ask participants: “What’s one task you wish you could hand off?”

1.2 Conversation vs. Completion Mental Model

To use Claude effectively, you need to understand how it actually processes your requests. This isn’t about the technical internals — it’s about the practical mental model that helps you write better prompts.

The context window: Every time you send Claude a message, it reads the ENTIRE conversation from the beginning. It doesn’t have memory between separate conversations. Think of each new conversation as meeting a brilliant colleague for the first time — they know nothing about your project unless you tell them.

Why the same prompt gives different results: Claude’s responses have a degree of randomness (controlled by a parameter called “temperature”). This means asking the exact same question twice might give you slightly different phrasing or structure. This is a feature, not a bug — it helps with creative tasks. But for tasks where you need consistency (like filling a template), you’ll want to be very specific about format.

Tokens and limits: Claude thinks in “tokens” — roughly 3/4 of a word. The context window has a limit. For Claude Sonnet, this is approximately 200,000 tokens — about 150,000 words or ~300 pages. This is enormous, but if you’re working with very large documents, you’ll need to chunk them.

Practical implication: Front-load context. Give Claude the project background, your role, the audience, and any constraints BEFORE asking for output. The more context you provide upfront, the better the output quality.

KEY TAKEAWAYS

- Each conversation starts fresh — Claude has no memory between sessions
- The full conversation is re-read every time, so front-loading context matters
- Slight variation between runs is normal; constrain format for consistency
- ~200K token context = ~300 pages; chunk larger documents

PROMPT TEMPLATE: CONTEXT-LOADING TEMPLATE

I’m a Senior Business Analyst working on [PROJECT NAME] for [COMPANY/DEPARTMENT].

Project context:

- Business objective: [WHAT THE PROJECT AIMS TO ACHIEVE]
- Key stakeholders: [LIST ROLES, NOT NAMES]
- Current phase: [DISCOVERY/ANALYSIS/DESIGN/UAT/ etc.]

- Key constraints: [TIMELINE, BUDGET, TECH STACK, REGULATORY]
- My specific need right now: [DESCRIBE WHAT YOU NEED CLAUDE TO HELP WITH]

Output requirements:

- Format: [TABLE / DOCUMENT / LIST / NARRATIVE]
- Audience: [WHO WILL READ THIS]
- Tone: [FORMAL / CONVERSATIONAL / EXECUTIVE]
- Length: [APPROXIMATE WORD COUNT OR PAGE COUNT]

SPEAKER NOTES

This lesson is about building the right mental model. Many people treat Claude like Google — short queries expecting perfect answers. The key shift is: Claude rewards context, not brevity.

Demo: Show the same question with minimal context vs. rich context. Compare outputs side by side.

Emphasise the “fresh conversation” point — this is the #1 surprise for new users.

1.3 Prompt Anatomy for BA Work

A well-structured prompt is the single biggest lever for getting great output from Claude. Think of it as a brief — the more precise your brief, the better the deliverable.

The 5-part prompt structure for BA work:

1. **Role:** Tell Claude who it is. “You are a senior business analyst with 10 years of experience in financial services” gives Claude a lens through which to filter its response. This is surprisingly effective at improving relevance and depth.

2. **Context:** Background information Claude needs. Project details, business domain, stakeholders, constraints, what's been decided so far. This is the most commonly skipped part— and the most impactful when included.
3. **Instruction:** The specific task. Be direct: “Write a BRD”, “Extract requirements from this transcript”, “Identify gaps in this user story set”. Verbs matter — “analyse” gives a different output than “list” or “compare”.
4. **Constraints:** Boundaries on the output. Word count, format, things to include or exclude, audience level, tone.
5. **Output format:** Be explicit about structure. “Return as a numbered list”, “Use a table with columns X, Y, Z”, “Follow this template: [paste template]”. Claude follows formatting instructions remarkably well.

The power move: When you combine all five parts, Claude's output quality jumps dramatically. Most people only use part 3 (the instruction). Adding the other four parts is what separates effective AI-augmented BAs from casual users.

KEY TAKEAWAYS

- The 5 parts: Role → Context → Instruction → Constraints → Output Format
- Role-setting dramatically improves response relevance and depth
- Context is the most skipped part but has the highest impact
- Explicit output format instructions give you consistent, reusable results
- Action verbs matter: ‘analyse’ ≠ ‘list’ ≠ ‘compare’ ≠ ‘recommend’

PROMPT TEMPLATE: 5-PART BA PROMPT TEMPLATE

[ROLE] You are a senior business analyst with deep experience in [DOMAIN]. You are meticulous about requirements traceability and stakeholder alignment.

[CONTEXT] Project: [NAME] Objective: [WHAT AND WHY]
Current phase: [PHASE] Key stakeholders: [ROLES]
Constraints: [LIST] What's been decided: [DECISIONS SO FAR]

[INSTRUCTION] [YOUR SPECIFIC REQUEST — use a strong action verb]

[CONSTRAINTS] - Do not include: [EXCLUSIONS] - Audience: [WHO READS THIS] - Tone: [FORMAL/CONVERSATIONAL]
- Length: [TARGET]

[OUTPUT FORMAT] Return as: [TABLE / NUMBERED LIST / DOCUMENT with headings / TEMPLATE / JSON] Structure: [DESCRIBE THE EXACT STRUCTURE YOU WANT]

SPEAKER NOTES

This is the most important lesson in the entire course. Everything that follows builds on this structure.

Walk through each of the 5 parts with a live example — build a prompt together with the class.

Show a before/after: same request with just the instruction vs. full 5-part prompt.

Assign homework: rewrite 3 of your recent Claude prompts using the 5-part structure.

1.4 Responsible Use and Data Governance

Before you start pasting company data into Claude, you need a clear understanding of what's safe, what's risky, and what's absolutely off-limits.

What NOT to paste into Claude: Never paste PII – customer names, email addresses, phone numbers, social security numbers, account numbers. Never paste proprietary source code, trade secrets, or confidential financial data unless your organisation has an enterprise agreement with Anthropic. Never paste credentials, API keys, passwords, or internal system URLs.

What IS generally safe: Anonymised or redacted data (replace names with “Stakeholder A”, “Customer #1234”), generic process descriptions, publicly available information, your own analysis and writing that doesn't contain sensitive data, template structures and frameworks.

Enterprise vs. consumer Claude: If your organisation uses Claude through the API or an enterprise plan, your data is typically not used for training. If you're using the free or Pro consumer version, check Anthropic's current data usage policies. When in doubt, anonymise.

The BA's responsibility: As a BA, you're often the person with the broadest access to business context. You see financial data, customer data, strategy documents, and internal communications. This makes data governance especially important for BA roles using AI tools. Build a habit: before pasting anything, ask “would I be comfortable if this appeared in a public search result?”

KEY TAKEAWAYS

- Never paste PII, credentials, trade secrets, or confidential financials
- Anonymise data before sharing — use role labels and generic identifiers
- Enterprise API/plan ≠ consumer plan for data handling — know which you're on
- The litmus test: would you be comfortable if this appeared publicly?
- BAs have especially broad data access — governance responsibility is higher

PROMPT TEMPLATE: DATA ANONYMISATION

I need to share a business scenario with an AI assistant for analysis. Below is the raw text. Please help me anonymise it by:

1. Replacing all person names with role-based labels (e.g., “Project Sponsor”, “Lead Developer”)
2. Replacing company names with generic labels (e.g., “Client Corp”, “Vendor Inc”)
3. Replacing specific financial figures with representative ranges
4. Removing any email addresses, phone numbers, or account IDs
5. Flagging anything else that might be sensitive

Here is the text to anonymise: [PASTE YOUR TEXT HERE]

Return the anonymised version and a separate list of what was changed.

SPEAKER NOTES

This lesson sets the ethical foundation. Don't skip it — governance concerns are the #1 reason enterprise adoption stalls.

The anonymisation prompt is genuinely useful — demo it live.

Close with: "This isn't about fear — it's about building trust so your organisation expands AI access rather than restricting it."

MODULE EXERCISE: BA TASK AUDIT

1. List your 10 most common weekly BA tasks
2. Use the Capability Assessment prompt to classify each task with Claude's help
3. For each AI-AUGMENTED task, describe how Claude helps and what you'd still do manually
4. Estimate total weekly hours saved if all AI-augmented tasks were optimised
5. Present your audit to the group with your rationale

Deliverable: A one-page task classification table with time-saved estimates and a 2-minute presentation.

Module 2

Requirements Elicitation with Claude

Use Claude as a thinking partner for interviews, requirement extraction, and gap analysis.

2.1 Stakeholder Interview Preparation

The quality of a requirements gathering session depends almost entirely on preparation. Claude can dramatically improve your interview prep by generating targeted, context-aware questions you might not think of.

How to use Claude for interview prep: Feed Claude the project brief, the stakeholder's role, and what you already know. Ask it to generate questions organised by theme – functional needs, pain points, constraints, success criteria, dependencies, and risks.

Going deeper – the “what might I miss” technique: After generating your initial question set, ask Claude: “What blind spots might I have? What questions would a more experienced BA in this domain ask that I haven't covered?” This meta-prompting technique often surfaces genuinely valuable questions around edge cases, organisational politics, and compliance requirements.

Practical tip: Don't use Claude's questions as-is. Review them, rewrite them in your voice, and reorder them to create

a natural conversation flow. Claude gives you the raw material; you craft the interview experience.

KEY TAKEAWAYS

- Feed Claude the project brief + stakeholder role for targeted question generation
- Organise questions by theme: functional, pain points, constraints, success criteria, risks
- Use the 'what might I miss' meta-prompt to surface blind spots
- Always rewrite generated questions in your voice and reorder for conversational flow

PROMPT TEMPLATE: INTERVIEW PREP GENERATOR

You are a senior BA preparing for a stakeholder interview.

Context:

- Project: [PROJECT NAME AND BRIEF DESCRIPTION]
- Stakeholder role: [e.g., VP of Operations, Head of Customer Service]
- Their likely concerns: [WHAT YOU ALREADY KNOW]
- What's been decided so far: [KEY DECISIONS]

Generate a structured interview guide with:

1. Opening questions (rapport + context confirmation) – 3 questions
2. Core functional requirements questions – 5-7 questions
3. Pain points and current challenges – 3-4 questions
4. Constraints and dependencies – 3-4 questions
5. Success criteria and KPIs – 2-3 questions
6. Risk and concern questions – 2-3 questions
7. Closing / "anything else" questions – 2 questions

For each question, add a brief note on WHY you're asking it.

