

BA DIGEST

- Why Successful BAs Need a Second Memory
- Prompting Is Not the Same as Thinking
- How BAs Can Stop a Cyberattack
- The Importance of Saying No
- And much more ...

Your regular round-up of useful BA content

*BA Digest is published by Blackmetric Business Solutions
Subscribe for free at www.badigest.co.uk*



Welcome

Welcome to the Q4 edition of BA Digest. It's hard to believe that BA Digest magazine started back in Q4 2020, six years ago now, and existed as a newsletter for several years before that. So much has changed in that time, and things continue to change. Yet some core things have stayed pretty reliably consistent.

Organisational change is still an inherently human endeavour. Humans have probably always used tools to make their efforts more efficient, and that's certainly a pattern that continues today. You'll read articles about AI in this edition, which is genuinely changing the way certain tasks are undertaken. It's increasingly important that, as a community, we are positioned to help our stakeholders use AI within organisational processes effectively, efficiently and ethically.

Yet the human side of business analysis is arguably more important than ever. That might mean building relationships and credibility so that we can get involved earlier and help shape the solutions that organisations deploy. Or it might mean taking the time to understand different stakeholder perspectives, particularly when those perspectives conflict. It might even mean ensuring there's sufficient oversight of an AI process ("human in the loop") to avoid unintended consequences.

None of these are completely new skills. In fact, that's partly the point. They can be easy to overlook when so much of the conversation is about technology and what it can do.

Being a balanced practitioner means understanding where technology can help, while recognising where human judgement, curiosity, empathy and conversation still matter. After all, we are aiming to achieve outcomes that our stakeholders help us define. Or, put differently, the initiatives we work on ultimately exist to serve human beings. So getting to know those human beings is crucial!

As usual, you'll find articles covering a whole range of topics in this edition, written by a diverse group of authors with different experiences and perspectives. I hope you'll find plenty that makes you think, perhaps even something you disagree with(!), and ideas that prompt conversations with your colleagues.

A massive thanks, and kudos, to everyone who has written an article and generously shared their expertise and perspectives. Be sure to connect with the authors whose articles resonate with you. I'm sure they'll appreciate it.

And, of course, a massive thanks to our advertisers for supporting BA Digest too. Their support is crucial, so please do take a look at what they offer.

Enjoy this edition of BA Digest!

Adrian

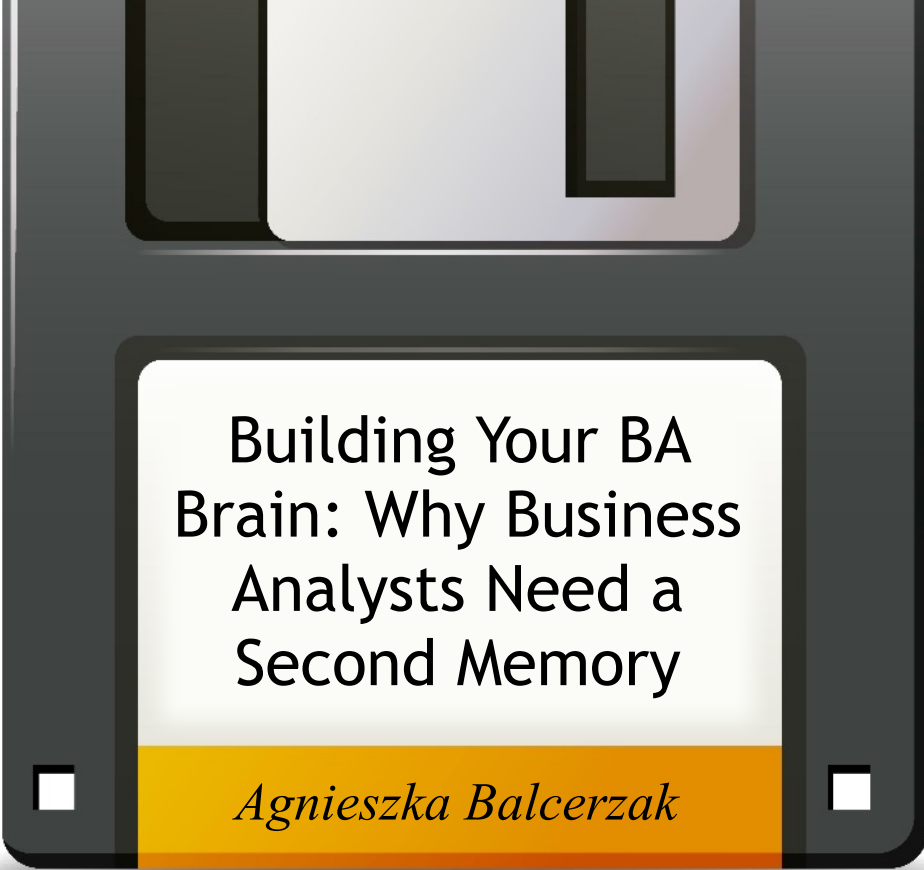
Adrian Reed

Editor-In-Chief, BA Digest

Principal Consultant, Blackmetric

Contents

- 4** Building Your BA Brain: Why Business Analysts Need a Second Memory
- *Agnieszka Balcerzak*
- 7** Finding a Voice in a Crowded Delivery: The BA in Modern Agile Teams
- *Brett Sawley*
- 10** The Business Analyst's Kinder Egg: Mermaid Diagrams! - *Line Karkov*
- 14** Finding Impact as a Business Analyst in Small and Mid-sized Businesses
- *Kent J. Mcdonald*
- 17** Lost Requirements: Lessons from a Toaster Oven - *Sherry Mcmillan*
- 20** Prompting Is Not the Same as Thinking: The Rise of Analysis Debt
- *Erivan De Sena Ramos*
- 23** Speeding Up Decision-making: Transforming Elicitation in Business Analysis
- *Olabisi Odusanya*
- 26** Social Media Loafing: The Feedback We're Missing - *Adrian Reed*
- 29** What You Need to Know to Prepare Your Product Discovery Strategy
- *Nuno Santos*
- 34** Expected Chaos Is Not a Launch Strategy: The Role of Business Architecture in Smoothing the Transition - *Li Yang*
- 38** Using Group Construct Analysis to Understand Stakeholder Worldviews in Ambiguous Situations, Part 22: Sailing by the Stars - *Nick de Voil*
- 41** Why Strong BAs Say No More Than They Say Yes - *Anurita Ruscoe*
- 44** Blueshift Part 2: How Business Analysis Can Stop a Cyberattack - *Mark Cross*
- 48** Prompt Engineering: A Requirements-based Approach to AI Success
- *Darya Miler*
- 52** I Don't Want to Be in a Meeting Room with Technical People - *Rakesh Patange*
- 56** Human-in-the-loop (HITL): How AI Is Accelerating the Transformation of a Business Analyst Role - *Pratik Pitale*
- 60** BA Backstories Interview: Debra Paul - *Interviewed by Christina Lovelock*



Building Your BA Brain: Why Business Analysts Need a Second Memory

Agnieszka Balcerzak

The Folder That Feels like Progress

Many BAs who have used AI tools on real engagements will have hit the same wall. The tool is only as good as the context you hand it, and many practitioners hand it almost nothing. We paste a paragraph into a chat window, get an answer that sounds right, and move on. Nothing persists. The next session starts from zero again, and so does the one after that. I have hit that wall myself, including the time I switched AI tools, or accidentally closed the one window that held all my context, and watched it disappear.

The instinctive fix is to write things down. Keep a folder of markdown files: stakeholder notes, decisions, meeting summaries. It feels like progress, and it is, briefly. Then the folder grows too large to hold in anyone's head, contradictions creep in, and nobody notices until the old information is used to drive a decision. A pile of files is not a scalable or effective 'memory', and it eventually becomes a liability once the author has moved on to a different project or team. When that

happens, whoever inherits the folder usually can't tell what is still relevant and what is not.

Where LLM Wiki Comes from

This is the gap a concept called “[LLM Wiki](#)” is meant to close. An idea Andrej Karpathy has been describing as the natural evolution of prompt engineering, then context engineering. His framing is useful because it is mechanical, not aspirational: the model is the CPU, the context window is memory, tools are peripherals, and an agent orchestrates all of it. [This is described further by Aakash Gupta in his blog.](#) Organisations have spent two years racing to get better at prompting and engineering a context window for a single query. What is usually missing is the second half: a properly maintained memory layer that persists between queries, and that is the role LLM Wiki is designed to fill.

Compiled Once, Not Rediscovered Every Time

The difference between an LLM Wiki and a folder of markdown notes comes down to when the work of organising happens. A folder of notes is retrieved at query time: something like RAG (Retrieval-Augmented Generation) searches across everything and pulls back whatever looks relevant to the current question, rediscovering your context from scratch every single session. An LLM Wiki is compiled at ingest time instead. New information gets read, summarised, and cross-linked into the existing structure as soon as it arrives, not when someone happens

to ask about it later.

That single difference changes three things a BA actually cares about. First, contradictions: when two sources disagree, retrieval-based search has no memory of the earlier version, so it simply serves up whichever fragment matches best, with no way to know about the contradiction. A compiled wiki can flag the old and new versions side by side, because it was already looking at both when the new one was filed.

Second, maintenance: a folder of notes is only as current as the last time someone tidied it, and most knowledge bases decay this way. In an LLM Wiki, the ingest step does the tidying continuously, so the burden doesn't sit entirely on the analyst.

Third, and most practically for BA work, traceability: the questions we need answered are rarely “What is relevant to this query,” they are closer to “Where did this requirement come from?”, “Who else weighed in?”, and “Why did we change it?”. I hear some version of this frustration in almost every mentoring session I run. A system built to compile and cross-reference as it goes can answer these directly. One built only to retrieve cannot.

