



The First 30 Days on a New BA Assignment

A practical guide for business analysts

Introduction

A new assignment gives you about a week of goodwill. After that, people expect you to know the project, the people and the vocabulary, and to have an opinion. The BAs who look steady at day 30 are rarely the cleverest in the room. They had a plan for the first month and they kept records from the first hour.

This guide is that plan, written the way I would explain it to a colleague over coffee. It assumes you can already write a requirement and run a meeting. It does not teach you what a stakeholder is. It tells you what to do on Monday morning of a contract or a new job in UK financial services, in what order, and what to have in your hands by the end of each week.

Every firm, person and figure in the examples is invented. "Harbourgate Payments" does not exist, and neither does anyone who works there. The examples are built to look like what you will meet: a settlements process nobody fully owns, three documents that disagree about a cut-off time, and a sponsor who wants "something to look at" by the end of the month.

How the month is laid out

- **Before day 1:** what to ask, and what to set up.
- **Week 1, orientation:** people, problem, documents, two logs.
- **Week 2, evidence:** the trail from source to requirement, a first workshop, contradictions.
- **Week 3, shaping:** a first draft you can defend.
- **Week 4, landing:** a playback, and a handover to delivery.
- **A short section on AI:** where it helps and where it needs a leash.
- **A one-page checklist** closes each week, so you can tear them off and use them.

Three habits run through all of it, and I will keep coming back to them. Write down where every statement came from. Write down who can decide. Write down what you do not know yet, in a place other people can see.

Before day 1

The gap between accepting an assignment and starting it is the cheapest time you will ever have to reduce risk. Nobody is waiting for your output yet. Use it.

Questions for the agency or hiring manager

If you are a contractor, you are usually dealing with an agency and, through them, a hiring manager you have met once. Employees get a manager and an induction pack, which tells you less than you would hope. Either way, you want facts that are not in the job advert. Ask these before you start, by email so the answers exist in writing:

1. **Who is the sponsor, and who do I report to day to day?** They are often different people. Get both names and job titles.
2. **What is the project stage?** Discovery, in flight, or recovering from a problem? "Mid-delivery with a missing BA" and "pre-funding" need entirely different first weeks.
3. **Who did this work before me, and are they still around?** A predecessor who has moved teams is a source. A predecessor who has left leaves a document trail you will have to interpret without them.
4. **What does the team already have?** Ask for any existing business case, requirements, process maps, previous workshop notes and the current backlog. Ask for read access, not summaries.
5. **What tools and access do I need, and how long does access take?** In regulated firms, access to a shared drive or ticketing system can take a week. Request it on day minus five, not day one.
6. **What is expected of me in the first month?** Ask this in the hiring manager's words. If the answer is vague ("get up to speed"), say what you plan to deliver at day 30 and ask whether that matches.
7. **Are there fixed dates?** A regulatory deadline, a committee date, a funding gate. Fixed dates set the shape of the project more than any plan does.
8. **How is AI use handled?** Many firms restrict which tools you can use and what data you can put into them. Ask now, so you do not discover the rule after pasting something into the wrong place.

Keep the reply. If the hiring manager says "no real constraints on tooling" and you later find a ban on external AI tools, that email helps you.

Set up your own working kit

You will carry three documents for the whole month. Create them before day 1, as empty files with headings.

A notebook. Paper or digital. One page per day, dated, with three headings: what I learned, what I need, what I told people. The third heading matters more than it looks. When a stakeholder says in week 4 "you never mentioned that", the notebook settles it.

A stakeholder list. Start it with the names from your agency emails. Columns: name, role, team, what they own, how they affect the project, how you will reach them, and a "met?" date. You will extend it in week 1. For now it is a list of people you must meet.

A questions log. Every question you cannot answer yourself goes here, with an owner and a status. Do not leave questions in your head, in chat, or in the margin of a document. A sample row:

ID	Question	Source that prompted it	Asked of	Date asked	Status	Answer and source
Q-004	Is the settlement cut-off 16:00, 16:30 or 17:00 for same-day processing?	Settlement Process Note v2, p.4 says 16:00; Ops runbook says 16:30	S. Okafor (Settlements Ops)	14 Oct	Open	

The "source that prompted it" column is the habit worth keeping. A question with no source is a hunch, and a hunch is harder to take seriously in a meeting.