Then add: "Blind spot check – questions you might not think to ask but should" with 3-5 additional questions.

SPEAKER NOTES

This is where participants start seeing real productivity gains.

Demo: Paste a real project brief and stakeholder role. Generate questions live.

The “blind spot” technique is the highlight – demo this separately.

2.2 Extracting Requirements from Transcripts

Meeting transcripts and notes are gold mines of requirements – buried under conversation, tangents, and informal language. Claude excels at mining this gold.

The extraction workflow: Step 1: Paste the transcript into Claude with a clear instruction to extract requirements. Step 2: Categorise each as functional, non-functional, or constraint. Step 3: Flag ambiguous items. Step 4: Suggest follow-up questions for each ambiguous item.

Handling messy notes: Real meeting notes are rarely clean. They have half-sentences, contradictions, and implicit requirements. Tell Claude explicitly: “This transcript is rough and informal. Extract requirements even when they’re stated implicitly or informally.”

The traceability angle: Ask Claude to include a reference to where in the transcript each requirement came from. This gives you traceability from day one.

KEY TAKEAWAYS

- Workflow: Paste transcript → Extract requirements → Categorise → Flag ambiguities → Generate follow-ups
- Tell Claude the notes are informal – it adjusts its extraction sensitivity
- Ask for implicit requirements explicitly
- Request source references for traceability back to stakeholder statements

PROMPT TEMPLATE: REQUIREMENTS EXTRACTION FROM TRANSCRIPT

You are a detail-oriented Business Analyst extracting requirements from a stakeholder meeting.

Here is the transcript / meeting notes:

[PASTE TRANSCRIPT OR MEETING NOTES]

Extract every requirement – stated explicitly OR implied implicitly.

For each: | # | Requirement | Type (Functional / Non-Functional / Constraint) | Source (quote or paraphrase) | Confidence (High/Medium/Low) | Ambiguity flag (Y/N) |

After the table:

1. List any CONTRADICTIONS between requirements
2. List any IMPLICIT requirements hinted but not directly stated
3. For each ambiguous item, suggest a specific follow-up question
4. Note any MISSING areas – topics you'd expect to be covered but weren't

SPEAKER NOTES

This lesson has the highest “wow factor” for experienced BAs.

Use a realistic transcript with contradictions and implicit requirements.

The “implicit requirements” extraction is often the most valuable part.

2.3 Ambiguity Detection and Gap Analysis

One of the highest-value BA skills is spotting what’s missing or unclear in a requirements set. Claude can serve as an always-available requirements reviewer.

Ambiguity detection: Paste your requirements and ask Claude to identify: vague language, undefined terms, missing boundary conditions, and implicit assumptions.

Gap analysis patterns: Compare requirements against common categories (security, performance, accessibility, audit, error handling, data migration, rollback, reporting), similar systems, and stakeholder roles not yet interviewed.

The “adversarial reviewer” technique: Ask Claude to role-play as a critical QA lead, a skeptical architect, or a demanding product owner. Each persona flags different types of gaps. This multi-perspective review normally requires multiple people and meetings.

KEY TAKEAWAYS

- Vague language, undefined terms, missing boundaries, and implicit assumptions are the four ambiguity types
- Compare requirements against standard NFR categories to find gaps systematically
- Use persona-based review: QA lead finds edge cases, architect finds integration gaps, PO finds scope gaps
- The adversarial reviewer technique replaces what would normally be a multi-person review meeting

PROMPT TEMPLATE: REQUIREMENTS GAP ANALYSIS

You are a critical requirements reviewer. Your job is to find every gap, ambiguity, and missing requirement.

Here are the current requirements:

[PASTE REQUIREMENTS]

Perform:

1. AMBIGUITY AUDIT

For each vague/undefined term: problematic phrase, why it's ambiguous, suggested clarification question.

2. GAP ANALYSIS

Check for missing requirements: Security & Authentication, Performance & Scalability, Error handling & Edge cases, Data migration, Audit & Compliance, Accessibility, Reporting & Analytics, Rollback / Disaster recovery.

3. ADVERSARIAL REVIEW

Review as: QA Lead (what's untestable?), Solutions Architect (what technical risks?), End User (what workflow scenarios missing?).

4. RECOMMENDATIONS

Prioritise the top 5 gaps that would cause the most project risk.

SPEAKER NOTES

This lesson demonstrates Claude as a genuine thinking partner. The adversarial reviewer technique always generates discussion.

Emphasise that Claude catches things missed in peer reviews because it's systematic and tireless.

2.4 Generating User Stories and Acceptance Criteria

Transforming raw requirements into well-structured user stories with given/when/then acceptance criteria is bread-and-butter BA work – and Claude does it remarkably well.

The generation workflow: Start with raw requirements. Generate user stories in standard format. For each, generate acceptance criteria in given/when/then. Estimate complexity (S/M/L) and flag dependencies.

Quality control points: Watch for: oversized stories (epics disguised as stories), missing negative paths, missing edge cases, acceptance criteria that are implementation details, and stories without clear business value.

The refinement loop: After generating the initial set, paste them back and ask: “Review these user stories. Split any that are too large, add negative/edge case scenarios, and ensure every acceptance criterion is testable.” This second pass typically improves quality significantly.

KEY TAKEAWAYS

- Claude generates user stories + acceptance criteria from raw requirements efficiently
- Always review for: oversized stories, missing negative paths, implementation-level ACs, weak 'so that' clauses
- The refinement loop (generate → review → improve) consistently improves output quality
- Ask Claude to flag dependencies between stories

PROMPT TEMPLATE: USER STORY GENERATOR

You are a senior BA writing user stories for sprint planning.

Raw requirements:

[PASTE RAW REQUIREMENTS]

For each requirement, generate:

User Story

****As a**** [specific user role] ****I want**** [capability] ****So that**** [measurable business value]

Acceptance Criteria (Given/When/Then)

- Given [precondition], When [action], Then [expected outcome]
- Include at least one happy path AND one error/edge case per story

Story Metadata

- ****Size estimate:**** S / M / L
- ****Dependencies:**** [List prerequisite stories]
- ****Assumptions:**** [What we're assuming]

After all stories, add: 1. Dependency map 2. Stories that should be split further 3. Requirements that don't fit into user stories (technical spikes, NFRs)

SPEAKER NOTES

This lesson has a clear before/after: show manually written stories vs. Claude-generated ones.

Demo the refinement loop live: generate, then paste back for self-review.

Discuss when stories need human judgement: priority, sequencing, political considerations.

MODULE EXERCISE: FULL REQUIREMENTS ELICITATION CYCLE

1. Start with the provided case study project brief
2. Use Interview Prep Generator for 2 different stakeholder roles
3. Feed sample transcript into Claude to extract requirements
4. Run Gap Analysis on extracted requirements
5. Generate user stories with acceptance criteria for top 10 requirements
6. Review and refine using the refinement loop

Deliverable: Complete requirements package: interview guide, extracted requirements table, gap analysis report, and refined user stories with ACs.

Module 3

Documentation & Artifacts

Produce BRDs, PRDs, process flows, and decision logs at 3× speed.

3.1 BRD Generation from Requirements

A Business Requirements Document is one of the most time-consuming BA deliverables. Claude can produce a structured first draft in minutes — but the key is providing a clear template and rich context.

The approach: Don't ask Claude to "write a BRD." Instead, provide your organisation's BRD template, extracted requirements, project context, and decisions already made. Claude populates the template with your content.

What Claude does well in BRDs: Executive summary writing, scope statements, requirements tables with traceability, assumptions/constraints sections, and glossary generation.

What still needs human touch: Stakeholder sign-off sections, specific system references, organisational terminology, political framing, and version history.

The iterative approach: Generate section by section rather than all at once. This gives better control and lets you feed output from one section as context for the next.

KEY TAKEAWAYS

- Provide your org's BRD template + extracted requirements for best results
- Generate section by section for better control
- Claude excels at narrative sections linking requirements to business value
- Human touch needed for: sign-offs, system specifics, political framing

PROMPT TEMPLATE: BRD GENERATOR

You are a senior BA drafting a Business Requirements Document.

Project context:

- Project name: [NAME]
- Business objective: [OBJECTIVE]
- Target users: [USER GROUPS]
- Scope: [IN-SCOPE AND OUT-OF-SCOPE]

Requirements:

[PASTE REQUIREMENTS]

Generate a BRD with sections:

1. Executive Summary
2. Business Objectives
3. Scope (in/out/future)
4. Stakeholders table
5. Functional Requirements (grouped, with ID, description, priority, source)
6. Non-Functional Requirements
7. Assumptions

8. Constraints
9. Dependencies
10. Risks table
11. Glossary
12. Appendix: traceability matrix

SPEAKER NOTES

This is where the “3× speed” claim becomes tangible.

Demo section-by-section generation live. Start with the executive summary, then feed it into the next prompt for the scope section.

Key point: Claude gives you 70-80%, you refine the last 20-30%. That refinement is where your BA expertise shines.

3.2 Process Documentation and BPMN Narratives

Documenting processes is often tedious but critical. Claude can transform verbal descriptions into structured process documentation with actors, decision points, exception paths, and handoffs.

From conversation to structure: Describe a process as you would explain it to a colleague. Claude structures it into numbered steps, decision diamonds, exception paths, and swimlane-style actor assignments.

Exception path coverage: Ask Claude: “For each step, what could go wrong? What’s the exception path?” Claude considers timeouts, validation failures, access denials, data quality issues, and system unavailability.

Mermaid diagram generation: Claude can generate Mermaid syntax that renders as flowcharts directly in Claude’s interface. This means you can go from a verbal

description to a visual process map in a single prompt. For formal BPMN, use the Mermaid output as a blueprint and recreate in your standard tool (Visio, Lucidchart, draw.io).

The swimlane technique: When describing cross-functional processes, ask Claude to organise by actor. This immediately surfaces handoff points — where most process failures occur.

KEY TAKEAWAYS

- Describe processes conversationally — Claude adds formal structure
- Claude can generate Mermaid syntax for automatic flowchart rendering
- Exception path generation is high-value — Claude is systematic about failure modes
- Swimlane organisation surfaces handoff points where most processes fail

PROMPT TEMPLATE: PROCESS DOCUMENTATION GENERATOR

You are a senior BA documenting a business process.