A Familiar Scenario

In the audits I run, I see a common pattern: a requirement is agreed in a workshop, then quietly narrowed weeks later in a short email exchange nobody thought to circulate widely. Under a retrieval setup, both sit in the same pile, and whichever matches the query's wording best gets surfaced, stale or current, with

no way to tell which. Under an LLM Wiki, the email's arrival is itself an event: it gets read against the existing entry for that requirement, the narrowing gets recorded as a change with its own reasoning attached, and the next person who asks about that requirement gets the current position along with the fact that it moved, and why. That is exactly the kind of quiet, unglamorous drift that consumes so much of a business analyst's actual time.

Still Just Markdown

None of this requires abandoning plain text files, which matters for BAs working across tools and clients. The wiki lives entirely in markdown, readable with no special software, by whichever AI model or agent is in use that week. What changes is not the file format, but the discipline layered on top of it. There are three operations that keep the structure honest, rather than trusting good intentions to do the job manually:

- **Ingest** reads a new source and folds it into the wiki, updating and cross-linking as it goes.
- **Query** answers a question from what is already there, and files the answer back in rather than letting it evaporate the moment the chat window closes.
- **Lint** is the health check, the pass that goes looking for entries that contradict each other, sources that were never properly linked, or sections that have quietly gone stale, and flags them before they cause a wrong decision downstream.

Ingest, query, lint: three operations that sound a great deal like the core discipline of business analysis itself.

What This Means for BAs

For BAs, the implication is straightforward. The context that AI tools need in order to be genuinely useful is exactly the context a BA already produces every day. This includes stakeholder information, the reasoning behind a decision, the history of what changed and why. Context has always been the real output of good analysis, even when the artefact handed over was a requirements document or a process map. Documentation was only ever a snapshot of the thinking, not the thinking itself.

Most attempts to bring AI into business analysis start with the same instinct: automate the tasks we already do a lot of. That instinct is understandable, but it's aimed at the wrong target. Those tasks were never what limited the quality of the work. The tasks we quietly stopped doing, because there was never enough time for them in the first place, are. An LLM Wiki does not just store what we already write down. It creates space to finally do the parts of analysis we've been skipping for years. That is where the real gain sits.

Agnieszka Balcerzak is a business analysis trainer, mentor, and consultant with 20 years at the intersection of IT and business, one of the first certified BAs in Poland. She is currently building theBABrain, an open source implementation of these ideas for business analysts. Connect on [LinkedIn](#).

A silhouette of a person with their arms raised in the air, holding a microphone, set against a background of bright, out-of-focus stage lights in yellow and green. The person is seen from behind, and the overall atmosphere is that of a large crowd at a concert or event.

Finding a Voice in a Crowded Delivery: The BA in Modern Agile Teams

Brett Srawley

Working as a business analyst in agile delivery can sometimes feel like standing in a very busy room, trying to work out when and how to speak. The room is full of people doing important work: product managers, product owners, user researchers,

service designers, delivery managers, etc. Much of what they do looks familiar. It overlaps with the space that BAs have traditionally occupied.

And that can raise an uncomfortable question: where does the BA fit now?

A Crowded Space with Blurred Edges

Agile has brought many positive changes into delivery, but one side effect has been the blurring of professional boundaries. Activities that once sat comfortably with a BA such as understanding the problem, working with stakeholders, shaping options, and clarifying requirements are now shared across multiple roles.

This isn't a criticism of agile methodologies, or of those roles. Collaboration is crucial for agile teams. But in practice, this shared space can leave BAs feeling their contribution is diluted or overlooked. When everyone is doing "a bit of analysis", it can be hard to articulate what makes the BA role distinctive.

When the Role Feels Diluted

If I'm honest, in my view agile has diluted the BA role, at least on the surface. What used to be recognised as a broad, end-to-end discipline has been broken and distributed among a myriad of other roles, each owning a slice of the work. User needs now sit with user research professionals. Vision and priorities often sit with product managers or owners. Flow and momentum sit with delivery managers and similar roles.

It can feel as though the BA role has been dismantled and redistributed.

For experienced BAs, particularly those who have worked in more traditional waterfall models, this can be unsettling. There was once more clarity and sometimes more authority

around what a BA did and where they sat. In agile teams, there is a risk that the BA becomes marginalised, and is pushed towards being the person who purely documents outcomes or tidies the backlog, rather than shaping the thinking that leads to them.

But that isn't the full picture.

Overlap Doesn't Mean Redundancy

The work hasn't disappeared. It has changed shape.

The real value of a BA has never been about producing specific artefacts. It has always been about how we think. BAs are often the ones keeping the wider system in mind—the business context, user needs, technical constraints, risks, dependencies and trade-offs—and making sense of it all.

While other roles go deep into specific areas, BAs tend to look across. We notice where assumptions are being made. We explore what sits outside the immediate scope. We ask the questions that don't always have neat answers.

That perspective still matters, especially as delivery environments become more complex rather than less.

Finding a Voice Rather than Defending a Role

One of the biggest shifts BAs need to make in agile delivery is moving away from defending role boundaries and towards articulating value. The question becomes less about "what does a BA own?" and more about

“what would this team lose if there were no BA here?”

Finding a voice in a crowded delivery space isn't about being the loudest person in the room. It's about knowing when to step in:

- When the problem hasn't been fully understood
- When solutions are running ahead of evidence
- When conversations are happening in silos
- When the wider impact isn't being considered

The BA voice often brings clarity rather than certainty—and that is no small thing.

A Shift in How We Think about Progression

There's an interesting irony here. While agile has fragmented the BA role, it has also opened up far more choice. Under the old waterfall delivery model, career progression for BAs was often limited. If you wanted to progress, the most obvious and sometimes only route was into project management, whether that suited your strengths or not.

Agile has changed that. Many of the roles that now overlap with BA work are genuine progression paths. Product management, discovery, service design, delivery leadership—all build on skills many BAs already have.

Rather than forcing a single upward route, agile allows BAs to choose how they want to grow.

Showing up Strongly as a BA

To find and keep a strong voice in agile delivery, BAs need to be intentional about how they show up:

- Stepping into discovery, not just delivery
- Making their thinking visible, not just their outputs
- Framing conversations around value, risk and outcomes
- Working alongside adjacent roles rather than competing with them
- Being clear about where they add the most value, team by team

Closing Thoughts

The BA role may no longer be neatly defined, and perhaps it never will be again. But that doesn't make it any less important. In a crowded delivery space, teams still need people who can navigate ambiguity, connect perspectives and ask the questions others may not pause to ask.

Finding a voice in agile delivery isn't about reclaiming old territory. It's about embracing how the role has evolved and choosing, deliberately, how and where to be heard.

Brett Srawley is a Business Analyst working in digital delivery within central government, with experience translating complex business needs into clear, practical solutions. Their work focuses on collaboration, problem-solving, and delivering meaningful outcomes across multidisciplinary teams.

LinkedIn Profile:

www.linkedin.com/in/brettsrawley

The Business Analyst's Kinder Egg: Mermaid Diagrams!

Line Karkov



Many readers, particularly those in Europe, will be familiar with Kinder Eggs. These tasty confectionery treats, made by the Italian company Ferrero, famously offer three things: chocolate, a surprise and a toy.

You may be thinking I am mixing my metaphors in the title of this article, after all, mermaids do not lay eggs. In this context, [Mermaid is an open-source, text-based diagramming tool/language](#), although what it has to do with mermaids is still a bit of a mystery to me. However, I've recently realised that, much like a Kinder Egg, it comes with three gifts in one: an iterative process for developing diagrams that can be supported by AI, improved knowledge sharing, and version control for diagrams.

How Does Mermaid Work?

Before I open the Kinder Egg, I

would like to briefly explain what it means for Mermaid to be an open-source standard. To have a Mermaid diagram, you need two things: the diagram written in text using the syntax of the diagram type, and software that can render a visualisation from that text. This is different from most other ways of creating diagrams, where visualisations are developed in a WYSIWYG editor (What You See Is What You Get). The standard is developed and maintained by an open-source project and its community. It supports many diagram types (with new ones released regularly), each with its own syntax.

For example, here is the syntax for a diagram describing the process in the next section, of the diagram type flowchart:

flowchart TD

subgraph Manual[Manual Process]

F[Create Freehand Diagram with Stakeholders]

J[Clarify Ambiguities and Incorporate Changes]

K[Gather Feedback from Stakeholders]

end

subgraph AI[AI-Assisted Process]

G[Upload Screenshot to AI]

H[AI Converts to Mermaid Diagram]

I[Identify Ambiguities and Gaps]

L[Break Down into User Stories/Scenarios with AI]

end

F --> G

G --> H

H --> I

I --> J

J --> K

K --> L

K --> I

This is what the diagram looks like when rendered by the [Mermaid Live Editor](#):

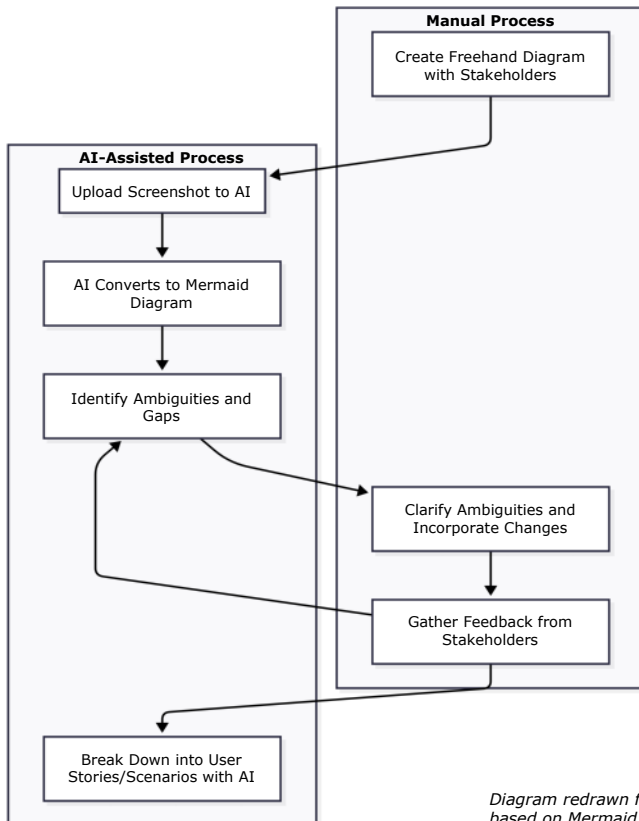


Diagram redrawn for BA Digest,
based on Mermaid Live Editor output

An Iterative Process for Developing Diagrams Supported by AI

For some time now, I have used generative AI to streamline and improve my analysis work, at least when it comes to written deliverables. I have always worked extensively with visualisations but had accepted that the same AI-supported workflow I use for text-based deliverables wouldn't be appropriate for them. That was until recently, when a colleague took an entity relationship diagram he had created in a data modelling tool and, within minutes, added it to the wiki tool in our DevOps platform. How? By uploading a screenshot of the diagram to a generative AI and asking it to produce a Mermaid diagram. That was when I realised that generative AI can also support the process I prefer to follow.