Prepare for the first conversation

Before your first stakeholder meeting, write three questions you will ask everyone, whatever their role:

- What is the problem we are trying to solve, in your words?
- Who else should I speak to that I have not listed?
- What would make this project a success for you, and what would make it a failure?

Asking the same questions of everyone gives you something to compare. People rarely give you the same answer, and the differences are where you start.

Read the room before you read the files

If you can, find out how the project is perceived. Is it a flagship, a regulatory obligation, or something senior people would like to see quietly finished? That changes how much attention you get, how willing people are to give you time, and how much political cover you have. You will not get a straight answer in an email. You will start to get one in week 1.

WEEK 1

Orientation

Days 1 to 5. The goal: know who matters, what the problem is, and what you are working from.

You are not producing requirements this week. Anyone who asks for them in week 1 is asking for guesses. What you produce is a map, two logs and a short list of things you do not believe yet.

Map the stakeholders

On day 1, meet your line contact and ask for introductions. By day 3, have a conversation with each of the people on your list, even for fifteen minutes. Use your three standard questions.

Then update the stakeholder list with columns you can only fill in after meeting them:

- **What they own.** A process, a budget, a system, a regulatory obligation.
- **What they can decide.** Distinct from what they own. A process owner may own the process but need a committee to change it.
- **What they want from the project,** in their own words, and what they are worried about.
- **Attitude and influence.** One honest word each. Keep these private. This is a working note, not a document you circulate.

Stakeholder maps drawn in a grid of power and interest are fine, but they hide the most useful finding, which is the gap between the org chart and the real decision route. Ask each person: "If we needed to change this, who would have to say yes?" Write down the answers. After six conversations you will usually find that two or three names keep coming up. Those are your real decision-makers, and they may not be on any project board.

If your firm uses a RACI, you can build the first draft from these conversations. Do not ask anyone to confirm it in week 1. Show it in week 2, once you have heard from enough people to be wrong in only small ways.

Find the real problem

The brief you were given is the sponsor's description of the problem on a particular day. It is useful, but it is a claim, not a fact. Every project has at least three versions of the problem:

- the one in the business case,
- the one the sponsor tells you,
- the one the people doing the work describe.

They overlap, but they are never identical. Your job in week 1 is to write the problem down three times, in plain sentences, and compare.

For Harbourgate, suppose the business case says "automate settlement reconciliation to reduce operational risk". The sponsor, Head of Settlements Operations, says "we miss our cut-off too often and the team works late fixing it". The reconciliation analysts say "we can't tell which payments were sent in which file, so we rebuild the picture from three systems every morning". Same project, three problems: a risk statement, a timeliness complaint, and a traceability gap. A requirement written against the first would miss the other two.

Do not resolve the difference yet. Write the three sentences in your notebook, and take them to your sponsor in week 2 or 3: "I'm hearing three versions of the problem. Which one do you want me to build the first release around?" That is a better question than anything in a requirements template.

Find the decision-makers

Ask three things about every decision that looks important:

1. Who can make it?
2. What do they need in order to make it?
3. When do they next sit down to make decisions?

The third question catches more delays than the first two. A steering group that meets monthly sets the speed of your project, like it or not. If you need a decision in week 3 and the next committee is in week 6, say so on day 4, not day 20.

Read the existing documents critically

On day 2 you will be given a folder. Resist reading in order. Triage it first:

1. **List every document** with its title, owner, date, version and status (draft, approved, superseded). Use a table. This takes an hour and will save you days.
2. **Mark each as one of three types:** a source of fact (a signed-off policy, a system specification, a contract), a statement of intent (a business case, a vision), or an opinion (a slide deck, an email summary).
3. **Read the sources of fact first.** Then the statements of intent. Treat opinions as leads to check.

When you read, keep a short list of what the document claims, and ask about each claim:

- Who wrote it, and what did they have in front of them?
- Is it dated, and is that date still reasonable?