Process name: [PROCESS NAME]

Process owner: [ROLE]

Description: [DESCRIBE THE PROCESS AS YOU'D EXPLAIN IT TO A COLLEAGUE]

Generate:

1. PROCESS NARRATIVE

Numbered steps with actor, action, system, input, and output for each step.

2. DECISION POINTS

For each decision: criteria, yes path, no path.

3. EXCEPTION PATHS

For each step: what could go wrong, exception handling, escalation.

4. HANDOFF POINTS

Every point where responsibility transfers between roles or systems.

5. MERMAID FLOWCHART

Generate a Mermaid flowchart with decision diamonds and actor annotations.

6. PROCESS METADATA

Estimated cycle time, SLA targets, key metrics, improvement opportunities.

SPEAKER NOTES

The Mermaid flowchart generation is visually impressive – demo it live in Claude.

Use a cross-functional process (like a procurement approval) for maximum impact.

The exception path section often surfaces risks that stakeholders hadn't considered.

Ask participants: "What process in your current project would benefit most from this?"

3.3 Decision Logs and Trade-off Matrices

Every project makes dozens of decisions — most undocumented, leading to revisited debates and scope confusion. Claude can help capture and structure decisions instantly.

Real-time decision documentation: After any meeting where a decision was made, feed Claude the context and generate a formal decision record. Include the decision,

alternatives considered, rationale, who decided, implications, and reversibility.

Trade-off matrices: When facing a decision with multiple options, define criteria and options – Claude generates a structured evaluation including trade-offs you might not have considered.

The “decisions register” pattern: Maintain a running log by pasting previous decisions when documenting new ones. Ask Claude to check consistency. This creates institutional memory within a project.

Why this matters: The #1 cause of scope creep is revisited decisions. A well-maintained decisions log with clear rationale stops the “but I thought we decided...” conversations.

KEY TAKEAWAYS

- Document decisions immediately after meetings – context fades fast
- Trade-off matrices with defined criteria make complex decisions transparent
- Running decisions registers help catch inconsistencies across a project
- Decisions logs are the #1 defence against scope creep and revisited debates

PROMPT TEMPLATE: DECISION RECORD GENERATOR

You are a senior BA documenting a project decision.

Context:

- Project: [PROJECT NAME]
- Decision topic: [WHAT WAS BEING DECIDED]

- Meeting/discussion: [DATE, ATTENDEES' ROLES]
- Background: [WHY THIS DECISION WAS NEEDED]

Generate a formal decision record:

- ****Decision ID:**** [AUTO-INCREMENT]
- ****Decision:**** [CLEAR STATEMENT]
- ****Alternatives considered:**** [LIST WITH BRIEF PROS/CONS]
- ****Rationale:**** [WHY THIS OPTION WAS CHOSEN]
- ****Decided by:**** [ROLE, NOT NAME]
- ****Date:**** [DATE]
- ****Implications:**** [WHAT THIS MEANS FOR THE PROJECT]
- ****Reversibility:**** [EASILY REVERSIBLE / REVERSIBLE WITH EFFORT / IRREVERSIBLE]
- ****Dependencies:**** [DECISIONS THIS DEPENDS ON OR THAT DEPEND ON THIS]

If I provide previous decisions, check for consistency conflicts.

SPEAKER NOTES

This is one of the most immediately useful templates in the course — participants can use it the same day.

Demo with a realistic scenario: “We decided to use microservices instead of monolith.” Show how Claude structures the rationale.

Emphasise the reversibility field — it forces teams to think about commitment level.

The consistency check across decisions is the advanced move — show it with 3-4 previous decisions.

3.4 Editing, Tone-Shifting, and Audience Adaptation

One of Claude's most underused capabilities: rewriting the same content for different audiences. You write it once; Claude adapts it for every reader.

The audience adaptation workflow: Write in whatever form is most natural. Then ask Claude to rewrite for: C-suite (concise, outcome-focused, no jargon), dev team (detailed, precise), end users (empathetic, benefit-focused), or procurement (requirements-focused, compliance).

Tone-shifting: From passive to assertive, hedging to confident, technical to accessible. This is especially valuable when you've written in your natural analytical style but need to present to a non-technical audience.

The register ladder: Think of communication registers as a ladder: academic (dense, formal), professional (structured, clear), conversational (accessible, warm), executive (terse, action-oriented). Ask Claude to move your content up or down the ladder as needed.

Practical value: A senior BA writes one analysis and needs 4 audience-specific versions. Without Claude: 4 writing sessions. With Claude: 1 writing session + 4 prompts — 60-70% time saving.

KEY TAKEAWAYS

- Write once; use Claude to adapt for each audience
- Four key audience types: executive, technical, end-user, vendor
- Tone-shifting handles passive→assertive, hedging→confident, technical→accessible
- The register ladder: academic → professional → conversational → executive
- Single analysis → multiple communications = 60-70% time saving

PROMPT TEMPLATE: AUDIENCE ADAPTATION

You are a communications specialist adapting BA content for different audiences.

Here is the original content:

[PASTE YOUR CONTENT]

Rewrite this for the following audience: [CHOOSE ONE]

- ****C-SUITE:**** Max 200 words. Lead with business impact. No technical jargon. End with a clear ask or recommendation.
- ****DEV TEAM:**** Be precise and technical. Include specifics about data, integrations, and constraints. Use their language.
- ****END USERS:**** Lead with benefits. Use empathetic, accessible language. Focus on what changes for them and why it's better.
- ****PROCUREMENT/VENDOR:**** Focus on requirements, SLAs, compliance. Use formal, contractual language.

After rewriting, note: what was removed (and why), what was added (and why), key framing differences.

SPEAKER NOTES

Demo this with a single paragraph of technical analysis. Show all four rewrites side by side.

The “what was removed and why” section teaches participants about audience awareness.

This is often the lesson where participants say “I’m going to use this today.”

Ask participants to bring a real piece of writing and adapt it for two audiences live.

MODULE EXERCISE: DOCUMENTATION SPRINT

1. Generate a complete BRD using the section-by-section approach
2. Document one process using the Process Documentation Generator (include Mermaid flowchart)
3. Create a decision record for one key project decision
4. Rewrite the BRD executive summary for three audiences: project sponsor, dev lead, end-user rep
5. Compare Claude-assisted output against manually written documents

Deliverable: Complete BRD, process document with flowchart, decision record, and 3 audience-adapted summaries.

Module 4

Data Analysis & Insight Generation

Interpret datasets, generate SQL, build pivot logic, and narrate findings.

4.1 Exploratory Data Analysis with Claude

BAs frequently receive data exports, reports, or database schemas and need to make sense of them quickly. Claude accelerates the exploratory phase dramatically.

Schema-first analysis: Paste column headers, data types, or a schema. Claude suggests what the data represents, which columns are analytically interesting, potential join keys, data quality checks to run first, and hypotheses worth investigating.

Small dataset analysis: For datasets under ~100 rows, paste the data directly into Claude. It can spot patterns, calculate basic statistics, identify outliers, and suggest visualisations. For larger datasets, describe summary statistics and let Claude guide your deeper analysis.

KPI definition: Give Claude business context and available data. It suggests KPIs with definitions, calculation methods, and data sources — valuable when stakeholders ask for “a dashboard” without defining what to measure.

The data profiling shortcut: Before diving into analysis, ask Claude to generate a data profiling checklist: completeness

(nulls), uniqueness (duplicates), validity (format/range), consistency (cross-field logic), and timeliness (data freshness). This saves hours of discovery.

KEY TAKEAWAYS

- Schema-first analysis helps Claude suggest what's analytically interesting before you query
- Small datasets (<100 rows) can be analysed directly in Claude
- KPI definition with business context turns vague dashboard requests into measurable specs
- Data profiling checklist: completeness, uniqueness, validity, consistency, timeliness

PROMPT TEMPLATE: EXPLORATORY DATA ANALYSIS

You are a data-savvy Business Analyst performing exploratory analysis.

Here is the data schema / column headers / sample data:

[PASTE SCHEMA, HEADERS, OR SAMPLE DATA]

Business context: [WHAT THIS DATA RELATES TO]

Provide:

1. ****Data description:**** What this dataset likely represents and key entities
2. ****Analytically interesting columns:**** Which fields are most useful for analysis and why
3. ****Data quality checks:**** Specific checks to run before analysis (nulls, duplicates, outliers, format issues)
4. ****Suggested KPIs:**** Based on business context, what should we measure?
5. ****Hypotheses:**** 3-5 questions this data could help answer
6. ****Join opportunities:**** If applicable, what other data sources would enrich this?

7. ****Visualisation suggestions:**** What charts would reveal the most insight?

SPEAKER NOTES

Start with a real schema — an e-commerce orders table works well for everyone.

Demo the data profiling checklist — it's practical and immediately applicable.

The KPI suggestion technique is the real star here — it solves the "just build me a dashboard" problem.

Ask participants to paste a schema from their own projects and compare suggestions.

4.2 SQL and Formula Generation

Describe what you need in plain English, and Claude writes the query. This is one of Claude's most reliable capabilities — and one of the biggest BA time-savers.

SQL from English: Tell Claude your database schema and describe the data you want in business terms. Claude generates SQL including JOINS, window functions, CTEs — and explains what each part does.

Excel formulas: Describe your spreadsheet layout and calculation needed. Claude generates VLOOKUP, INDEX/MATCH, SUMIFS, pivot table configurations, and VBA macros.

The schema-context trick: For complex queries, paste the full schema (table names, column names, types, relationships) once at the start. Then ask multiple questions. Claude uses the schema context for every subsequent query — this is where the context window shines.

Validation is essential: Always review generated SQL before running on production. Ask Claude to explain the logic, check for performance issues, and confirm it handles edge cases (nulls, duplicates, empty results).

KEY TAKEAWAYS

- Describe data needs in business English; Claude writes the SQL
- Claude handles JOINS, window functions, CTEs – and explains them
- Works for Excel formulas too: VLOOKUP, SUMIFS, pivot configs, VBA
- Paste full schema once, then ask multiple questions in the same conversation
- Always validate SQL before running on production

PROMPT TEMPLATE: SQL GENERATOR

You are a database analyst helping a Business Analyst extract data.