First, I draft a freehand diagram in collaboration with stakeholders. Then, I upload the diagram to a generative AI, which converts it into one or more Mermaid diagrams. Because Mermaid follows a strict syntax, ambiguities or gaps usually emerge. I clarify and incorporate these into the diagram, gather feedback from relevant stakeholders, and refine it further. After that, the AI and I can break it down into user stories or [Gherkin scenarios](#).

Improved Knowledge Sharing

Often, tools for creating visualisations use WYSIWYG editors, which can be cumbersome to work with. The cost of licensing often prevents the whole

team having access, and the default formats are usually proprietary. As a result, I have never truly succeeded in working with visualisations collaboratively. The result has been that I create the diagram in an isolated tool, take a screenshot, and insert it into a document or backlog item.

However, I have discovered that several of our existing collaboration tools support Mermaid, meaning we can work with it without introducing specialised software. The same diagram can be rendered for developers in our wiki and for the business in our standard productivity tools. This also means that diagrams can be edited by others, enabling genuine collaboration.

Enabled Version Control

Traditional approaches to diagramming do not always provide straightforward version control, particularly when diagrams are stored as standalone files. Because Mermaid is an open standard, I can save the syntax for a diagram in any text file. If I switch my Mermaid editor, I can simply paste in the syntax, and it will render my diagram. By storing the diagram as text, I can leverage the versioning functionality that text editors typically offer (including our backlog platform). This means I can finally achieve full version control for diagrams—something I've never had in my nearly 20 years as a business analyst.

With transparency over changes, I can easily allow others to edit the diagrams without losing oversight. Version control is quite simply a prerequisite for genuine collaboration.

The Practical Side

When I use generative AI to draft the syntax, it provides me with a skeleton that I can further develop the diagram from. For example, I can add more activities or swimlanes to a flowchart diagram. I can write this faster in Mermaid than I could move elements around in a WYSIWYG editor. If I want to add other types of elements or if the rendering fails, I can seek guidance from the AI agent.

Mermaid offers excellent documentation and includes an AI feature in the [Mermaid Live Editor](#). However, I prefer using a general-purpose generative AI agent because it can provide broader guidance tailored to my specific role as a business analyst. I always start by declaring that I work according to IIBA® standards. This gives the AI useful context for its suggestions, although I still evaluate those suggestions using my own professional judgement. If I provide it with a hand-drawn sketch in a free format, it can recommend the most suitable diagram type for the purpose.

There are several different editors for Mermaid. The [Mermaid Live Editor](#) supports a wide range of diagram types, and many tools used by development teams also support Mermaid. It's important to note that not all editors support every diagram type—for example, not all support mind maps.

The Promise We Have Heard a Thousand Times but That I Believe to Be True This Time!

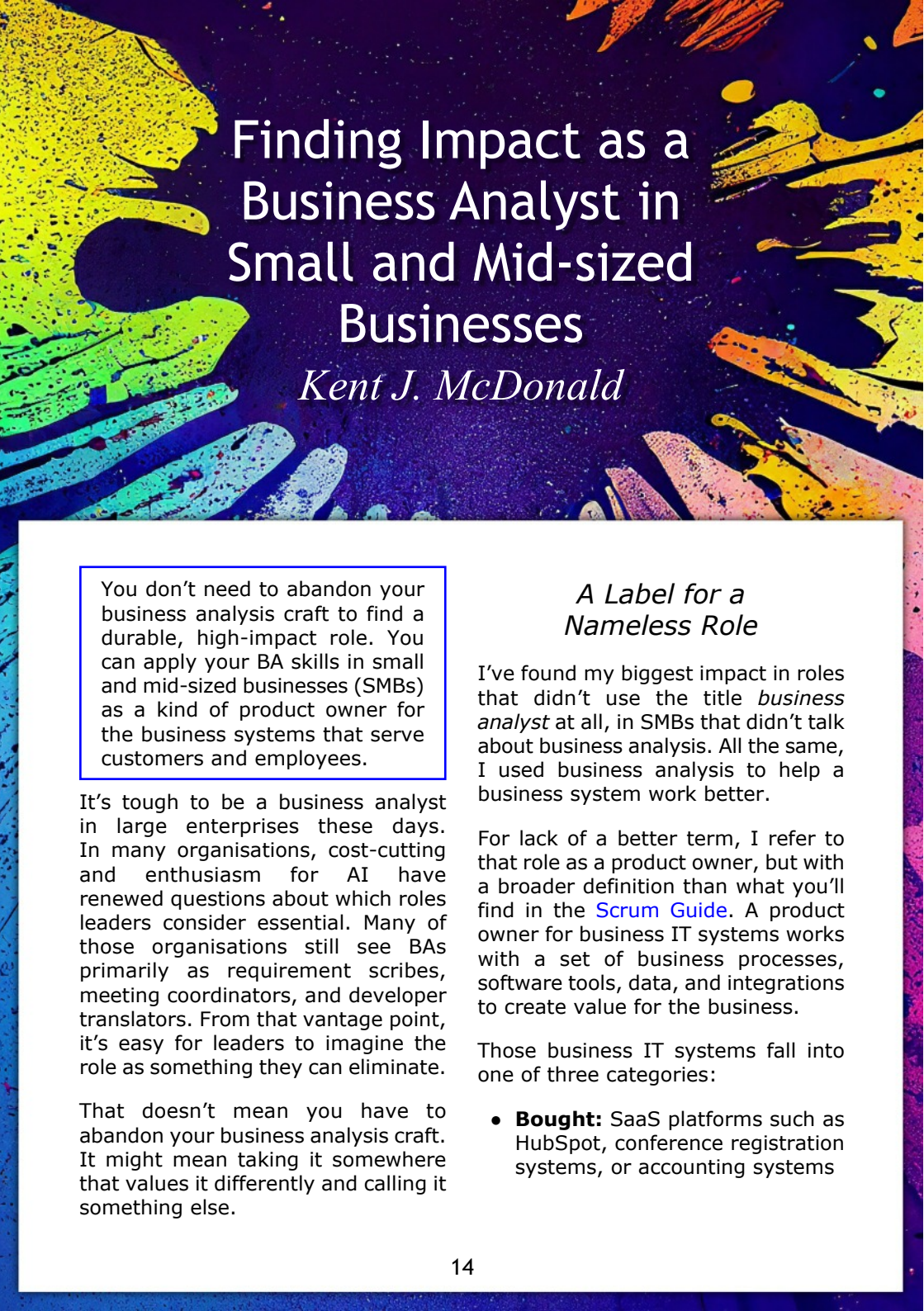
A standard like Mermaid enables more efficient analysis for the individual business analyst and better collaboration for the team. With generative AI, the barrier to reaping these benefits has been lowered.

Line Karkov is a senior business analyst with more than 15 years of experience. She is president of IIBA Denmark Chapter and writes on Medium about business analysis and product ownership. She lives in Copenhagen, Denmark and would be happy to connect via LinkedIn: [Line Karkov](#) | [LinkedIn](#)

Get BA Digest
in your mailbox
every quarter.
Subscribe at:

BAdigest.link





Finding Impact as a Business Analyst in Small and Mid-sized Businesses

Kent J. McDonald

You don't need to abandon your business analysis craft to find a durable, high-impact role. You can apply your BA skills in small and mid-sized businesses (SMBs) as a kind of product owner for the business systems that serve customers and employees.

It's tough to be a business analyst in large enterprises these days. In many organisations, cost-cutting and enthusiasm for AI have renewed questions about which roles leaders consider essential. Many of those organisations still see BAs primarily as requirement scribes, meeting coordinators, and developer translators. From that vantage point, it's easy for leaders to imagine the role as something they can eliminate.

That doesn't mean you have to abandon your business analysis craft. It might mean taking it somewhere that values it differently and calling it something else.

A Label for a Nameless Role

I've found my biggest impact in roles that didn't use the title *business analyst* at all, in SMBs that didn't talk about business analysis. All the same, I used business analysis to help a business system work better.

For lack of a better term, I refer to that role as a product owner, but with a broader definition than what you'll find in the [Scrum Guide](#). A product owner for business IT systems works with a set of business processes, software tools, data, and integrations to create value for the business.

Those business IT systems fall into one of three categories:

- **Bought:** SaaS platforms such as HubSpot, conference registration systems, or accounting systems

- **Built:** websites, conference session submission systems, price quoting tools, or integrations
- **Blended:** where custom tools fill important gaps around purchased platforms

Most product ownership advice assumes you're building software. In SMBs, just as much (or more) of the work is about configuring, integrating, and improving a mix of existing tools and a few custom pieces.

I Delivered Most Impact Where You'd Least Expect It

The two gigs where I felt the biggest impact weren't stints at corporate giants like General Motors, Principal Financial, or Wells Fargo. They were at a nonprofit and a dealership that sells warranties online.

The nonprofit was Agile Alliance. I was the product owner for their website, conference registration, membership, and conference session submissions. These were a mix of custom-built and SaaS systems, and my job was to ensure they enabled Agile Alliance's conference and membership operations.

I had a big impact in this role because I saw how changes to the submission system or website affected the conference experience and membership signups. I had a direct line of sight from my decisions to outcomes.