- Does it say how the author knew this, or only that it is so?
- What does it leave out?

A process document that describes an ideal flow with no exception handling was written by someone who wanted the process to work. A system specification that says "the file is generated daily" does not say when, or what happens on bank holidays. The gap is where your questions go.

One caution on versions. Folders often hold "final", "final v2" and "final-approved". Ask for the approved version from the person who owns it, and note the version in every reference you make. A requirement tied to the wrong version of a policy is a defect you will carry for weeks.

Start a decisions log

Make a decisions log on day 1, even if it stays empty for a week. Record every decision that shapes scope, behaviour or priority, including small ones. The point is to stop re-opening decisions and to answer "who agreed that?" without drama.

A sample entry:

ID	Date	Decision	Decided by	Basis / source	Alternatives rejected	Impacts	Status
D-003	16 Oct	First release covers card settlements only; account-to-account is a later release	A. Mensah (Head of Settlements Ops), confirmed in steering note of 15 Oct	Steering note SG-12; funding limited to first phase	Both rail types in release 1 (no capacity)	Scope of requirements R-010 to R-040; interface to payments hub deferred	Agreed

"Decided by" should be a person with authority, not "the team". "Basis" should point to something someone else can open. If a decision is made in a meeting and nobody writes it down, you write it down and send it round the same day. The cost of being corrected is low. The cost of never having written it is high.

Start an assumptions log

The assumptions log is the one nobody keeps and everyone needs. An assumption is something you are treating as true without evidence. They accumulate fast, because you cannot ask about everything on day 3.

ID	Assumption	Why you are making it	Risk if wrong	Owner to confirm	Confirm by	Status
A-005	The settlement file is sent once a day	Process note describes "the daily file" but no exceptions	Intraday settlement would change the design and the reconciliation window	S. Okafor	24 Oct	Unconfirmed

Review it every Friday. An assumption that has been unchallenged for two weeks is starting to behave like a fact, and that is the point at which it gets expensive.

The week 1 rule: no requirements yet

By Friday you will feel pressure to write something. Write the one-page summary instead: the three problem statements, the decision-makers you have identified, the documents you have triaged, the top five open questions, and the assumptions you are working under. Send it to your line contact with "this is my understanding at the end of week 1; please correct anything". That single email will teach you more about the project than another day of reading.

Week 1 checklist

- ☐ Met every person on the stakeholder list, or booked a time with the ones I have not.
- ☐ Asked everyone the same three questions, and recorded the answers.
- ☐ Added "what they own" and "what they can decide" for each stakeholder.
- ☐ Written the problem three ways: business case, sponsor, people doing the work.
- ☐ Identified who actually decides, and when they next meet.
- ☐ Listed every document with owner, date, version and status.
- ☐ Read the sources of fact. Noted claims I cannot yet verify.
- ☐ Decisions log started, with at least one entry.
- ☐ Assumptions log started, with a named person to confirm each.
- ☐ Questions log up to date. Every question has a source and an owner.
- ☐ Sent a one-page "my understanding" note to my line contact.
- ☐ Checked the rules on AI tools and data before using any.

WEEK 2

Evidence

Days 6 to 10. The goal: every statement in your analysis can be traced to a source, and the contradictions are on the table.

By now you have opinions. This week you turn them into evidence. The discipline is simple to state and tedious to keep: nothing goes into a requirement unless you can point to where it came from.

The source-to-requirement trail

A trail has five links. Each one should be a row in a table, not a memory.

1. **Source.** A document, a meeting, a person, a system observation. It gets an ID and a version or date.
2. **Statement.** The exact wording, quoted, with a page or timestamp.
3. **Interpretation.** What you think it means. Written separately from the statement, because you might be wrong.
4. **Requirement.** A behaviour the solution must have, derived from the interpretation and confirmed by the owner.
5. **Test.** How someone will check that the behaviour is present.

The reason to keep the statement apart from the interpretation is that stakeholders correct interpretations cheaply and resent corrections to quotations. "I read this as meaning X; did you mean that?" is a normal conversation. "You've misquoted the policy" is not.