Database schema:

[PASTE TABLE NAMES, COLUMNS, TYPES, AND RELATIONSHIPS]

Business question: [DESCRIBE WHAT DATA YOU NEED IN PLAIN ENGLISH]

Generate:

1. **SQL query** (use CTEs for readability if complex)
2. **Line-by-line explanation** of what each part does
3. **Edge case handling:** How does this handle nulls, duplicates, empty results?
4. **Performance notes:** Any indexes that would help, estimated complexity

5. ****Validation query:**** A simpler query to spot-check the results

SPEAKER NOTES

This is universally popular. Even BAs who know SQL use this to write complex queries faster.

Demo: start with a 3-table schema, ask a business question, show the generated SQL with explanation.

The validation query idea is a power move — demo it as a safety net.

For the Excel demo, show a SUMIFS with multiple criteria — it's immediately useful.

4.3 Interpreting and Narrating Data

Numbers don't speak for themselves. The BA's job is to turn data into a narrative that drives decisions. Claude is remarkably good at this — if you provide context.

Data narration workflow: Paste data summary with business context and audience. Ask for an executive narrative: headline finding, context (trends, comparisons), anomalies, and recommendations.

The “so what” technique: For every data point, ask: “What does this mean for the business? What should someone do with this information?” This forces Claude beyond description into analysis.

Handling uncertainty: When data is ambiguous, ask for multiple interpretations with likelihood and what additional data would confirm each. This shows analytical rigour rather than false certainty.

The comparison anchor: Raw numbers are hard to interpret. Always ask Claude to provide comparisons: previous period, industry benchmark, target, peer companies. “Revenue is £2.1M” means nothing; “Revenue is £2.1M, up 18% from last quarter and 12% above target” tells a story.

KEY TAKEAWAYS

- Lead with headline finding, then context, anomalies, recommendations
- The ‘so what’ technique pushes Claude from description to actionable analysis
- Multiple interpretations with likelihood builds credibility
- Always anchor numbers with comparisons: period-over-period, benchmark, target

PROMPT TEMPLATE: DATA NARRATIVE GENERATOR

You are a senior analyst writing an executive data narrative.

Data:

[PASTE DATA SUMMARY, KEY METRICS, OR DASHBOARD EXPORT]

Business context: [WHAT THIS DATA RELATES TO, RECENT EVENTS, STRATEGY]

Audience: [WHO WILL READ THIS]

Comparison benchmarks: [PREVIOUS PERIOD, TARGETS, INDUSTRY AVERAGES]

Generate a narrative with:

1. **Headline:** One sentence capturing the most important finding
2. **Context:** How these numbers compare to benchmarks, trends, expectations

3. **Key findings:** 3-5 findings, each with “so what” business implications
4. **Anomalies:** Anything unexpected, with possible explanations
5. **Risks:** What the data suggests could go wrong
6. **Recommendations:** 2-3 specific actions based on the data
7. **Confidence rating:** How reliable are these conclusions? What additional data would strengthen them?

SPEAKER NOTES

Demo with a simple dataset: monthly sales figures for 6 months. Show how context transforms the narrative.

The “so what” technique is the single most impactful data skill – practice it live.

Show the difference between narration with and without comparison anchors.

This lesson bridges the gap between data analysis and stakeholder communication.

4.4 Validating Claude’s Analysis

This is perhaps the most important lesson in the data module: how to verify that Claude’s analysis is correct. Trust but verify is the operating principle.

Common failure modes: Claude can invent statistics, make math errors with percentages and ratios, misinterpret columns, present correlation as causation, and overlook caveats.

Verification techniques: Spot-check numbers manually. Ask Claude to show its working step by step. Cross-reference claims against source data. Run the same question rephrased – inconsistency signals unreliability.

The “confidence calibration” technique: Ask Claude: “Rate your confidence in each finding on a scale of 1-10 and explain why.” Low-confidence items get manual verification. This creates an efficient verification workflow – you don’t verify everything, just the risky parts.

The cross-check pattern: Ask Claude to generate two different analyses of the same data using different methods. If results converge, confidence increases. If they diverge, investigate the discrepancy.

KEY TAKEAWAYS

- Claude can hallucinate statistics, make math errors, and confuse correlation with causation
- Spot-check numbers manually; ask Claude to show its working
- Use confidence self-rating to prioritise what to verify
- Cross-check with multiple methods – convergence increases reliability
- Verification is not optional – it’s part of the AI-augmented workflow

PROMPT TEMPLATE: ANALYSIS VALIDATION

You are a critical data reviewer checking analysis for accuracy.
Here is the analysis to validate:

[PASTE THE ANALYSIS OR FINDINGS]

Source data: [PASTE OR DESCRIBE THE ORIGINAL DATA]

Perform:

1. ****Fact check:**** For each claim with a number, verify the calculation. Show your working.

2. **Logic check:** Are conclusions logically supported by the data? Flag any leaps.
3. **Bias check:** Is the analysis presenting data selectively? What counter-narratives does the data support?
4. **Confidence rating:** Rate each finding 1-10 with explanation.
5. **Missing caveats:** What limitations or disclaimers should be added?
6. **Verification plan:** For low-confidence findings, suggest how to verify independently.

SPEAKER NOTES

This lesson is critical for credibility. BAs who present unverified AI-generated analysis will lose trust fast.

Demo a deliberate error: feed Claude data and show it making an incorrect percentage calculation. Then use the validation prompt to catch it.

The confidence self-rating technique is the most practical takeaway here.

Close with: "AI-augmented analysis is only as good as your verification discipline."

MODULE EXERCISE: DATA-TO-INSIGHT PIPELINE

1. Use Claude for exploratory analysis on a sample dataset
2. Generate 3 SQL queries for specific business questions
3. Write an executive data narrative with comparison anchors
4. Run Analysis Validation – identify at least one finding needing manual verification
5. Present verified findings in a 3-minute executive brief

Deliverable: Exploratory analysis, 3 SQL queries, executive narrative, validation report.

Module 5

Stakeholder Communication

Draft emails, presentations, and status reports tuned to different audiences.

5.1 Executive Status Updates

Senior stakeholders want answers to three questions: Are we on track? What should I worry about? What do you need from me? Every status update should answer these clearly and concisely.

The RAG status pattern: Feed Claude your project data and ask for a Red/Amber/Green status report: overall status with colour, key accomplishments this period, upcoming milestones with dates, risks and issues with mitigations, and decisions or escalations needed.

The “decisions needed” section: Executives read status reports to know if they need to act. Always end with specific asks — even if “no action needed, FYI only.” This frames the executive as a decision-maker, not just a passive recipient.

Frequency and consistency: Using Claude with a consistent template means every weekly update has the same structure. Executives learn where to look for what matters. This consistency builds trust faster than the content itself.

The “one-page” discipline: Executives scan, they don’t read. Tell Claude: “Keep this under 300 words. Use bullet points for risks. Bold the RAG status.” Format constraints force clarity.

KEY TAKEAWAYS

- Three executive questions: On track? What to worry about? What do you need?
- RAG format: overall status, accomplishments, milestones, risks, decisions needed
- Always include a 'decisions needed' section – even if the answer is 'none'
- Consistency builds trust – use the same template every week
- One-page discipline: under 300 words forces clarity

PROMPT TEMPLATE: EXECUTIVE STATUS REPORT

You are a senior BA writing a weekly executive status report.

Project: [PROJECT NAME]

Reporting period: [DATES]

Data:

- Milestones completed: [LIST]
- Milestones upcoming: [LIST WITH DATES]
- Risks/issues: [LIST WITH SEVERITY]
- Budget status: [ON TRACK / VARIANCE]
- Team status: [ANY CAPACITY OR MORALE ISSUES]

Generate a status report (max 300 words) with:

1. **Overall Status:** [GREEN/AMBER/RED] with one-sentence summary
2. **Key Accomplishments:** 3-4 bullet points
3. **Upcoming Milestones:** Next 2 weeks with dates and owners
4. **Risks & Issues:** Table with Risk | Impact | Likelihood | Mitigation | Owner
5. **Decisions Needed:** Specific asks with deadline and context

6. ****FYI Items:**** Non-urgent but noteworthy updates

Tone: Confident, concise, action-oriented. No jargon. No hedging unless genuinely uncertain.

SPEAKER NOTES

Demo with realistic project data. Show the generated report, then critique it together.

The “decisions needed” section is the most impactful improvement most BAs can make to their status reports.

Show a before/after: a typical narrative status update vs. the RAG format. Ask which one an executive would prefer.

Emphasise consistency: the template matters more than any single report.

5.2 Difficult Conversations and Pushback Emails

BAs often need to deliver unwelcome news: scope changes that delay timelines, technical constraints that invalidate stakeholder preferences, or resource shortfalls. Claude helps frame these diplomatically while maintaining clarity.

The diplomatic assertiveness pattern: Tell Claude the situation, your message, the relationship context, and your desired outcome. Claude drafts a message leading with empathy, stating the issue clearly, explaining impact with data, offering alternatives, and ending with a concrete next step.

The multiple-drafts approach: Ask for 2-3 versions — direct (clear, minimal softening), diplomatic (empathetic, relationship-preserving), and collaborative (frames as shared problem-solving). Choose the best fit or blend elements from each.

Handling scope creep diplomatically: One of the most common BA challenges. Claude can frame scope additions in terms of trade-offs: “We can absolutely include Feature X. To maintain the deadline, we’d need to defer Features Y and Z. Alternatively, with 3 additional weeks, we can include all three.” This transforms a “no” into a structured choice.

The pre-send checklist: Before sending a sensitive email, ask Claude to review it for: unintended tone, ambiguous language, missing context, and whether it achieves the stated objective. This is a safety net that takes 30 seconds and prevents hours of relationship repair.

KEY TAKEAWAYS

- Lead with empathy, then state the issue clearly with data
- Always quantify impact and offer alternatives
- Generate multiple drafts – direct, diplomatic, collaborative – and choose the best fit
- Frame scope creep as trade-off choices, not rejections
- Pre-send review catches tone issues before they cause damage

PROMPT TEMPLATE: DIFFICULT CONVERSATION DRAFTER

You are a communications advisor helping a BA navigate a sensitive stakeholder conversation.