At the dealership, I worked on an internal product, even though nobody

inside the company called it that. The dealership has a website where customers provide information about their vehicle and desired coverage, get a price, and purchase a warranty. The website then provides that information to HubSpot (a CRM system), where dealership staff track warranty registrations.

None of those components is independently the product. The customer experience and operational outcome depend on the entire chain working together.

My work was no longer limited to website enhancements or HubSpot configuration changes. I managed changes to the business systems that enabled the workflow, enforced pricing rules, and tracked order data. Those are all familiar business analysis activities, but they directly supported the dealership's operations.

A product owner for business systems still uses the skills experienced BAs recognise. The difference is that the work is continuous and outcome-oriented. Rather than handing requirements to a project team and moving on, you stay connected to whether the system achieves the intended business outcomes.

Why SMBs Create the Opportunity

All SMBs use business systems of some sort. Because of their size, they rarely have an extensive IT department, which means they have to find someone to steward those systems. That stewardship involves more than administering them. It includes determining what you'll use

the system for and what changes you need to make to it.

That system steward might be a sales leader, operations manager, or revenue operations lead who inherited responsibility because someone had to do it. They know their business deeply but aren't familiar with what it takes to turn a collection of tools and workarounds into a cohesive and functioning system.

In other cases, founders or executive leaders never gave up the system stewardship responsibility. As their businesses grow, spreadsheets and informal handoffs stop being enough, or their time is better spent elsewhere. They need someone who can connect operational goals to investments in systems, not simply someone who can administer software.

Through my work with SMBs, I've seen this pattern multiple times. An organisation has essential systems without a clear owner, which provides an obvious opportunity for business analysts. You can step into the stewardship role directly, or partner with the existing system steward who needs the discipline you bring.

What Makes the Work Rewarding

In a smaller business, the distance between analysis and outcome is often much shorter than in an enterprise setting. When you change how the submission system supports reviews, you can see the impact on people who submit and review those sessions. When you change a few HubSpot settings, staff can spend

more time processing valid orders and less time resolving avoidable questions.

SMB work isn't easier or automatically better. It often involves ambiguity, limited budgets, and imperfect systems. But those conditions also make sound analysis and thoughtful prioritisation highly visible.

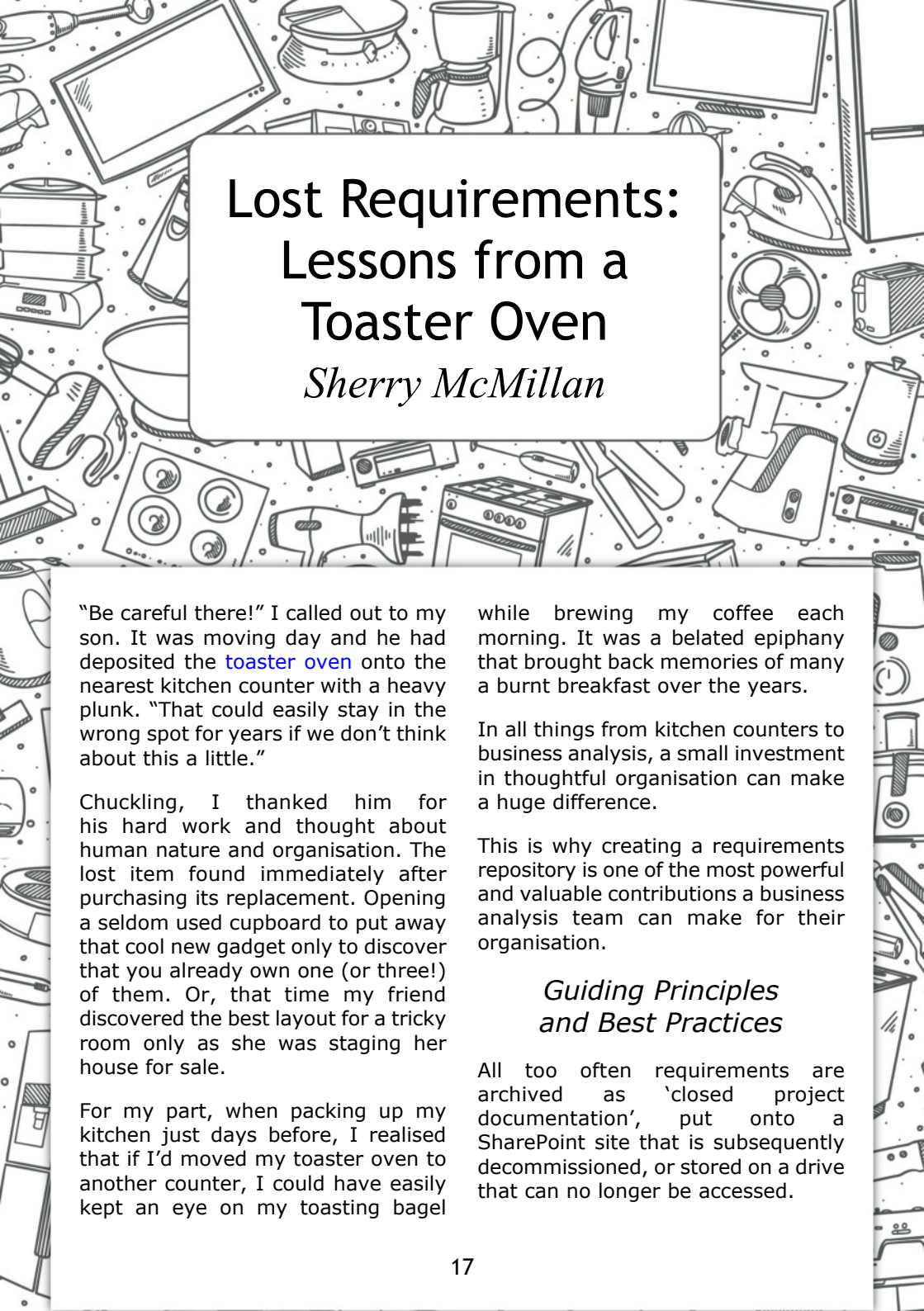
The systems you work on may not be innovative. But that doesn't matter as much when you can see the meaningful difference to the people using it and the customers relying on it.

An Opportunity If You're Willing to Explore It

I didn't set out to become a product owner for business systems. I started by helping support a website and a CRM implementation. Over time, I realised the impact I had on the business by stewarding its business systems.

Looking for a more direct connection between your work and business outcomes? Seek the places where an SMB depends on a messy collection of tools to serve customers. That's often where a product owner opportunity is hiding in plain sight, waiting for someone with your skills to step into it.

Kent J. McDonald helps small and mid-sized businesses design and own their business systems through InsideProduct. He has worked as a business analyst, product owner, and product manager across industries including financial services, health insurance, nonprofit, automotive, and performance marketing.



Lost Requirements: Lessons from a Toaster Oven

Sherry McMillan

"Be careful there!" I called out to my son. It was moving day and he had deposited the **toaster oven** onto the nearest kitchen counter with a heavy plunk. "That could easily stay in the wrong spot for years if we don't think about this a little."

Chuckling, I thanked him for his hard work and thought about human nature and organisation. The lost item found immediately after purchasing its replacement. Opening a seldom used cupboard to put away that cool new gadget only to discover that you already own one (or three!) of them. Or, that time my friend discovered the best layout for a tricky room only as she was staging her house for sale.

For my part, when packing up my kitchen just days before, I realised that if I'd moved my toaster oven to another counter, I could have easily kept an eye on my toasting bagel

while brewing my coffee each morning. It was a belated epiphany that brought back memories of many a burnt breakfast over the years.

In all things from kitchen counters to business analysis, a small investment in thoughtful organisation can make a huge difference.

This is why creating a requirements repository is one of the most powerful and valuable contributions a business analysis team can make for their organisation.

Guiding Principles and Best Practices

All too often requirements are archived as 'closed project documentation', put onto a SharePoint site that is subsequently decommissioned, or stored on a drive that can no longer be accessed.



Here are three **guiding principles** to help you advocate for, build, and maintain a requirements repository:

1. Requirements exist for the life of the product—not just for its implementation or the work done in any given project.
2. Store things so that you can find them. If it's worth documenting, it's worth storing those documents so that they can be found and used in the future. (And assume that you will need to find them in the future—because you, or someone in your organisation, almost certainly will.)
3. There will never be a perfect time to create a centralised document repository. But it doesn't take much to start building a system that is immediately valuable and will provide increasing value over time.

Here are some **best practices** and practical insights for building a requirements repository:

First, decide on a location to store all requirements for the organisation regardless of which BA writes them or which development team implements them. This storage location can be sophisticated or simple. Some examples: A Confluence space, secure shared drive, or SharePoint site for the BA practice. Note that requirements can be linked to other tools such as a Jira project and permissions can be fine-tuned for various stakeholder groups.

Next, organise your requirements by business need and feature set. Not

only does this make it easier to find requirements by navigation or search request, it helps identify any misses or duplication.

We typically look for information based on functionality. For example, we might ask "How is shipping calculated?", "What happens if the contract expires?", or "How does the customer request a duplicate statement?" We don't typically ask "What did we deliver in Release 562?" A good configuration includes the ability to find the requirement by project or release, but it doesn't make that the primary organisational structure.

With that in mind, organise and name the requirements to be consistent with the user interface and customer experience. For example, account management requirements might include "Add Account", "Modify Account", and "Archive Account". Even if the "Archive Account" feature was added years after Add and Modify Account, it logically belongs alongside other account functions.

Use tags, labels, and keywords to also identify requirements by product, project, owner, release, etc. Wherever possible, link requirements to other relevant artefacts such as architecture diagrams, Jira issues, support tickets, test cases, and release notes. This enables great searchability and provides a robust view of the product and its current state.

Benefits of a Requirements Repository

As the source of truth for requirements, the requirements



repository provides a myriad of benefits for the entire organisation.

It leads to better estimation and decision-making as all relevant requirements can be reviewed in context with related artefacts. History and traceability are baked in, as an audit trail is created when requirements are added or updated. This helps teams better understand why the system works the way it does.