A short extract of the register, using the Harbournate example:

Src ID	Source and version	Exact wording	My reading	Requirement	Owner to confirm
S-014	Settlement Process Note v2, §4.1	"Instructions received after cut-off are processed the next business day."	Late instructions roll to the next business day; no same-day exception	R-021 (draft)	S. Okafor
S-015	Ops runbook, step 7	"Late items may be sent by 16:30 if approved by the duty manager."	A same-day exception exists, controlled by approval	Conflicts with R-021; see Q-004	S. Okafor, Duty Manager role

The second row is how the trail earns its keep. Two documents, both apparently authoritative, say different things. Without the register you would probably have followed whichever you read first.

When a requirement has no source, flag it. Some will be fine: a stakeholder in a meeting said "it must work on mobile" and that is a legitimate source if you recorded it. Others are your own good ideas. Mark those as proposals. A proposal needs an owner to adopt it before it becomes a requirement.

Running your first workshop

You will probably run one workshop this week. It is the first time many stakeholders see you work, so the preparation matters more than the facilitation.

Choose a narrow subject. "Settlement exceptions" is a workshop. "The whole settlement process" is a series. Aim for 90 minutes and one outcome.

Decide the outcome in advance. Write it as a sentence: "By the end, we will have agreed the list of exception types and who owns each." If you cannot write that sentence, you are not ready to run the meeting.

Invite the right people, not all of them. Four to eight. Include the person who does the work, the person who owns the process and, if possible, someone who will be affected by a change. Leave out anyone who will spend the session on status.

Send a pre-read, and keep it short. One page: the purpose, the agenda, three questions you want to settle, and the documents to skim. Anyone who has not read it will tell you by the second agenda item.

An agenda that works:

Time	Item	Output
0:00	Purpose and ground rules (5 min)	Agreement on scope of the session
0:05	Walk the current process as it really runs (30 min)	Annotated process, with variations noted
0:35	Exceptions and pain points (30 min)	Numbered list of exceptions with an owner each
1:05	Decisions and open questions (20 min)	Decisions D-nnn logged, questions Q-nnn assigned
1:25	Recap and next steps (5 min)	Read-back of every decision and action

During the session, write on a shared screen or a board, where people can see and correct it. Ask for examples ("tell me about the last time that happened") instead of generalities. When two people disagree, do not arbitrate. Write both versions down with the names against them and move on; that is a contradiction to resolve outside the room.

Within 24 hours, send the notes. Keep them to decisions, actions with owners and dates, and the open questions. A page and a half is plenty. Add the decisions to the log and the questions to the questions log while you write.

Stakeholders who attended will tell you if you got something wrong. Treat it as a good sign: it means they read it.

Spotting contradictions

Contradictions are normal. A process described by four documents written by four teams in three years will not agree. The skill is to notice them early and to avoid resolving them by guesswork.

Where to look:

- **Numbers that appear in more than one place:** cut-off times, thresholds, retention periods, service levels.
- **Terms used by different teams to mean different things.** "Settled" can mean funds moved, instruction accepted, or reconciled.
- **Documents of different ages describing the same process.** The newer one is not always right; the older one may still be what the system does.
- **What people say against what the system does.** A manual workaround that everyone has forgotten is a favourite.

A fictional example. In week 2 at Harbourgate you find three statements about the daily cut-off for same-day settlement:

Source	Statement	Date	Authority
Settlement Process Note v2	Cut-off is 16:00	March	Process owner's approved note
Ops runbook, step 7	Items can still be sent until 16:30 with duty manager approval	June	Working document, owned by the duty team
Scheduler configuration (screenshot from Tech)	Batch closes at 17:00	Current	What the system actually does

Each one is "true" for a different purpose. The note describes policy, the runbook describes practice, and the scheduler describes behaviour. Which one the new design follows is a business decision, and it is not yours.