Situation: [DESCRIBE THE ISSUE]

Message I need to convey: [THE CORE POINT]

Relationship context: [SENIORITY, HISTORY, POLITICAL DYNAMICS]

Desired outcome: [WHAT I WANT TO HAPPEN AFTER THEY READ THIS]

Channel: [EMAIL / SLACK / IN-PERSON TALKING POINTS]

Generate THREE versions:

- **Version 1 — DIRECT:** Clear, professional, minimal softening. Good when urgency is high or the audience prefers directness.
- **Version 2 — DIPLOMATIC:** Leads with empathy and shared goals. Softens the message without losing clarity. Good for sensitive relationships.
- **Version 3 — COLLABORATIVE:** Frames the issue as a shared challenge. Invites the stakeholder into the solution. Good when you need buy-in.

For each version, note: what this approach prioritises, what it risks, and when to use it.

SPEAKER NOTES

This lesson consistently gets the strongest participant engagement — everyone has a difficult email they need to send.

Demo with a common scenario: a sponsor wants to add scope without extending the timeline.

The three-version technique shows participants they have options. Many BAs default to only one communication style.

Ask participants to bring a real difficult message and draft it together.

Close with: “The 30-second pre-send review is the highest-ROI habit in this entire course.”

5.3 Presentation Outline and Talking Points

BAs present frequently — to steering committees, sprint reviews, vendor demos, and training sessions. Claude structures presentations and generates talking points in minutes.

The presentation framework: Tell Claude: the audience, your key message (one sentence), time limit, and what you want the audience to do afterwards. Claude generates an

outline with slide titles, key points per slide, speaker notes, and transition sentences.

The Q&A prep technique: After generating your outline, ask: “What are the 10 toughest questions this audience might ask? For each, provide a concise, confident answer.” This is invaluable preparation most presenters skip.

The “one-slide summary” test: For any presentation, ask Claude to distil the entire message into one slide with a headline, 3 supporting points, and one call to action. If this single slide doesn’t make sense, the full presentation needs restructuring.

Handling objections in real time: Before a high-stakes presentation, ask Claude to simulate hostile questions from specific stakeholder personas. Practice your responses. This builds confidence and reduces surprises.

KEY TAKEAWAYS

- Specify: audience, key message, time limit, desired action
- Claude generates slide outlines, speaker notes, and transitions
- Q&A anticipation is the highest-value presentation prep
- The one-slide summary test validates your core message
- Simulating hostile questions builds confidence before high-stakes presentations

PROMPT TEMPLATE: PRESENTATION BUILDER

You are a presentation coach helping a BA prepare for a stakeholder presentation.

Presentation context:

- Audience: [WHO, HOW MANY, THEIR CONCERNS]
- Key message: [ONE SENTENCE: WHAT DO YOU WANT THEM TO REMEMBER?]
- Time limit: [MINUTES]
- Desired outcome: [WHAT SHOULD THEY DO AFTER THIS PRESENTATION?]
- Content to cover: [KEY TOPICS / DATA]

Generate:

1. **Slide outline:** Title, 3 key points, and speaker notes for each slide
2. **Opening hook:** A compelling first 30 seconds
3. **Transition sentences:** How each slide connects to the next
4. **Closing and call to action:** Specific ask, not vague “any questions?”
5. **One-slide summary:** Headline + 3 points + CTA – if this slide doesn’t work, flag what’s unclear
6. **Q&A prep:** 10 toughest questions with concise answers
7. **Time allocation:** Suggested minutes per section, including Q&A buffer

SPEAKER NOTES

Demo by building a real presentation outline live. Use a scenario everyone can relate to: recommending a tool or process change.

The Q&A prep section always gets the biggest reaction – participants immediately see the value.

The one-slide summary test is a great teaching moment: if you can’t summarise in one slide, your message isn’t clear enough.

Time allocation is practical and often overlooked – most presenters run out of time before their conclusion.

5.4 Cross-Functional Translation

BAs are translators between business and technology. This is often cited as the core BA skill, and Claude makes it faster, more consistent, and more thorough.

Business-to-technical: Translate stakeholder language into specs, data model implications, API requirements, performance considerations, and integration points. When a stakeholder says “I want real-time updates,” Claude helps you probe: does “real-time” mean sub-second WebSocket updates, polling every 5 minutes, or a daily batch refresh?

Technical-to-business: Explain constraints and architectural decisions in terms of project impact, user experience, timeline, and cost. When an architect says “we need to re-platform the API gateway,” Claude helps translate: “This infrastructure change will take 3 sprints but will reduce page load times by 40% and cut hosting costs by 25%.”

Jargon glossaries: For multi-domain projects, build a cross-functional glossary where each term is defined from each domain’s perspective. “Onboarding” means something different to HR, IT, and Customer Success. Claude generates these multi-perspective glossaries quickly.

The translation audit: After writing any cross-functional document, ask Claude: “Identify any terms in this document that might mean different things to different teams. For each, provide the definition from each stakeholder’s perspective.” This prevents costly misunderstandings.

KEY TAKEAWAYS

- Business→technical: stakeholder language into specs, probing vague terms
- Technical→business: constraints into project impact, user experience, cost
- Cross-functional glossaries prevent expensive misunderstandings
- Translation audits catch ambiguous terms before they cause confusion

PROMPT TEMPLATE: CROSS-FUNCTIONAL TRANSLATOR

You are a bilingual BA who speaks both business and technical fluently.

Translation direction: [BUSINESS-TO-TECHNICAL/
TECHNICAL-TO-BUSINESS]

Source content:

[PASTE THE CONTENT TO TRANSLATE]

Target audience: [ROLE AND TECHNICAL LEVEL]

Provide:

1. ****Translated version:**** Rewritten for the target audience
2. ****Ambiguous terms:**** Terms that might mean different things to different teams, with definitions from each perspective
3. ****Implied requirements:**** Technical or business requirements implied but not stated
4. ****Clarification questions:**** Questions the target audience will likely ask
5. ****Risk flags:**** Anything that might cause misalignment between business and technical teams

SPEAKER NOTES

This lesson resonates especially with BAs working in cross-functional teams or vendor-managed projects.

Demo the “real-time” example: show how a single vague business term maps to 5 different technical implementations.

The glossary exercise is fun and practical — pick a term everyone uses and show how each team defines it differently.

Close with: “Every misalignment between business and tech started as a term both sides thought they agreed on.”

MODULE EXERCISE: COMMUNICATION SPRINT

1. Generate an executive status report using the RAG template
2. Draft a difficult conversation email — generate 3 versions and choose the best
3. Build a presentation outline with Q&A prep for a 15-minute stakeholder update
4. Translate one technical decision for a non-technical sponsor
5. Time yourself and compare against your normal effort for these deliverables

Deliverable: Status report, 3 email versions, presentation outline with Q&A, translated brief, and time comparison.

Module 6

Process Improvement & As-Is/To-Be

*Map current state, identify bottlenecks,
design future workflows with Claude.*

6.1 As-Is Process Extraction

Documenting the current state is often the first step in any process improvement initiative — and the most time-consuming. Claude structures messy descriptions into formal documentation rapidly.

The extraction approach: Tell Claude the process as you’d explain it to a new team member — including informal workarounds. Claude structures it into numbered steps, actors, decision points, inputs/outputs, handoffs, and time estimates.

The “invisible work” discovery: Ask Claude to identify undocumented steps — manual data entry, email approvals, spreadsheet handoffs, workarounds. These invisible steps often represent the biggest improvement opportunities because they’re the ones nobody has thought to optimise.

Stakeholder-specific views: The same process looks different to different participants. Ask Claude to describe the process from each actor’s perspective. This reveals pain points that a single process map misses — what feels

seamless to the approver might involve 30 minutes of data preparation by the requester.

Time and cost mapping: For each step, ask Claude to estimate time, cost, and value-add classification (value-adding, necessary but non-value-adding, or waste). This sets up the waste analysis in the next lesson.

KEY TAKEAWAYS

- Describe processes informally — Claude adds formal structure
- Ask about ‘invisible work’: workarounds, email approvals, spreadsheet handoffs
- Invisible work often represents the biggest improvement opportunities
- Multi-perspective views reveal hidden pain points
- Time/cost/value mapping sets up systematic improvement analysis

PROMPT TEMPLATE: AS-IS PROCESS EXTRACTOR

You are a process analyst documenting a current-state business process.

Process name: [PROCESS NAME]

Description (informal):

[DESCRIBE THE PROCESS AS YOU’D EXPLAIN IT TO A NEW TEAM MEMBER, INCLUDING WORKAROUNDS AND INFORMAL STEPS]

Generate:

1. **Formal process steps:** # | Actor | Action | System/Tool | Input | Output | Estimated Time
2. **Decision points:** For each: criteria, outcomes, who decides

3. ****Handoff points:**** Every transfer between roles or systems, with potential failure points
4. ****Invisible work:**** Undocumented steps (manual entry, email chasing, spreadsheet manipulation, workarounds)
5. ****Pain points:**** Where is friction, delay, or rework most likely?
6. ****Value classification:**** For each step: Value-adding / Necessary non-value-adding / Waste
7. ****Mermaid flowchart:**** Visual representation with decision diamonds and swimlanes
8. ****Process metrics:**** Estimated total cycle time, touch time, wait time, and rework rate

SPEAKER NOTES

Start by having participants describe a process they know well – verbally, informally, with tangents. Then paste it into Claude. The “invisible work” discovery is the highlight. Every process has hidden steps that nobody has documented.

The value classification (value-adding vs. waste) is a bridge to the next lesson on Lean analysis.

Ask: “How many of your current processes have ever been formally documented?” The answer is usually “very few.”

6.2 Bottleneck and Waste Identification

Claude can apply Lean and Six Sigma thinking to identify improvements systematically. This turns what’s normally a facilitated workshop into something a single BA can produce as a draft in 20 minutes.

The eight wastes (TIMWOODS): Transport (unnecessary movement of data/documents), Inventory (work-in-progress piling up), Motion (unnecessary human effort), Waiting (idle time between steps), Over-processing (doing

more than needed), Overproduction (creating output nobody uses), Defects (errors requiring rework), and Skills underutilisation (people doing work below their capability).

Root cause analysis: For each bottleneck, use the “5 Whys” technique to find root causes. Claude is excellent at this because it’s systematic and doesn’t accept the first answer. It keeps pushing deeper.

Quantifying waste: Ask Claude to estimate the cost of each waste category: time lost, rework hours, delay cost, and opportunity cost of skilled people doing low-value work. Even rough estimates make a compelling business case.

The constraint theory view: Beyond Lean, ask Claude to identify the single biggest constraint in the process – the one step that limits the throughput of everything else. Improving this constraint gives more return than optimising anything else.