It enables teams to respond quickly and factually to a wide variety of requests from providing information for an audit, building an RFP, or vetting an urgent fix request.

A requirements repository saves time and reduces mistakes by allowing BAs to modify existing requirements rather than writing new ones with each enhancement. It's a huge win when a BA can show existing requirements to a stakeholder group and ask "Is this still true or does it need to change?" It builds trust with your stakeholders and they'll appreciate not having to repeat themselves. Other teams will benefit too. For example, QA can easily find, revise, and/or re-run existing test cases linked to a modified requirement.

Using a centralised requirements repository will also reveal potential conflicts, constraints, and dependencies when multiple initiatives impact the same area. Issues can be identified and escalated quickly and easily.

Additional requirements repository benefits include improved BA reporting and workload management

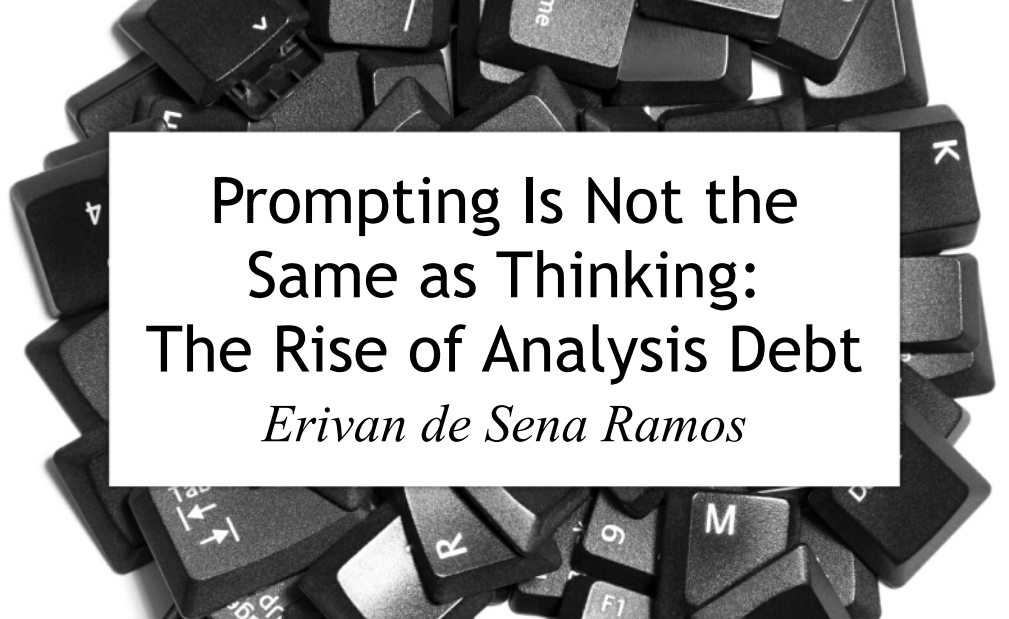
as requirement status can be tracked from draft to sign-off and beyond. The repository is a great resource for onboarding new staff. And it will naturally help the entire BA team write better and more consistent requirements. This is especially true when using templates for various requirement types.

Managing the repository can be identified as a Sprint 0 activity, as part of the Definition of Done (DoD), or a project hand-over task. Standing up the repository, if none exists, is a great stretch goal for a BA's annual performance review.

Don't forget to report back on benefits realised. You can highlight issues your team was able to efficiently triage, or even resolve, when the necessary information was found via the requirements repository. You can report on estimated hours saved by not duplicating effort. And you can capture positive feedback as the repository becomes a go-to resource for effective decision-making and problem-solving.

Perhaps you'd like to report on your success over a hot cup of coffee and a perfectly toasted bagel.

Sherry McMillan, CBAP has decades of experience helping organizations build and maintain their Information Architecture. She is also a best-selling, award-winning author who applies her BA skills to her creative endeavours. She provides 1:1 coaching to BAs and creative professionals helping them navigate complex initiatives and reach their goals.



Prompting Is Not the Same as Thinking: The Rise of Analysis Debt

Erivan de Sena Ramos

Artificial intelligence (AI) cannot compensate for poor analysis; it will only make it faster, cleaner and more convincing. That may sound uncomfortable when organisations are under pressure to adopt AI and accelerate delivery, but it is exactly why business analysis matters now more than ever.

Generative AI is already changing how BAs produce common artefacts. Requirements, user stories and acceptance criteria can now be drafted in seconds. [An experiment showed](#) that generative AI raised productivity and quality in professional writing and even made the work more enjoyable. That helps explain its strong attraction.

However, when producing documentation becomes easier than developing a shared understanding of the problem, organisations risk mistaking polished outputs for completed analysis. The danger is subtle because AI can refine weak

analysis, structure it and make it sound more mature than it is. Limited discovery is perceived as confident requirements, unresolved disagreement is hidden within a neat summary, and incomplete business rules become acceptance criteria that appear delivery-ready before the analysis has earned that confidence.

When AI Makes Weak Analysis Look Complete

That is why prompting is not the same as thinking. Prompting is becoming an important professional skill, and I do not dismiss it. A well-structured prompt can give AI clearer context, stronger boundaries and a more useful task. Yet the quality of a prompt still depends on the quality of the analysis behind it. If the BA has not understood the business problem, stakeholder context, evidence and decision to be supported, the prompt may simply produce a better-looking version of weak thinking.

There is also a gap between perceived and actual productivity. [One study observed](#) that tasks took longer with AI assistance, even though people believed AI had made them faster. Feeling efficient is not the same as producing a better outcome. This matters because [a survey on thinking skills](#) linked higher confidence in generative AI to less critical thinking. Our value has never been in accepting the first plausible answer, but in knowing when an answer needs to be questioned, tested or taken back to stakeholders.

A requirement is not valuable because it reads well; it is valuable because it reflects a genuine business need, validated stakeholder expectations, unambiguous business rules, feasible delivery conditions and trusted evidence. AI can assist with wording and structure, but it cannot take accountability for whether the organisation is solving the right problem.

The Rise of Analysis Debt

There is a professional risk emerging at a new pace. I call it '*analysis debt*': the future cost created when organisations make decisions based on incomplete, weak or insufficiently validated analysis, even when AI makes the outputs look complete.

This idea builds on an established family of debt concepts in software delivery. Technical debt, [as catalogued in the software engineering literature](#), describes future cost created by shortcuts that appear useful in the short term but create later maintenance burden. [Requirements engineering researchers have](#)

[defined requirements debt](#) as debt incurred during the identification, formalisation and implementation of requirements. Analysis debt sits earlier in the decision chain: it is not only about whether the requirement is weak, but whether the thinking, evidence and validation behind the decision are weak. Analysis debt was not born with AI, but AI has made it accumulate faster and matter more than ever.

I have seen this pattern in real delivery environments. Early discovery notes are quickly turned into requirements, and because the result looks structured and professionally written, the team believes the analysis is more mature than it is. The problem appears later, when developers ask about business rules nobody clarified, testers uncover exceptions nobody discussed, and users explain that the solution satisfies the documented requirement while missing the real business problem.

The issue was not poor documentation. The issue was incomplete analysis that became harder to challenge because AI made it look complete. Analysis debt rarely appears as a dramatic failure. It accumulates quietly through weak discovery, untested assumptions, missing stakeholder perspectives, implied business rules, shallow validation and solution-first thinking.

Why BAs Still Need to Think

There is also a cognitive dimension. AI reduces the effort of organising information, but that convenience can

encourage cognitive offloading, where the tool starts doing too much of the thinking on the user's behalf. [Recent research on cognitive offloading suggests](#) the benefit depends on whether people use AI as a thinking scaffold or simply delegate the work to it. AI may reduce the effort of producing artefacts, but it must not reduce our responsibility to question, validate and think.

None of this means avoiding AI. It is an excellent analysis partner for exploring perspectives, preparing workshops, challenging assumptions and identifying evidence gaps. The challenge is knowing where AI should assist and where judgement must remain human.

The [IIBA® 2026 Global State of Business Analysis Report](#) reinforces that communication, critical thinking, listening and stakeholder engagement remain central capabilities as AI absorbs more routine work. Our role is not simply to produce artefacts; it is to improve organisational understanding before decisions become expensive.

A Thinking Loop for Better Decision-making

In practice, I find it useful to treat AI-assisted analysis as a loop rather than a shortcut:

Frame → Generate or Challenge → Review → Validate → Decide.

Frame the problem and set boundaries before generating anything. Use AI to generate a draft, or to challenge what already exists. Review the output critically before it travels. Validate with stakeholders

and evidence. Only then decide what should influence delivery.

The order matters, and whatever guidelines or prompt techniques you use, the loop should always return to people before it reaches a decision. [A systematic review of human-in-the-loop AI](#) notes that various challenges still limit its effectiveness in practice, which is why the return to people must be designed rather than assumed.

Organisations are tempted to move straight from generation to decision because the output looks ready. Business analysis protects the space in between, where questioning, validation, judgement and stakeholder sense-making happen. Documentation will continue becoming cheaper, but understanding will remain expensive because it depends on context, evidence, trust and human judgement.

That is why prompting is not the same as thinking. In the age of AI, business analysis protects the space between AI-generated output and organisational decision-making: questioning what looks complete, validating what sounds convincing, and reducing analysis debt before weak thinking becomes an expensive reality.

Erivan de Sena Ramos is a BA Lead and Requirements Engineer with nearly 20 years of experience across Brazil and Australia. Certified in CBAP®, PMP®, CSM®, ITIL® and COBIT®, he shares insights on [LinkedIn](#) and [Medium](#). He authored [Analysis Debt: Why AI Still Needs Business Analysis](#) and [The BA Renaissance](#).



Speeding Up Decision-making: Transforming Elicitation in Business Analysis

Olabisi Odusanya

I recently attended a Financial Times training session, and one of the speakers clearly articulated something that resonated with my own experience of using AI at work. They described the shift of moving from "Can AI do this?" to "How do I redesign my processes and workflows since AI can do this?"