Here is how I handle it:

1. **Record all three in the register**, with source IDs and the exact wording.
2. **Do not pick one.** Do not average them either.
3. **Write the question**, in neutral terms: "Three sources give 16:00, 16:30 with approval, and 17:00. Which is the intended cut-off for the new process, and is the 16:30 exception to remain?"
4. **Name who can decide.** Here it is the process owner, with Technology as a consulted party because the 17:00 setting has consequences for the reconciliation window.
5. **Show the evidence to that person** along with the question. A one-line table is enough. People answer faster when they can see why you are asking.
6. **Log the decision** once made, with the basis. And update every requirement the answer touches.

The temptation is to resolve it yourself because you are fairly sure which is right. Don't. If you are wrong, you have built a requirement on your own guess. If you are right, you have taken a decision that belonged to someone else, and people do notice.

Make the week visible

On Friday, send a short note to the sponsor and your line contact: what you learned, what you are now treating as established, the contradictions found and who is deciding them. This is the first point where your questions log becomes useful to someone else. It also turns a quiet analyst into a visibly productive one without adding any invented progress.

Week 2 checklist

- ☐ Source register in place: every source has an ID, version and owner.
- ☐ Exact quotations kept apart from my interpretations.
- ☐ Every draft requirement has a source, or is marked as a proposal.
- ☐ First workshop held, with a stated outcome and a pre-read.
- ☐ Notes sent within 24 hours; decisions and questions logged.
- ☐ Contradictions listed with sources and a named decision-maker.
- ☐ No contradiction resolved by my own guess.
- ☐ Assumptions log reviewed; at least two assumptions confirmed or dropped.
- ☐ Draft RACI shown to at least two stakeholders for correction.
- ☐ Friday note sent to sponsor and line contact.

WEEK 3

Shaping

Days 11 to 15. The goal: a first requirements draft you can defend, with acceptance criteria, traceability and an owner review.

This is the week you write. If weeks 1 and 2 went well, writing is the easy part, because the content is already in your registers.

A first requirements draft

Pick a unit of scope small enough to finish. A release, a process, a capability. Do not draft the whole project. A complete, reviewed 15 requirements are worth more than 120 half-agreed ones.

Each requirement should have:

- **An ID and a short title.**
- **A statement**, one behaviour, in plain language, using "must" for mandatory behaviour and "should" for desirable behaviour. If the sentence has an "and" doing structural work, it is probably two requirements.
- **A source link** to the register.
- **An owner** who can confirm it.
- **A priority**, agreed with the owner and not assigned by you.
- **Acceptance criteria.**
- **Status**: draft, reviewed, agreed, or withdrawn.

Check each statement against four tests. Is it unambiguous: could two competent people read it differently? Is it testable: could someone prove it was met? Is it necessary: what happens if we leave it out? Does it avoid prescribing a solution without a reason?

A weak statement, and a better one:

Weak: "The system must reconcile settlements quickly."

Better: "The system must produce a reconciliation report for the previous business day's settlement file, available to the reconciliation team by 08:00 each business day." (Source S-014, S-016. Owner: Head of Reconciliation.)

"Quickly" cannot be tested. "By 08:00" can. If you do not know the right number, write the requirement with a placeholder and the question: "[time to be confirmed: Q-011]". An honest gap is better than a number you invented.

Acceptance criteria

Acceptance criteria describe, from the user's or owner's side, how you will know a requirement has been met. Write them before you hand the requirement to anyone for build, because writing them is how you find the vague parts.

My preferred structure is Given/When/Then. It makes you state the starting condition, the action and the observable result. A sample for the cut-off example, assuming the owner has confirmed a 16:30 exception with approval:

R-021: Late settlement instruction

AC-021-1. Given a settlement instruction is received at 16:20 on a business day, and the duty manager has approved it, when the daily settlement file is generated, then the instruction is included in that day's file and its status is "Sent: approved late".

AC-021-2. Given a settlement instruction is received at 16:20 on a business day with no approval recorded, when the daily settlement file is generated, then the instruction is excluded, its status is "Held: awaiting next business day", and it appears on the next day's file.

AC-021-3. Given a settlement instruction is received at 16:31 on a business day, when approval is attempted, then the system rejects the approval and displays the cut-off message. (*Open: confirm with owner whether any approver can override this. See Q-004.*)

Notice what the example does. It tests the positive case, the negative case and the boundary. The third criterion carries its own open question, instead of pretending the decision has been made. This is how you stay honest in a draft.