KEY TAKEAWAYS

- Apply the 8 Lean wastes (TIMWOODS) for systematic identification
- Root cause via ‘5 Whys’ prevents solving symptoms instead of problems
- Quantify waste in time, cost, and opportunity cost for business case
- Identify the single biggest constraint – this gives the highest improvement ROI
- Claude is systematic in a way that’s hard to match in brainstorming sessions

PROMPT TEMPLATE: WASTE AND BOTTLENECK ANALYSIS

You are a Lean process improvement specialist analysing a business process.

As-is process:

[PASTE THE AS-IS PROCESS DOCUMENTATION FROM LESSON 6.1]

Perform:

1. ****Eight Wastes Audit (TIMWOODS):**** For each waste type, identify specific instances in this process with estimated time/cost impact
2. ****Bottleneck identification:**** Which step(s) limit overall throughput? Evidence?
3. ****Root cause analysis (5 Whys):**** For the top 3 bottlenecks, drill to root cause
4. ****Rework loops:**** Where do errors send work backwards? What causes them?
5. ****Wait time analysis:**** Where is work sitting idle? Why?
6. ****Constraint identification:**** What is the single biggest constraint limiting this process?
7. ****Quick wins:**** Improvements that could be implemented in <2 weeks with minimal cost
8. ****Waste cost estimate:**** Rough annual cost of identified waste (time × rate + delay cost + rework cost)

SPEAKER NOTES

TIMWOODS is the framework participants remember long after the course.

Demo with the process documented in 6.1. Show how waste categories apply to real steps.

The 5 Whys demo is powerful: take a common bottleneck (“approvals are slow”) and drill to root cause live.

The waste cost estimate is what gets executive attention — even rough numbers are persuasive.

Close with: “The bottleneck is always the lever. Everything else is just rearranging deck chairs.”

6.3 To-Be Process Design

Claude helps explore solutions systematically before committing to a design. The key is generating options, not jumping to one answer.

Solution generation — three options: Ask for three options: conservative (minimal change, quick wins, low risk), moderate (significant improvement, manageable effort, measured risk), and ambitious (transformational, substantial investment, high potential return). This gives stakeholders a genuine choice rather than a take-it-or-leave-it proposal.

Automation opportunity mapping: For each step in the as-is process, classify as: fully automatable (no human needed), partially automatable (human validates or handles exceptions), AI-assistable (Claude or similar tools accelerate the work), or must remain manual (requires judgement, relationships, or physical presence).

The comparison matrix: Build a structured comparison across all three options: implementation effort, timeline, cost, risk, expected improvement (cycle time, quality, cost), and change management complexity. This is the deliverable stakeholders need to make a decision.

Transition planning: For the recommended option, ask Claude to outline: phased rollout plan, dependencies,

parallel running period, success criteria for each phase, and rollback triggers.

KEY TAKEAWAYS

- Generate 3 options: conservative, moderate, ambitious
- Automation mapping: fully automated, partially automated, AI-assisted, manual-only
- Comparison matrices make stakeholder decisions clearer and more defensible
- Transition planning with phases, success criteria, and rollback triggers de-risks change

PROMPT TEMPLATE: TO-BE PROCESS DESIGNER

You are a process improvement architect designing future-state workflows.

As-is process: [PASTE AS-IS DOCUMENTATION]

Waste analysis: [PASTE KEY FINDINGS FROM LESSON 6.2]

Constraints: [BUDGET, TIMELINE, TECH STACK, ORGANISATIONAL APPETITE FOR CHANGE]

Design THREE to-be options:

- ****Option 1 — CONSERVATIVE:**** Minimal change, quick wins only. Target: implement in <4 weeks.
- ****Option 2 — MODERATE:**** Significant improvement with manageable effort. Target: implement in 2-3 months.
- ****Option 3 — AMBITIOUS:**** Transformational redesign. Target: implement in 6-12 months.

For each option:

- Redesigned process steps (numbered, with actors and systems)
- What changes from the as-is (added, removed, modified, automated)
- Automation opportunity map per step

- Expected improvement metrics (cycle time, quality, cost, capacity)
- Implementation effort (weeks, people, cost estimate)
- Risk and dependencies
- Mermaid flowchart of the to-be process

Then: Comparison matrix across all 3 options and a recommendation with rationale.

SPEAKER NOTES

The three-option technique is a career skill, not just a Claude technique. It transforms the conversation from “accept or reject” to “choose your level of investment.”

Demo the automation opportunity mapping on a familiar process — invoice processing works well.

The comparison matrix is the deliverable that sponsors love — it makes their decision easy and defensible.

Ask participants: “Which option would your organisation choose?” This sparks great discussion about organisational appetite for change.

6.4 Impact Assessment and Change Readiness

Process improvement requires people to change how they work. The best process redesign fails if people resist it. Claude helps prepare for this human side systematically.

Impact assessment per role: For each affected role, map: how their daily work changes, what they lose (even if the lost thing was inefficient — people mourn familiar inefficiency), what they gain, new skills or tools they need, and the emotional impact (anxiety, relief, resistance).

Resistance prediction: Ask Claude to predict likely sources of resistance and suggest mitigation strategies. Common

patterns: loss of control (“the system decides now”), loss of expertise (“my knowledge of the workaround is no longer valuable”), increased visibility (“my work was hidden, now it’s tracked”), and fear of job loss.

Change readiness checklist: Stakeholder buy-in (who’s championing this?), training requirements (who needs what?), communication plan (who hears what, when?), pilot plan (where to test first?), success metrics (how do we know it’s working?), and rollback plan (what if it doesn’t work?).

The ADKAR lens: Ask Claude to assess each stakeholder group across ADKAR: Awareness (do they know why?), Desire (do they want to?), Knowledge (do they know how?), Ability (can they actually do it?), and Reinforcement (will the change stick?). This identifies exactly where each group needs support.

KEY TAKEAWAYS

- Map impact per role: what changes, what’s lost, what’s gained, skills needed
- People resist what surprises them – impact assessments prevent surprises
- Predict resistance patterns: loss of control, expertise, visibility; fear of job loss
- Change readiness covers: buy-in, training, communication, pilot, metrics, rollback
- ADKAR framework identifies exactly where each group needs support

**PROMPT TEMPLATE: IMPACT ASSESSMENT AND
CHANGE READINESS**

You are a change management specialist assessing the impact of a process improvement.

Current process: [PASTE AS-IS SUMMARY]

Proposed change: [PASTE TO-BE SUMMARY]

Affected roles: [LIST ALL ROLES TOUCHED BY THIS CHANGE]

Generate:

1. ****Impact Assessment per Role:**** For each role: daily work changes, what they lose, what they gain, new skills needed, emotional impact prediction
2. ****Resistance Prediction:**** Likely resistance sources, root causes, and mitigation strategies
3. ****ADKAR Analysis:**** For each stakeholder group: Awareness | Desire | Knowledge | Ability | Reinforcement – rate each 1-5 and identify gaps
4. ****Change Readiness Checklist:**** Champion identified? Training plan? Communication plan? Pilot plan? Success metrics? Rollback plan?
5. ****Communication Plan:**** Who hears what, when, through which channel, from whom
6. ****Recommended pilot approach:**** Where to pilot, success criteria, duration, decision point for full rollout

SPEAKER NOTES

This lesson is often underestimated but it's where process improvement projects actually succeed or fail.

The "what they lose" exercise is eye-opening. Even losing an inefficient process creates grief.

Demo ADKAR on a recent change in participants' organisations. It quickly reveals blind spots.

The resistance prediction exercise generates real conversation about organisational dynamics.

Close with: "The process is the easy part. The people are the hard part. Plan for both."

MODULE EXERCISE: PROCESS IMPROVEMENT CYCLE

1. Map an as-is process from your current project (or use the case study)
2. Run waste analysis and identify top 3 bottlenecks with root causes
3. Design 3 to-be options (conservative, moderate, ambitious) with comparison matrix
4. Create impact assessment and change readiness checklist for the recommended option
5. Present your recommendation as a 5-minute business case

Deliverable: As-is document with Mermaid flowchart, waste analysis, 3 to-be options with comparison matrix, impact assessment, change readiness checklist.

Module 7

Prompt Engineering Patterns for BAs

*Advanced techniques for consistent,
BA-grade outputs from Claude.*

7.1 Chain-of-Thought for Complex Analysis

Chain-of-thought prompting forces Claude to show reasoning step by step. This produces better analysis AND creates an audit trail for stakeholders who ask “how did you reach that conclusion?”

When to use: Complex decisions, impact analysis, risk assessment, root cause analysis, and any situation where the reasoning matters as much as the conclusion. In regulated environments, chain-of-thought output serves as auditable documentation.

How to trigger: Add: “Think through this step by step,” “Show your reasoning at each stage,” or “Explain your logic chain from assumption to conclusion.” The more specific you are about what steps to show, the better the output.

The structured reasoning pattern: For complex BA analysis, specify the reasoning structure: “For each option, evaluate: (1) alignment with business objectives, (2) technical feasibility, (3) cost-benefit, (4) risk profile, (5) stakeholder impact. Show your reasoning for each dimension before giving a recommendation.”

When NOT to use: Simple extraction tasks, formatting requests, or situations where you just need the answer. Chain-of-thought adds length and cost. Use it when the reasoning adds value.

KEY TAKEAWAYS

- Chain-of-thought reduces errors AND creates an audit trail
- Use for complex decisions, impact analysis, risk assessment
- Specify the reasoning structure for consistent, thorough analysis
- In regulated environments, CoT provides auditable documentation
- Don't use for simple tasks – it adds length without value

PROMPT TEMPLATE: CHAIN-OF-THOUGHT ANALYSIS

You are a senior BA performing a structured analysis. Think through this step by step, showing your reasoning at each stage.
Decision/analysis: [DESCRIBE WHAT NEEDS TO BE ANALYSED]

Context: [RELEVANT BACKGROUND]

Options: [LIST OPTIONS IF APPLICABLE]

For each option or finding, evaluate systematically:

1. ****Alignment:**** How well does this align with stated business objectives? Evidence?
2. ****Feasibility:**** What are the technical, organisational, and resource constraints? Rating: High/Medium/Low.
3. ****Cost-Benefit:**** What are the costs (direct, indirect, opportunity)? What are the benefits (quantified where possible)?
4. ****Risks:**** What could go wrong? Likelihood and impact for each.
5. ****Stakeholder impact:**** Who benefits, who is disrupted, who needs to change?