Before reaching that stage, though, I went through multiple trial-and-error cycles to get these tools to actually deliver what was so widely claimed online. What I discovered is that AI doesn't replace the business

analyst; it changes where the BA spends their time.

BA Effort

When it comes to analysis effort, elicitation is an area that takes up a large proportion of time. It's also the stage with the deepest human interaction. Whatever you decide to automate or streamline, it's essential to keep the "human in the loop", building your workflows around the parts AI can handle while you focus on the parts it can't.

A 2023 study by Dell'Acqua et al, published by the Institute for Operations Research and the Management Sciences, compared the outputs of 758 knowledge workers at Boston Consulting Group, with some being given access to AI tools and others not. The study found that those using AI completed 12.2% more tasks, 25.1% faster, but critically, those who used AI for tasks outside its effective range were 19% less likely to produce correct solutions. That finding captures the exact balance I've experienced: AI amplifies the BA's capability when applied to the right tasks, and undermines it when applied without judgement. Knowing the difference is the BA's job.

Here's how I've redesigned my elicitation workflow across three stages, using a recent project to build an onboarding tool for my team as a running example.

Interview and Feedback Analysis

In the past, this was a process where multiple meetings were held, clarification messages were sent back and forth, and a BA would try to capture as much information as possible. Alongside all of this, it was necessary to consider nuances, tone, and what people meant versus what they actually said. You'd walk out of a meeting with incomplete notes and spend hours trying to reconstruct what was discussed.

That's changed. Meeting summarisation tools now allow conversations to be recorded in their entirety, generate structured

summaries, and revisit them anytime. The interview doesn't disappear when the meeting ends; it becomes a permanent, searchable record.

For the onboarding tool, I collected feedback from the team on what they wanted to see, what helped them when they first joined, and what was missing. I fed all of this into AI, and it identified the most common themes across everyone's input. It grouped similar feedback items together, flagged edge cases, and made the collective voice more coherent. With little to no manual synthesis from me, I could see exactly where the team aligned and where they diverged.

Instead of spending days cross-referencing notes and manually identifying patterns, I focused my time on evaluating which themes actually mattered most, using my product knowledge to prioritise. Previously, significant time would be spent synthesising and organising the feedback items.

Document Analysis

Elicitation typically builds on what already exists: procedure documents, legacy process guides, previous project notes, etc. These documents are often long, outdated, and full of content that no longer applies. Traditionally, you'd read through everything yourself, mentally filter what's relevant, and try to build a picture over days.

AI is genuinely good at this. Pattern identification, streamlining documents, removing redundancy—this is where it shines. For the onboarding tool, I fed AI the existing procedure documents, previous

onboarding notes, and successful examples from other teams. It stripped out the fluff, outdated steps, duplicated content, irrelevant context, and surfaced the core procedural content that actually mattered.

I still had to validate what it kept and what it discarded. That's the BA's role. AI can clean and organise, but it doesn't know what's important. I do. The difference is I'm starting from a clean foundation in an hour instead of building one over days.

Mockup and Prototyping

This is arguably the most significant way AI has changed my elicitation workflow. Apps and solution prototypes can now be built in hours, a process that previously took days or weeks of wireframing, design reviews, and back-and-forth with developers.

For the onboarding tool, I shared the analysed documents, my prioritised notes, and the expected solution structure with AI. The AI tool I used produced a working mockup in two hours. I didn't stop there. I built a feedback mechanism directly into the prototype so the team could interact with it and tell me what worked and what didn't.

AI then helped me analyse that feedback the same way it handled the initial interviews: grouping common responses and identifying the most frequent points of friction. I made targeted iterations based on what the team actually said, not what I assumed. Within three days, the tool moved from an initial mockup to a product ready for use.

One thing I've found particularly useful is to use AI to create different versions of the same prototype for different audiences. A technical version for one group, a simplified version for another. Previously that meant doubling the design work. Now it's a prompt variation.

The Human Part

Business analysis, like other fields, is changing rapidly in the ways described above and many others too. However, elicitation is one area where human interaction and judgement remain the key differentiators. AI can summarise, synthesise, identify patterns, and generate prototypes, but it can't read the room. It can't tell you that a stakeholder said yes but meant no. It can't prioritise competing requirements based on organisational politics that it doesn't understand.

What redesigning my workflow with AI has done is free me to focus more on that human work, the quality of the questions I ask, the relationships I build, and the judgement calls I make, while AI handles the grunt work of organising, synthesising, and drafting.

The trade-off is not replacing the human part of the process. Rather, it is giving the human part more room to think.

Olabisi Odusanya is a Business Intelligence professional and the 2025 IIBA Ideas in Action global competition winner. She specialises in data-driven analytics across emerging and institutional markets.



Social Media Loafing: The Feedback We're Missing

Adrian Reed

When I was studying as an undergraduate, sociology and psychology fascinated me. Even though they were somewhat peripheral to my main area of study, the way that *people* behave (individually and in groups) has always interested me.

I remember coming across a term *social loafing* which describes the tendency for individuals to put in less effort (on average) if they are part of a group, compared with if they undertook the same task individually. I was reminded recently that there's a tangential and conceptual similarity between this and the *bystander effect*, where individuals are less likely to intervene in a situation if other people are present. For example, if someone collapses and only one person is present, that person will almost certainly call the emergency services. If there are ten people around, each person might assume "somebody else" has made the call.

I was thinking about these concepts recently having *very briefly* logged into Facebook (a social media network that, in my view, is about 15 years past its usefulness). I made the mistake of looking into a local neighbourhood group. I can almost *hear* the collective sigh of people reading this—if you've ever been part of a neighbourhood FB group you'll know what's coming...

While social media has the potential to bring people together and share information (and it does fulfil that, at least to an extent), the group I happened to briefly scroll through seemed to be largely filled with people "shaming" people who park in a particular way, and complaining about things like the refuse collectors missing their individual bin during the weekly collection.

What always fascinates me is how many people jump on to those threads, talking about the time their bin wasn't collected once seventeen

years ago, turning them into a metaphorical 'pity party', relentlessly reciting everything that's wrong with the city. **Yet nobody actually reports it.** Nobody makes the *actual* intervention that would solve the immediate problem.

The irony being that the city I live in actually has a reasonably easy to use mechanism for reporting things like missed bin collections. Which leads to the question, why aren't people using it?

Social Media Loafing

In my view, we seem to live in a world where sharing a meme, hitting share, or writing "hopes and prayers" on a social media update is a substitute for *actual* action. Perhaps people think that by contributing to a thread on a neighbourhood forum they are raising awareness of the issue... and they may be right. Certainly local elected members (councillors) are active on social media. However, personally, I'd rather they were focusing on the strategic policy-making issues at hand, rather than acting as a 'forwarding mailbox' for enquiries about refuse collection. In fairness, I fully understand *why* they do what they do... after all, visibility and popularity win at the ballot box. Long-term policy-making is less visible... but I digress.

The point here is that people may legitimately *think* they've taken action by posting something on social media and might, reasonably or unreasonably, expect that issue to be resolved. They might (rightly or wrongly) feel they've *done* what they need to do by posting on a Facebook group.

Unless we conclude that they are there just to have a whine (or, as we'd say in Pompey 'to have a good squinny'), then surely they are aiming to make a change? Surely they are *trying* to get their issue resolved? If so, what does this mean for those of us that design services and processes?

Build Services and Processes That Listen

Whether we're involved in designing an end-to-end service, or a process that contributes to a service, it's worth considering how that service or process (and the people who orchestrate it) *listens* to its environment. This involves *listening* to what people are saying about the service, or absence thereof, on all sorts of channels.

For example, one might argue that if people are complaining on Facebook about a missed bin collection, there are several possible hypotheses to consider:

- They might not know a 'report it' option is available. So is there a communication gap?
- Or, perhaps they've used it and it's too difficult. Is there a usability or inclusion issue?
- Maybe there are some people who are 'social media natives' and want a way to message on social media, rather than log in to a portal. Is that something that's technically and socially feasible? Perhaps it should be researched and considered.
- If there's a pattern of missed (individual bin) collections, then perhaps that's something that

needs investigating and addressing.

- Etc...

And yes, yes, there *is* also a possibility they're just having a squinny. But we can't assume *everyone* is a complete [dinny squin-bag](#), can we? And even if they *are* just having a moan, if the underlying complaint is valid, then it's an opportunity to look at how the service could be improved.

Feedback Is Everywhere

We often think of feedback as being something formal that's submitted via an official channel. Yet the reality is, if you look for it, feedback is *everywhere*. People using (or not using) a service is feedback. People

getting to the last page of a digital journey and abandoning is feedback. And yes, people complaining on social media rather than contacting your organisation *is* feedback. The key is to build processes that listen to it and can adapt accordingly.

So the question is: Does *your* organisation genuinely monitor for, and hear, feedback? And if not, whose voices and perspectives are being missed?

Adrian Reed helps organisations make sense of nonsense, one box and arrow at a time. He edits BA Digest and is Principal Consultant at Blackmetric. Curiosity fuels his thinking. Industrial amounts of caffeinated cola fuel the rest. Connect before the caffeine wears off: [linkedin.com/in/adrianreed](https://www.linkedin.com/in/adrianreed)

Mastering the Requirements Process

Join us for a two day immersive, interactive, online course

Facilitated by Adrian Reed

*13th - 14th October 2026
£995 + VAT (£1,194)*



FIND OUT MORE



What You Need to Know to Prepare Your Product Discovery Strategy

Nuno Santos



The way business analysts elicit requirements has evolved. While interviewing stakeholders, analysing documents and facilitating workshops remain essential techniques, there is now greater emphasis on discovering what will truly bring value to customers. However, discovery itself is built on assumptions that need to be tested and validated. In this article, I present a practical approach that brings together complementary techniques to support an effective discovery journey.

Throughout this article I refer to several well-known product discovery approaches from authors such as Teresa Torres, Itamar Gilad and Strategyzer. Rather than explaining each framework in depth, the focus is on showing how they can be combined into a practical discovery workflow that takes us from customer problems to validated product decisions.