Watch for these gaps in your criteria:

- **No negative case.** What should *not* happen is often where the risk sits.
- **Boundaries.** Exactly at the time, exactly at the limit, exactly on a bank holiday.
- **Roles.** Who is allowed to do this, and what happens when someone else tries?
- **Data states.** Missing values, duplicates, reversed or amended items.
- **Audit.** For anything touching money or customer data, what record must exist afterwards?

Traceability

Traceability sounds like paperwork. It is actually how you find out, in three seconds, what a change will affect.

Keep one matrix with a row per requirement and columns for the links in both directions:

Req	Source(s)	Decision(s)	Acceptance criteria	Design item	Test case	Status
R-021	S-014, S-015	D-006 (16:30 exception retained)	AC-021-1 to -3	Pending	Pending	Draft

If a source changes, you can see which requirements it touches. If a requirement is withdrawn, you can see which tests and design items go with it. If a test fails, you can follow it back to the decision that created the behaviour.

The mistake I see most is building the matrix at the end, from memory, to satisfy a governance gate. By then it is fiction. Build it row by row as each requirement is written, which adds perhaps a minute per requirement.

Review with the owners

A requirement nobody owns is a requirement nobody will defend when the budget is cut. In week 3, take your draft to the owners, one at a time or in small groups, depending on the volume.

For each review:

1. **Send the draft 24 hours ahead**, limited to the requirements that person owns. Do not send them 60 pages and ask for comments.
2. **Ask them to mark every line** as agree, change or disagree. You want a decision per line, not a general "looks fine".
3. **In the meeting, go through disagreements first**. Agreements take care of themselves.
4. **Capture changes live**, and read them back.
5. **Record the outcome**: who reviewed, which version, on what date, with what changes. This is your evidence that the content is agreed.

Expect the review to produce new questions. That is a good outcome. It means the owner read the draft closely enough to spot something you had missed.

One practical point. Owners often say "yes" in a meeting and "no" in an email two days later, after consulting their team. The record is only agreed when it is written and sent back. A reply of "OK, noted" on your summary is enough, and you are entitled to ask for it.

Handle the pressure to hurry

Somewhere in week 3, a stakeholder will ask for the requirements document "as it stands" so they can start design. You will be tempted to hand it over. If it contains unresolved contradictions or unreviewed requirements, label it clearly: status, version, and a list of open items on page one. Send the open items with it, not separately. A draft that announces what is missing is safe. A draft that looks finished and is not will be treated as finished.

Week 3 checklist

- ☐ Scope for the first draft defined, small enough to finish.
- ☐ Every requirement has an ID, one behaviour, a source and an owner.
- ☐ No untestable words left ("quickly", "user-friendly", "appropriate"), or each has a question against it.
- ☐ Acceptance criteria written for each requirement, covering positive, negative and boundary cases.
- ☐ Roles, data states and audit needs considered for money or customer data.
- ☐ Traceability matrix updated row by row, not rebuilt.
- ☐ Each owner has seen their requirements and marked agree, change or disagree.
- ☐ Review outcomes recorded: who, which version, which date.
- ☐ Open items listed on page one of the draft.
- ☐ Decisions and questions logs updated after each review.

WEEK 4

Landing

Days 16 to 20. The goal: a playback the sponsor can act on, and a position from which delivery can begin.

The last week is about being understood. Your analysis is only valuable if the sponsor can act on it, and if the delivery team can build and test from it.

Prepare the playback

A playback is a short meeting where you show what you have learned and what you recommend. It is not a status update, and it is not a walkthrough of every requirement. Keep it to 30 to 45 minutes.

Build it on a short deck or a four-page document. I use this order:

1. **The problem, as I now understand it.** One slide, in the sponsor's language. If the three versions of the problem from week 1 are still different, say so and recommend one.
2. **What I looked at.** Sources, people and workshops, with counts. "Fourteen documents, nine interviews, two workshops" is believable and brief.
3. **What I found.** The three or four findings that matter, with evidence. Contradictions belong here, with the decision needed.
4. **What is agreed and what is not.** Requirements reviewed and agreed, requirements in draft, and decisions still open, with owners and dates.
5. **What I recommend,** including scope for the first release and the things that should wait.
6. **Risks and assumptions,** the five that most affect the plan.
7. **What I need from you:** decisions, access and people, each with a date.