Show your reasoning for each dimension. Then synthesise into:

- **Recommendation:** Clear statement
- **Confidence level:** How certain are you, and what would change your recommendation?
- **Key assumption:** The single biggest assumption underlying this recommendation

SPEAKER NOTES

Demo with a real decision: “Should we build custom or buy a SaaS solution?” Show how CoT produces a structured analysis.

Compare: same question with and without chain-of-thought. The difference in depth is dramatic.

The “key assumption” output is a powerful addition — it forces transparency about what the analysis depends on.

In regulated industries (finance, healthcare), the audit trail aspect is a genuine compliance benefit.

7.2 Few-Shot Examples and Templates

Give Claude 1-2 examples of what you want before asking it to produce more. This is the most reliable way to get consistent, format-compliant outputs.

Why it works: Claude is extremely good at pattern matching. One example is often enough; two nail the pattern almost perfectly. This is more reliable than lengthy format descriptions — show, don’t just tell.

Template-based prompting: Provide a literal template with placeholders. This guarantees structural consistency across documents and team members. When your entire BA team uses the same templates, outputs become standardised regardless of who wrote the prompt.

The golden example technique: Take your best existing work — your best BRD section, your best user story, your best status report — and use it as the example. Claude will match the quality, tone, and structure of your best work rather than producing generic output.

Negative examples: Sometimes showing what you DON'T want is as powerful as showing what you do. "Do NOT write acceptance criteria like this: [bad example]. Instead, follow this pattern: [good example]." Negative examples eliminate common failure modes.

KEY TAKEAWAYS

- 1-2 examples is the most reliable way to get consistent output
- Use your best existing documents as examples — Claude matches your quality ceiling
- Template-based prompting guarantees consistency across team members
- Negative examples eliminate common failure modes
- Show, don't just tell — examples beat format descriptions

PROMPT TEMPLATE: FEW-SHOT EXAMPLE PATTERN

I need you to generate [TYPE OF OUTPUT] following the exact pattern shown in my examples.

Example 1 (GOOD):

[PASTE YOUR BEST EXAMPLE]

Example 2 (GOOD):

[PASTE A SECOND GOOD EXAMPLE IF AVAILABLE]

Example of what I do NOT want:

[PASTE A BAD EXAMPLE WITH ANNOTATION: "This is too vague / too long / missing acceptance criteria / etc."]

Now generate [NUMBER] more following the same pattern for:
[YOUR INPUT DATA / REQUIREMENTS]

Match the tone, structure, level of detail, and format exactly. If anything in my examples is inconsistent, follow Example 1.

SPEAKER NOTES

This is the technique that makes Claude outputs production-ready rather than generic.

Demo: take a well-written user story from a participant and use it as the few-shot example. Generate 5 more. The quality difference vs. generic generation is obvious.

The negative example technique is underused — show how it eliminates the "I keep getting vague acceptance criteria" problem.

Homework: build a "golden examples" folder with your best work for each deliverable type.

7.3 Structured Output: JSON, Tables, XML

When output needs to feed into tools — Jira, Confluence, Excel, BPM systems — you need structured output, not prose. Claude handles this reliably when given clear schema instructions.

The consistency trick: Add: "Return ONLY the [JSON/table/CSV]. No preamble, no explanation, no markdown formatting." This gives clean, copy-paste-ready output that can go directly into your tools.

JSON for tool imports: Jira bulk import, API payloads, configuration files. Specify the exact schema: field names, types, required vs. optional. Claude follows JSON schemas precisely.

Tables for reporting: Specify columns, data types, and sort order. Claude generates markdown tables that paste cleanly into Confluence, Notion, and SharePoint.

CSV/XML for integrations: For data migration, ETL pipelines, or system imports. Specify delimiters, encoding, and header format. Claude generates compliant output.

The schema-first pattern: Define the output schema before the content. This inverts the usual approach (content first, format later) but produces dramatically more consistent results.

KEY TAKEAWAYS

- JSON for tool imports, tables for reporting, CSV/XML for integrations
- Specify the exact schema or column structure upfront
- Add 'Return ONLY the [format]' for clean, tool-ready output
- Schema-first pattern: define structure before content for consistency
- This enables Claude outputs to flow directly into your tool ecosystem

PROMPT TEMPLATE: STRUCTURED OUTPUT GENERATOR

Generate structured output for tool import.

Output format: [JSON / CSV / XML / MARKDOWN TABLE]

Schema:

[DEFINE EXACT FIELDS, TYPES, AND CONSTRAINTS]

Example of expected output:

[ONE EXAMPLE ROW/OBJECT]

Input data:

[PASTE SOURCE DATA]

Return ONLY the [FORMAT]. No preamble, no explanation, no markdown code fences.

Ensure: valid syntax, consistent types, no null values unless schema allows them.

SPEAKER NOTES

Demo: generate Jira-importable JSON for 10 user stories. Show the direct import.

The “no preamble” instruction is critical – without it, Claude wraps output in explanation that breaks parsing.

CSV generation is useful for Excel users who want to avoid manual data entry.

Show how schema-first prompting eliminates the “wrong format, try again” loop.

7.4 Iterative Refinement and Critique Loops

The most powerful technique in the entire course: use Claude to review and improve its own output. Multi-pass refinement consistently produces higher quality than any single-pass prompt.

The refinement loop: Step 1: Generate initial output. Step 2: Ask Claude to critically review it (“What’s weak, missing, or could be improved?”). Step 3: Produce a revised version

addressing its own feedback. Self-critique typically improves quality by 20-30%.

Persona-based critique: Review from different perspectives: skeptical CTO (“What’s technically naive?”), detail-oriented QA lead (“What’s untestable or ambiguous?”), cost-conscious CFO (“Where’s the business case weak?”). Each persona catches different issues.

The “red team” technique: Ask: “Try to break this. Find every flaw, every assumption, every way this could fail.” This is especially valuable for business cases, proposals, and requirements documents that will face scrutiny.

When to stop: 2-3 passes is the sweet spot. Beyond that, you hit diminishing returns — changes become cosmetic rather than substantive. If the third pass doesn’t produce meaningful improvements, you’re done.

The team review simulation: Before a peer review or stakeholder review, ask Claude to simulate the review process: “Review this as [specific colleague’s role] would. What feedback would they give?” This lets you pre-address concerns before the meeting.

KEY TAKEAWAYS

- Self-critique improves output by 20-30%
- Persona-based critique catches different issue types per perspective
- Red teaming finds flaws in business cases, proposals, and project plans
- 2-3 passes is the sweet spot — beyond that, diminishing returns
- Simulating peer review before the actual review eliminates surprises

PROMPT TEMPLATE: ITERATIVE REFINEMENT

I'm going to use a multi-pass refinement process. Here is my initial output:

[PASTE YOUR INITIAL CLAUDE OUTPUT OR YOUR OWN DRAFT]

PASS 1 – SELF-CRITIQUE:

Review this critically. For each issue, specify: what's wrong, why it matters, and how to fix it.

Categories: Completeness, Accuracy, Clarity, Actionability, Audience fit, Consistency.

PASS 2 – PERSONA CRITIQUE:

Review as:

- Skeptical CTO: What's technically naive or unrealistic?
- QA Lead: What's untestable, ambiguous, or has gaps?
- CFO: Where is the business case weak or unsupported?

PASS 3 – RED TEAM:

Try to break this. Find every flaw, every unstated assumption, every way this could fail in practice.

Finally: Produce a REVISED VERSION incorporating all feedback. Mark what changed and why.

SPEAKER NOTES

This is the technique that separates good AI users from great ones.

Demo live: generate a business case, then run all three critique passes. Show the quality improvement at each stage.

The persona critique is powerful because it surfaces issues no single reviewer would catch.

Participants should practice this on their own work — it's just as effective on human-written content.

Close with: “The first draft is never the final draft. With Claude, the second and third drafts take minutes, not hours.”

MODULE EXERCISE: BUILD YOUR BA PROMPT LIBRARY

1. Create 5 production-ready prompt templates covering: elicitation, documentation, data analysis, communication, process improvement
2. Each should use at least 2 advanced techniques from this module (CoT, few-shot, structured output, refinement)
3. Test each template with real project data and refine through 2+ iterations
4. Document each template: purpose, when to use, example input, expected output, tips and pitfalls
5. Share your library with the group and collect feedback

Deliverable: Documented prompt library with 5 templates. This becomes your personal BA toolkit.

Module 8

Capstone — End-to-End BA Workflow

Apply everything in a realistic project — from kickoff to sign-off, fully Claude-augmented.

8.1 Project Kickoff: Context Gathering and Scoping

The capstone begins where every project begins — with a charter, a brief, and a lot of ambiguity. Your job is to turn that ambiguity into clarity, fast.

Charter analysis: Feed Claude the charter and ask it to: summarise the business case in one paragraph, identify stated vs. implied scope, list what's NOT mentioned but probably needs to be, flag visible risks, and generate a stakeholder map with roles, interests, and influence levels.

The “day one deliverables” technique: Before your first stakeholder meeting, use Claude to generate: project glossary (defining every domain term to prevent misunderstanding), interview schedule with role-specific questions, preliminary risk register seeded from charter analysis, scope confirmation document (what's in, what's out, what's deferred), and a RACI draft.

Scope boundaries: Ask Claude to explicitly list out-of-scope items and “future phase” items. This prevents scope creep from day one. The act of documenting what's excluded is as valuable as documenting what's included.

The “project health check”: Before the first sprint, ask Claude to assess the charter for: clarity of business case, measurability of success criteria, completeness of stakeholder identification, and realism of timeline. This surfaces risks before they become problems.

KEY TAKEAWAYS

- Feed Claude the charter for business case, scope, risks, stakeholder map
- Day one deliverables demonstrate immediate BA value and set the project’s foundations
- Explicit out-of-scope items prevent scope creep from day one
- Charter health check surfaces risks before the first sprint
- The glossary is a small deliverable with outsized impact on team alignment

PROMPT TEMPLATE: PROJECT KICKOFF ANALYSER

You are a senior BA performing a project kickoff analysis.