Clarity in Terminology

A BA typically begins with an understanding of users' pain points, gained through conversations and other discovery activities. As you start shaping a potential solution, it's important to be clear about what you are trying to validate and what outcome you expect to achieve.

Before exploring validation techniques, it is worth clarifying some commonly confused concepts. Product development uses a range of artefacts that serve different purposes. A prototype explores design ideas and user interactions. A Proof of Concept (PoC) validates technical feasibility. A Minimum Viable Product (MVP) can be used to test whether a product or feature set creates sufficient customer value. This differs from a Minimum Marketable Feature (MMF) or Minimum Lovable Product (MLP), that

represent different stages of market readiness and user experience.

Discover Opportunities

Have you gone to the Gemba? In Lean thinking, Gemba refers to the place where the work happens and where value is created. For BAs, that often means observing customers in the context where they actually use a product or service. This is a fundamental part of developing empathy with customers, and I explored the topic in more detail in my [BA Digest 2022 Q2](#) article.

Interviews are also a vital part of product requirements elicitation. A business analyst can simply record the stakeholder's dictated "wish list", or they can engage more deeply to understand the underlying needs and problems that those requests are intended to address.

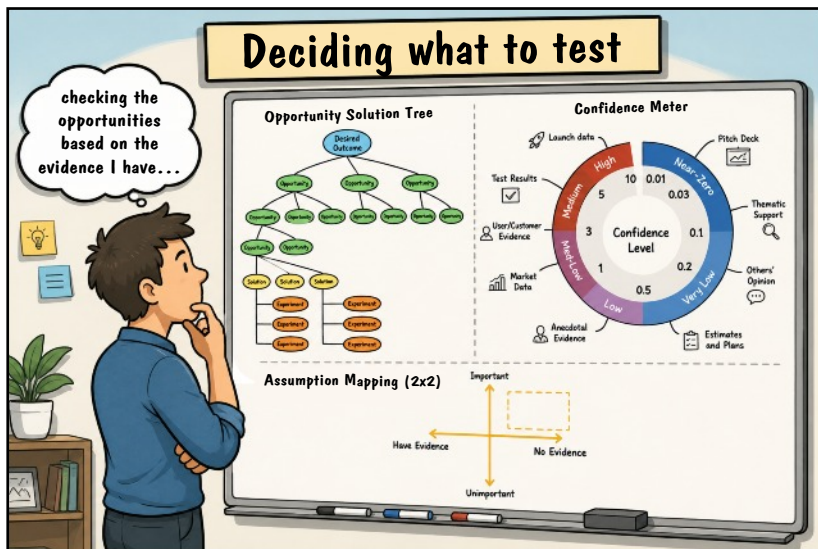
Techniques such as [continuous interviews](#) and [The Mom Test](#)

advocate a different approach to stakeholder interviews. Rather than asking customers what they want, the emphasis is on understanding their past experiences, in order to discover opportunities.

The insights gained from these interviews should allow us to identify opportunities worth exploring. In [Continuous discovery habits](#), Teresa Torres proposes [opportunity solution trees](#) as a way to decompose customer problems into opportunities, explore multiple solutions, and prioritise where to experiment first.

Decide What to Test

At this stage you may have identified dozens of customer problems and potential opportunities. The challenge is that not every opportunity deserves the same level of investment. Opportunity solution trees can help organise these opportunities, while Strategyzer's [assumption mapping](#)



business idea would fail if the assumption proved to be false.

Once the critical assumptions have been identified, discovery work needs can be organised into concrete experiments. The **GIST method** (Goals, Ideas, Steps, and Tasks), described by Itamar Gilad in his book [Evidence-Guided: Creating High-Impact Products in the Face of Uncertainty](#), provides a practical structure for prioritisation and planning. GIST extends this model by introducing a **confidence meter** that helps teams assess how much evidence actually supports each idea.

From the previous section, you should have identified the evidence supporting each hypothesis, so now the next logical step is to validate each of them. The most appropriate validation technique will depend on the level of evidence already available. To support this decision,



you can use the AFTER framework and test cards.

The **AFTER framework** (Assessment, Fact-finding, Tests, Experiments, Release) organises 28 validation methods according to both their cost and the strength of evidence they generate. Teams should typically use the least costly method capable of reducing uncertainty before investing in more expensive experiments.

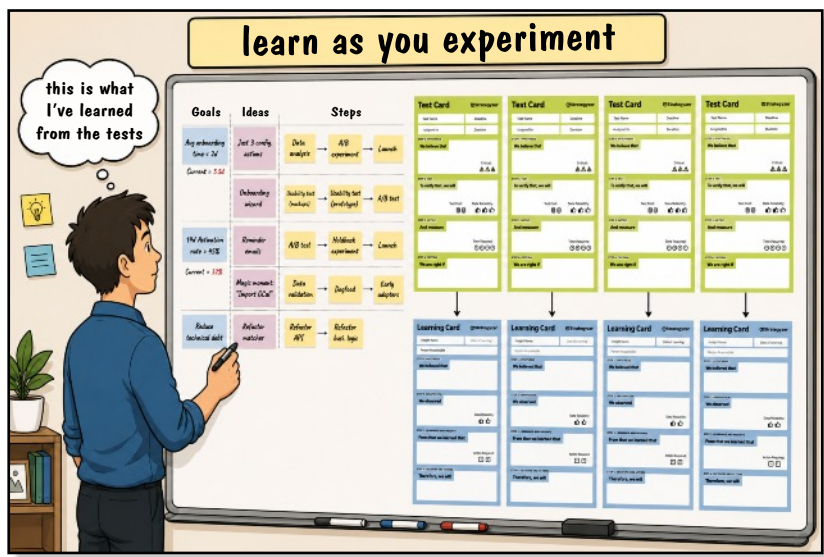
Once the discovery work has been prioritised, the testing of each hypothesis can be planned utilising **test cards**. A test card captures the hypothesis being tested, the experiment to perform, the success metric and the criteria for deciding whether the hypothesis has been validated or rejected.

BAs can contribute throughout this process, both in terms of helping to determine appropriate techniques to use, and also in setting up relevant tests.

Learn as You Experiment

The next step is to learn from the results of your experiments. Just as agile development teams deliver increments and gather feedback, product discovery teams gather feedback from experiments and tests. They can use a GIST board to follow them, and **learning cards** for the test results. The GIST Board is a visual tool for organising and tracking the progress of the discovery. It's used to map out and manage the different phases of the GIST framework.

The GIST Board is a visual management tool that connects Goals, Ideas, Steps and Tasks, providing traceability from strategic objectives down to individual discovery activities. Teams use it to coordinate experiments, discuss findings and continuously refine their roadmap. Although these conversations can happen at any time, regular discussions during daily



stand-ups or other agile events help keep discovery work visible and aligned.

Every completed experiment should produce explicit learning, regardless of whether the hypothesis is validated or disproved. Learning cards capture that evidence, document what was observed and define the next action. This prevents teams from unnecessarily repeating experiments and gradually increases confidence in product decisions. The resulting insights feed back into the assumption map and increase (or decrease) the confidence score.

By this stage, each important hypothesis should fall into one of two

categories: either there is sufficient evidence to proceed with delivery, or further discovery is required. Product discovery is therefore not about proving ideas right, but about reducing uncertainty before investing in implementation.

The images in this article were generated by AI, prompted by the author.

Nuno Santos is currently a Senior Business Analyst at Natixis Portugal, ambassador at the IREB Portuguese chapter and a volunteer at the IIBA® Brazil chapter. Reach out at LinkedIn (www.linkedin.com/in/nunoasantos) and Medium (medium.com/@nunosantos.vv).

SHOW THE WORLD YOU MEAN BUSINESS

CBAP®

Certified Business Analysis Professional



Expected Chaos Is Not a Launch Strategy: The Role of Business Architecture in Smoothing the Transition

Li Yang

As business analysts, we often work on projects involving system changes, while others require only process changes. The most complex can involve changes to systems, processes, facilities, organisation structure, and external partners. The nature and size of a project, regulatory requirements, project management and change management practices, and stakeholder expectations can all influence the level of planning and coordination required for a smooth

project launch. On a recent project that I worked on, despite a large amount of effort by many dedicated practitioners, a senior manager predicted chaos right before the project launch. Surprised by this, I wondered whether such a low expectation was justified.

During a project launch, issues often arise when accountabilities are unclear or when assumptions are proved to be incorrect. Problems can also occur when cross-functional

processes or hand-offs are not fully fleshed out, or when integration or system gaps are not identified. The list could go on. Accepting chaos as inevitable may seem pragmatic, but it can erode accountability, mask preventable risks, and undermine stakeholder confidence before value delivery even begins. Treating chaos as the default does not make organisations resilient. It risks making them complacent.

While it's impossible to anticipate every event, leveraging business architecture practice early in the project lifecycle can show how different aspects of the operations connect and make the impact of project changes visible.

Business architecture describes what an organisation does or tries to do. IIBA®'s [Business Analysis Body of Knowledge \(BABOK®\)](#) guide states: "Business architecture models the enterprise in order to show how strategic concerns of key stakeholders are met and to support ongoing business transformation efforts". The scope of business architecture is the entire business. By separating different concerns, as the BABOK® states, business architecture "...provide[s] an understanding of interactions, integration opportunities, redundancies, and inconsistencies". Such understanding would be key to any organisation before introducing changes.

BAs Utilising Business Architecture Principles

A business analyst can apply business architecture principles on any project by intentionally analysing the

organisational context and defining core business concepts early on.

A BA should define the context within which an organisation exists from an outside-in perspective. In [Business Architecture: Collecting, Connecting and Correcting the Dots](#), Roger Burlton clarified that "The purpose of any business (or any organisation) is to deliver a right mix of outcomes of value for all of the main stakeholders of the business". Without examining the inner workings of the organisation in detail, a BA can quickly create a one-page context diagram that focuses on the most essential value exchanges between the organisation and its external stakeholders. In other words, what matters most from the external stakeholders' perspective, such as:

- Products and services delivered to the customers.
- Information requests, orders, and payments collected from the customers.
- Materials or services received from the suppliers.
- Orders and payments sent to the suppliers.
- Regulations imposed by the regulatory bodies.
- Compliance evidence delivered to the regulatory bodies.
- Funding and future directions from the owners or funders.
- Performance reports and likely profits to the owners or funders.