Rehearse with your line contact first. They know what the sponsor reacts to, and they will tell you which slide is carrying too much.

Two rules for the room. Open with the conclusion; do not make people wait for it. And end by asking for the decisions you listed, one at a time, not by asking "any questions?"

What "good" looks like at 30 days

Nobody will tell you the standard, so here is mine. At day 30 you should be able to say, with documents to show:

- **People:** a stakeholder map with owners and decision-makers, a draft RACI that two or more stakeholders have corrected.
- **Problem:** a written problem statement the sponsor has accepted or amended.
- **Evidence:** a source register in which every source has an ID and a version, and every quotation is exact.
- **Logs:** decisions, assumptions, risks and questions, each kept up to date, none with undated or ownerless entries.
- **Requirements:** a reviewed first draft for a defined scope, each requirement with an owner and acceptance criteria.
- **Traceability:** a matrix covering that draft, not a plan to build one.
- **Open items:** a short list that is honest about what is unresolved, who owns it and when it is due.
- **Next steps:** a view of what delivery needs next, including the dependencies.

You do not need a perfect specification. You need a body of work that someone else could pick up without calling you, which is the only real test of documentation.

Set up for delivery and UAT

The last piece of the month is making sure the work can be built and tested.

Hand over to the delivery team. Walk the lead developer and the test lead through the draft. Ask them to read the acceptance criteria as testers would: can they write a test from each one without asking you? Questions at this point are cheap. Questions in build are not.

Agree how change will be handled. Requirements will change. Agree before they do: who may request a change, who assesses its impact (this is where your traceability matrix pays for itself), who approves it, and where it is recorded. A simple change request template with an impact field is enough.

Plan UAT early. User acceptance testing is where business owners find out what they asked for. Find out now:

- Who will carry out UAT, and how much time have they been given?
- What environment and data will they use? In regulated firms, real customer data is usually not available, so test data has to be built.
- How will defects be classified, and who decides whether a defect blocks go-live?
- What counts as "accepted", and who signs it off?

Link your acceptance criteria to the test cases now. UAT scripts written by someone who never saw the criteria tend to test the system the tester imagines instead of the one the owner approved.

Write the handover note. One to two pages. Scope covered, status of each requirement, key decisions and where to find the log, open questions with owners, and the risks you would watch. It is also a good note to send to yourself, because at day 60 you will have forgotten half of it.

Schedule a day-45 checkpoint with your sponsor. Ask for it at the playback, while the sponsor is paying attention.

Week 4 checklist

- ☐ Playback prepared: problem, findings, agreed/open, recommendation, asks.
- ☐ Rehearsed with my line contact.
- ☐ Opened with the conclusion; ended with decisions requested, each with a date.
- ☐ Sponsor's decisions logged and circulated within 24 hours.
- ☐ Stakeholder map, RACI, source register and logs all current.
- ☐ Reviewed requirements flagged as agreed; the rest flagged as draft.
- ☐ Delivery and test leads have read the acceptance criteria.
- ☐ Change process agreed: who requests, who assesses, who approves.
- ☐ UAT questions answered: who, when, environment, data, defects, sign-off.
- ☐ Handover note written; day-45 checkpoint booked.

Using AI safely on an assignment

AI can speed up some of the work in this month and slow down other parts if you let it. I treat it as a fast junior colleague who has read everything and checked nothing. Useful, and not to be left unsupervised.

Rules first

- **Check the firm's rule before the first prompt.** Many regulated firms ban external tools for work data, or allow only an approved internal one. If in doubt, assume that no client or customer data goes in.
- **Do not paste confidential material** into a public tool. Not meeting notes, not extracts, not "just the structure".
- **You are accountable for the output.** "The AI said so" is not a source.