Project charter:

[PASTE PROJECT CHARTER OR BRIEF]

Generate a complete kickoff analysis:

1. ****Business case summary:**** One paragraph, plain language
2. ****Scope analysis:**** Stated scope | Implied scope | Likely out-of-scope | Deferred to future phases
3. ****Stakeholder map:**** Role | Interest | Influence (H/M/L) | Likely concerns | Engagement strategy
4. ****Risk register (preliminary):**** Risk | Likelihood | Impact | Mitigation | Owner

5. ****Project glossary:**** Every domain term that could be misunderstood, with clear definitions
6. ****Charter health check:**** Rate 1-5 on: Business case clarity | Success measurability | Stakeholder completeness | Timeline realism | Scope clarity. Flag anything below 3.
7. ****Day one action items:**** What should the BA do in the first 48 hours?

SPEAKER NOTES

This is where everything comes together. Participants use techniques from every previous module.

Provide a realistic but slightly ambiguous charter — the ambiguity is the point.

The charter health check is a confidence-builder: participants see they can assess project readiness systematically.

Day one deliverables set the tone for the entire project — emphasise that first impressions matter.

8.2 Requirements and Documentation Sprint

A timed sprint: produce a full requirements set, user stories, and BRD in under 90 minutes. This is where you prove the productivity claim.

The sprint sequence: 0-15 min: Extract requirements from the case study transcript. 15-30: Run gap analysis and ambiguity detection. 30-50: Generate user stories with acceptance criteria, then refine with a second pass. 50-75: Generate BRD section by section, feeding each section as context for the next. 75-90: Final quality review using persona-based critique.

The productivity comparison: A typical BA might take 2-3 days manually to produce this package. Your target is 90

minutes. This 10-15× speedup is the core value proposition of AI-augmented BA work. Track your actual time for the ROI calculation in Lesson 8.4.

Quality vs. speed trade-off: Speed without quality is worthless. The sprint isn't about cutting corners — it's about using Claude for the heavy lifting (first drafts, structure, completeness checks) so you can focus your time on judgement, refinement, and stakeholder context.

The assembly technique: Rather than asking Claude to generate a complete BRD in one go, generate each section separately with full context. Then assemble the sections, asking Claude to check for consistency, tone, and cross-references between sections.

KEY TAKEAWAYS

- Full requirements package in 90 minutes: extract → gap analysis → stories → BRD → review
- 10-15× speedup vs. manual work is the core value proposition
- Speed without quality is worthless — Claude handles drafts, you add judgement
- Section-by-section assembly with consistency checks produces the best BRDs
- Track time vs. manual effort — this is your productivity case

**PROMPT TEMPLATE: DOCUMENTATION SPRINT
SEQUENCE**

You are a senior BA in a timed documentation sprint. I will provide inputs in sequence.

SPRINT STEP 1: Requirements Extraction

Transcript: [PASTE]

Extract all requirements with categorisation, confidence, and ambiguity flags.

SPRINT STEP 2: Gap Analysis

[Paste Step 1 output]

Run gap analysis: ambiguity audit, NFR check, adversarial review.

SPRINT STEP 3: User Stories

[Paste refined requirements]

Generate user stories with Given/When/Then ACs, size estimates, and dependencies.

SPRINT STEP 4: BRD Assembly

[Paste all previous outputs]

Generate BRD section by section. Check cross-references and consistency.

SPRINT STEP 5: Quality Review

[Paste complete BRD]

Review as: QA Lead, Solutions Architect, End User. Flag issues and produce revised version.

SPEAKER NOTES

This is the capstone's centrepiece. Time it. Create urgency.

The sprint sequence builds on every technique taught in Modules 2, 3, and 7.

Have participants work in pairs — one drives, one reviews. Swap at Step 3.

The assembly technique is the professional move — it produces a coherent document rather than a patchwork.

Track actual time and compare at the end. The speed difference is the course's proof point.

8.3 Stakeholder Review Simulation

One of Claude's most underused capabilities: simulating stakeholder conversations. Practice defending your requirements before the real meeting.

The role-play technique: Ask Claude to act as a difficult executive who questions ROI, a skeptical architect who challenges technical feasibility, a resistant end user who doesn't want their process to change, or a budget-conscious sponsor who pushes for scope cuts. Present your work and let Claude push back.

Post-role-play analysis: After the simulation, ask Claude to step out of character and analyse: what objections came up, how well you addressed them, what the stakeholder's real underlying concerns were (often different from stated objections), and what preparation would have helped.

The change request simulation: Simulate a stakeholder requesting a major scope change mid-project. Practice your response: acknowledge, assess impact, present options, recommend, and document. This builds negotiation muscle memory.

Group simulation: In a training setting, have one participant present while Claude (operated by another participant) plays the difficult stakeholder. The group observes and debriefs together. This creates shared learning from individual practice.

KEY TAKEAWAYS

- Claude simulates difficult stakeholders — practice before the real meeting
- Role-play with: resistant executive, skeptical architect, concerned end user
- Post-role-play analysis reveals underlying concerns vs. stated objections
- Change request simulation builds negotiation muscle memory
- Group simulation creates shared learning and builds team capability

PROMPT TEMPLATE: STAKEHOLDER SIMULATION

You are going to role-play as a challenging stakeholder. Stay in character throughout.

Your role: [CHOOSE ONE]

- **DIFFICULT EXECUTIVE:** You're time-poor, skeptical of the ROI, and will challenge any vague claim. You want hard numbers and clear timelines.
- **SKEPTICAL ARCHITECT:** You've seen projects like this fail. You'll challenge technical assumptions, integration complexity, and scalability.
- **RESISTANT END USER:** You've been doing this job for 15 years. You're worried about job security and don't trust "another new system."
- **BUDGET-CONSCIOUS SPONSOR:** Every feature needs a cost justification. You'll push for the minimum viable scope.

Context: [BRIEFLY DESCRIBE THE PROJECT AND WHAT'S BEING PRESENTED]

I'll present my work. You respond in character — push back, ask hard questions, express concerns. Be realistic, not cartoonishly difficult.

After 5-6 exchanges, I'll say "END SIMULATION." Then step out of character and provide:

1. What objections came up and how well they were addressed
2. The stakeholder's real underlying concerns (vs. stated ones)
3. What preparation would have strengthened the presentation
4. Specific suggestions for the next real stakeholder meeting

SPEAKER NOTES

This is the most engaging lesson in the capstone — participants love the role-play format.

Have participants choose the stakeholder type that matches their real upcoming meetings.

The post-simulation debrief is where the real learning happens. Underlying concerns vs. stated objections is a crucial BA skill.

For group exercises, rotate roles: presenter, stakeholder operator, observer. Debrief after each round.

Close with: "Every difficult meeting you can rehearse with Claude is one you walk into with more confidence."

8.4 Retrospective: Measuring Your AI-Augmented Efficiency

The final lesson: measuring and articulating the value of AI-augmented BA work — for your own professional development and for making the business case to your organisation.

Quantifying the ROI: Hours saved per week × hourly cost = direct savings. But also capture: fewer rework cycles

(stakeholders accept first drafts more often), better stakeholder alignment (more complete requirements from day one), more thorough coverage (gap analysis that manual review misses), and faster time-to-first-draft (the speed that impresses sponsors).

The personal prompt library: Your Module 7 library is living intellectual property. Add new prompts for new scenarios, refine based on results, share with your team, and version-control. This library compounds in value – every week you use it, it gets better.

Building organisational capability: Document your approach in a format others can follow. Create a “BA AI Playbook” for your team: when to use Claude, which templates for which tasks, governance rules, and quality checkpoints. This scales your impact beyond your own work.

Continuous improvement: Claude evolves. Set quarterly reminders to test your prompts against new model versions, explore new features (web search, file analysis, memory), and share learnings with your BA community. The BAs who stay current will have compounding advantages.

KEY TAKEAWAYS

- Compare capstone output vs. pre-course baseline: time, quality, completeness
- ROI = time savings + quality improvements + reduced rework + faster delivery
- Your prompt library is living IP – maintain, refine, version-control, and share
- Build a BA AI Playbook for your team to scale impact beyond yourself

→ This skill compounds — the more you use it, the more valuable it becomes

PROMPT TEMPLATE: PERSONAL ROI CALCULATOR

You are helping a BA calculate the ROI of their AI-augmented workflow.

Pre-course baseline:

- Hours per week on: requirements elicitation [X], documentation [X], data analysis [X], communication [X], process work [X]
- Typical rework rate: [X]% of deliverables need significant revision after stakeholder review
- Time to first draft for a BRD: [X] hours

Post-course (with Claude):

- Hours per week on same tasks: [ESTIMATE]
- New rework rate: [ESTIMATE]%
- Time to first draft for a BRD: [ESTIMATE] hours

Calculate:

1. **Weekly time savings:** Hours saved per task category
2. **Annual value:** Time savings × 52 weeks × hourly rate (use [RATE])
3. **Quality improvements:** Reduced rework cycles × average rework cost
4. **Intangible benefits:** Stakeholder satisfaction, coverage completeness, confidence
5. **Net ROI:** Total value minus Claude subscription cost
6. **Break-even point:** How many weeks of use to pay for the subscription

Present as a one-page executive summary suitable for sharing with management.

SPEAKER NOTES

This lesson turns the course from “interesting” to “actionable.” Participants leave with a concrete value proposition.

Have participants calculate their personal ROI live. The numbers are usually compelling.

The prompt library as living IP is a powerful framing. It positions Claude as a professional investment, not a toy.

The BA AI Playbook concept gives senior participants a leadership deliverable they can champion.

Close the entire course with: “You’re not just a BA who knows AI. You’re a BA whose output is consistently faster, more thorough, and more polished. That’s a career differentiator that compounds every week.”

MODULE EXERCISE: FULL DELIVERY PACKAGE

Using the case study project, produce in a single timed session:

- Project kickoff package (charter analysis, scope, stakeholder map, glossary)
- Complete requirements set with traceability
- User stories with acceptance criteria (refined through 2 passes)
- Full BRD (section-by-section with consistency check)
- One process document with Mermaid flowchart
- Executive status report
- Personal prompt library (5 templates, documented with usage notes)

Then: Stakeholder simulation (defend your requirements to a difficult executive), personal ROI calculation, peer review exchange, and 3-minute lightning talk presenting your approach.

Deliverables: Complete project package, documented prompt library, ROI calculation, and peer review feedback. This is your course portfolio.

Mastering Claude for Business Analysts

Designed for enterprise BA teams adopting
AI-augmented workflows.