Where it helps

- Turning rough workshop notes you are entitled to process into a tidy draft.
- Suggesting questions you might have missed from a document, as a prompt for your own thinking.
- Reformatting: converting a list into a table, standardising wording across requirements.
- Proposing edge cases for acceptance criteria, which you then evaluate.

Where it needs checking

Anything that looks like a fact: a regulation, a threshold, a name, a quotation, a "typical" figure. AI output reads confidently whether or not it is right. In the Harbourgate example, ask a model to "summarise the settlement cut-off rules" and it may well produce a tidy statement with a single time. That statement may blend the three sources, or invent a fourth. It has no way of knowing which document your project treats as authoritative.

A routine for checking

1. **Split the output into separate claims.** One line each.
2. **For each claim, find the source** in your register. Write the ID and the exact supporting passage.

3. **Mark each claim:** supported, supported by inference (needs confirmation), unsupported, or contradicted.
4. **Remove or rewrite the unsupported ones,** and turn them into questions for the owner.
5. **Record the review:** what you asked, what you checked, what you changed, who approved it.

An example of a claim that would fail. The model says: "Instructions received after cut-off are rejected." Your source (S-014) says they are "processed the next business day". The claim is contradicted. Turned into a question, it becomes: "Are late instructions held for the next day or rejected back to the sender?" Left in the draft, it would have become a requirement with no authority behind it.

Two more habits. Keep your prompts: they are part of the audit trail if anyone asks how a document was produced. And never let AI-generated text enter your logs as a decision or a source. Decisions come from people. Sources come from documents.

The free AI review worksheet and playbook from BA AI Collective cover this in more detail, with a fictional worked case.

Where the toolkits help

Everything in this guide can be done with a notebook and a blank spreadsheet. I did it that way for years. The toolkits exist because I got tired of rebuilding the same tables.

If you have a spare hour and no budget, start with the free material at baaicollective.com: the BA Quick Reference Cheat Sheet, two fictional practice cases (a payments mapping case and a service-request closure case), and the AI review playbook with its worksheet. The practice cases use the same habits as this guide.

Starter toolkit, £30 including UK VAT (21 files)

Starter covers discovery and requirements, which is weeks 1 to 3 of this guide. The files that match:

Guide section	Starter files
Before day 1; week 1	Project One-Pager; Discovery Plan; Stakeholder Map & RACI
Weeks 1 to 3 (quality check)	Requirements Quality Checklist
Week 2 (workshop)	Workshop Agenda & Output Pack
Week 2 to 3 (trail, requirements)	Requirements Backlog; Functional Specification; Data Mapping
Week 3 (acceptance, traceability)	User Story & Acceptance Criteria Library; Traceability Matrix; NFR Checklist
Week 4 (change)	Change Request & Sign-Off Pack
Worked examples	Filled Stakeholder Map & RACI; Filled Requirements Backlog; Worked Functional Specification
Throughout	AI Prompt Library; BA Quick Reference Cheat Sheet

Complete toolkit, £80 including UK VAT (37 files)

Complete includes everything in Starter and adds delivery, governance and contractor planning:

- **Decision Log and RAID Log** for the logs in weeks 1 to 4, with **Filled RAID Log** as a worked example.
- **Status Report** and **Governance Checklist** for week 4 and beyond, with **Filled Status Report** as an example.
- **Business Case** and **Benefits Tracker** if your assignment reaches back to the case for change, with **Filled Business Case** as a worked example.
- **First 90 Days** and **Quick Start**, two of the seven contractor-planning frameworks, which extend this month into the next two. The others cover interview preparation, rate negotiation, extension planning, exit and an IR35 self-assessment structure. The IR35 material is educational and not legal or tax advice.

For an employee on a fixed assignment, Starter is usually enough. Contractors, and anyone running governance as well as analysis, will want Complete, which includes Starter.

Both are Word and Excel files in a ZIP with a README and licence. One person may adapt them for their own work; the library cannot be redistributed. Courses and coaching are not included.

See the file lists, previews and current terms: <https://baaicollective.com/toolkits>

Whatever you buy, or don't, the three habits stay the same: where did it come from, who can decide, and what do I not yet know.

Jennifer Brandt, BA AI Collective