Using the draft context diagram, a BA can work with the project sponsor and the project manager to identify any missing external stakeholders or exchanges. They can also assess the quality of those exchanges and the experiences associated with delivering or receiving them.

Missing exchanges, poor-quality exchanges or suboptimal experiences are often among the reasons why a project exists in the first place. Once confirmed, these problem areas can be marked in red on the context diagram. In this way, a deceptively simple diagram can provide a starting point for visual traceability and become an effective communication tool.

Project Context & Concept Models

Once the context of the organisation is confirmed, a BA may consider creating a context diagram for the project or product in focus. This can help clarify how the proposed changes connect with the external stakeholders, including which elements of the value exchanges will and will not be affected. The project context diagram can therefore provide a powerful visual representation of the project scope.

Additionally, it is useful for BAs to create business concept models to create shared understanding and improve business communication. In [Business Knowledge Blueprints: Enabling Your Data to Speak the Language of the Business](#), Ronald G. Ross defines a concept model as “a set of concepts structured according to the relations among them”.

A good starting point to create the concept model is to consider the nouns used in the context diagram, such as “customer”, “product”, “service”. Each is likely to represent a concept that is important to the organisation. Any significant project is likely to affect at least some of these core concepts.

The value of this effort is threefold. First, while most organisations have glossaries, their definitions do not always capture the essential meaning of a term. Some instead describe its purpose, function or use. The [Merriam-Webster Unabridged Dictionary](#) defines *definition* as “a word or phrase expressing the essential nature of a thing or class of things: an answer to the question “what is x?” Or “what is an x?”. By applying this rigour to the definition of core business concepts, a BA can help reduce ambiguity and improve clarity. The earlier this clarity is achieved, the better.

Secondly, defining core concepts often surfaces a common problem faced by many organisations: fundamental concepts, such as “customer”, may have many definitions. Where a single definition is essential, as might be the case for a “customer 360” project, this exercise alone can help reveal the scope of work required to resolve those differences holistically.

Where the project does not have the mandate to resolve these differences, making them visible still allows the project team to consider the implications. These might include additional considerations around system integration, process change, or post-launch reporting and

measurement. Understanding these implications in advance is far better than discovering them through surprises and disruption after launch.

Finally, analysing the relationships between concepts or groups of concepts provides an opportunity to understand how they affect internal stakeholders and functional areas. Creating and validating concept models can be a great way to engage with stakeholders.

If a project involves an off-the-shelf system, concept models and the associated definitions can help both the project team and system vendors to assess how well the organisation's needs align with what the system offers.

Conclusion

Business architecture does not remove every uncertainty from a project, but it can help BAs make important connections, gaps and impacts visible earlier. By applying these principles from the outset, BAs can help replace avoidable launch chaos with greater clarity and preparedness.

Li Yang, CBAP, a top-notch business analyst and aspiring business architect, instructor, and speaker, brings her multi-industry experience and passion of delivering value to every engagement. You can reach Li at www.linkedin.com/in/li-yang-cbap-3733a21/

CALLING ALL CHANGE AGENTS AND CORPORATE MAVERICKS

The Agilist is a nonprofit print-only publication supporting individuals and organisations to respond to a rapidly changing world.

For a few dollars, euros or pounds a month, receive a monthly digest of thought-leadership covering:

Strategic Alignment and Execution, Innovation, Organisational Change, Org Design, Leadership, Coaching, Management, Personal Change, Collaboration, Decision-Making, Product Development, Psychology, Customer Centricity, Academic Research in the field of Business, Market Competitiveness, Team Ways of Working.

Subscribe today at theagilist.org



Sailing by the Stars

Nick de Voil

I'm not much of an astronomer, but when I read about this summer's Perseid meteor shower it sounded exciting, so I went to a nearby park at midnight and watched the shooting stars. It got me thinking about how, for thousands of years, seafarers have used the stars to navigate. And about how we often talk about 'orientation' figuratively to refer to getting our bearings in an environment. The word originates from the idea of using the sunrise to know which direction is East.

In [article 9 of this series](#) we broached a question that's often crucial for business analysts—when we're dealing with a group, how do we know that everyone in the group

understands something the same way? Of course, one possible answer to this is: if people say clearly and unambiguously what they mean, there will be no problem. It's a naïve answer which ignores some basics of humans and language:

- It's impossible to communicate without assuming some degree of shared context;
- Language is inherently ambiguous. It's a feature, not a bug. As the logician Bertrand Russell argued, "it would be altogether incredibly inconvenient to have an unambiguous language, and therefore mercifully we have not got one."

So how are we to do it? The answer is that we can use the sailor's methods: orientation, triangulation and constellation. What's different is that, instead of stars, we're talking about constructs.

Orientation

In its literal meaning, orientation is about using one point—the sun—as a reference from which we infer everything else we need to know. If we were to take an analogous approach to navigating construct systems, however, we would probably get lost. For example, we might think, "I know what this stakeholder means by 'successful project' because I worked with her on a successful project last year and it was characterised by swift completion and adherence to budgetary constraints"—but actually we're failing to recognise that on this occasion she's addressing herself to the IT project leader and is making a coded point about delivering on the business requirements.

Unfortunately, it's very common for people to use this approach. We all do it. By latching on to just one aspect of a situation, or one characteristic of a person, and drawing conclusions based only on that, we can go seriously astray.

Triangulation

The general idea of triangulation is that you use more than one data item—typically two—to deduce another one. For example, a sailor would once have used a [sextant](#) to measure the angle between the Pole Star (item 1) and the horizon (item 2) in order to work out their latitude

(item 3). Joining these three points creates a triangle, a stable two-dimensional shape.

Applying triangulation to a construct system means taking into account more than one of the constructs, and their relationship to one another. This gives us greater confidence in believing a meeting of minds has taken place. But that confidence can still be misplaced. We might say, "This stakeholder has used the word 'actor' and also mentioned the 'attributes' of a customer, therefore I'm on safe ground assuming that they understand UML modelling" without recognising that those two terms are not exclusive to UML or even systems modelling in general, and in fact this stakeholder knows nothing whatever about UML.

Constellation

A rookie stargazer like myself could easily go wrong using an orientation or triangulation approach based on the North Star, because I'm not even confident I can locate it in the night sky. Fortunately, though, it's not necessary to think of the sky as simply a random collection of points. Many generations of people have looked at the same spectacle and, using the amazing human propensity to deduce meaning even where the facts don't justify it, have seen constellations—groups of stars that appear to form *patterns*. The patterns are named after animals, objects or characters which they seem to resemble. While an individual star is just an isolated point, patterns are easy to recognise. For example, probably the best-known constellation is the Plough (sometimes called the Big Dipper).

It's easy to find the Plough in the sky. And it just so happens that, by following the imaginary line which connects two specific stars in the Plough, you can find the North Star.

Construct systems work the same way when we're communicating with someone. The more of the other person's constructs we're aware of, and the more we can see the patterns that they form, the more confident we are of understanding what they're trying to say, and the more readily we're able to grasp new constructs which they offer, by seeing their relationships to the previously understood constructs. Again, this process isn't entirely free of risk: when we think we see a pattern, we might be jumping to a conclusion that isn't really justified, but our high degree of confidence blinds us to the possibility.

Working through a construct map—even a fragmentary one—with stakeholders is a hugely impactful way of ensuring shared understanding. The same is true of concept maps and data models. A conversation about as few as three constructs, where you discuss their definition, their meaning, and their interrelationships, always identifies areas of agreement and relative certainty, as well as areas of conflict and doubt.

Social Constructs and Brute Facts

Unlike stars, most constructs are abstract concepts which refer to something that isn't directly visible or part of the natural world. An example of this is anything to do with money.

Money only exists because humans have agreed that it exists. The philosopher John Searle referred to this as an *institutional fact* and contrasted these with *brute facts*. A brute fact is something indisputably perceptible, like a star for example.

Brute facts, such as documents or equipment, are immensely helpful for a business analyst because they provide a stable reference point. We can include them in our construct maps in the knowledge that everyone's perception of them will be very much the same. This is one reason why externalising thought processes by creating visual models—such as construct maps—or prototypes is so valuable when seeking shared understanding amongst stakeholders. Having said that, we need to recognise that, although everyone will see the same *external form* when they look at something, the cognitive or emotional *meaning* ascribed to it by different individuals may vary quite widely.

Nick helps organisations create people-oriented systems, products and services. He's trained thousands of professionals in business analysis, user experience, facilitation, systems design, project management and agile development. Nick is author of "User Experience Foundations" and is working on a book about GCA.

Contact Nick via LinkedIn
www.linkedin.com/in/nickdevoil/
or BlueSky [@nickdevoil.bsky.social](https://bsky.app/profile/nickdevoil.bsky.social)

You can read the previous articles
in this series by accessing
[the BA Digest archive](#)

Why Strong BAs Say No More Than They Say Yes

Anurita Ruscoe



"Can We Just Add One More Thing?"

If you've worked in delivery, you've almost certainly heard that phrase.

One extra feature. One stakeholder

request. Individually they seem harmless. Multiply that conversation across the entire delivery and, before you know it, the backlog has taken root, grown branches, and become an entire woodland of "must-have" features. The result? Delivery slows,

priorities blur, and teams spend months building features that add very little real value.

Strong BAs understand something that many teams overlook: every “yes” has a cost. Every additional feature consumes time, budget, and delivery capacity. That's why strong BAs often help teams say “no” more readily, rather than automatically saying “yes”. Not to block progress, but to protect it.

The Hidden Skill of Prioritisation Discipline

Many people think prioritisation starts once the backlog has been written. It does not!

This is where prioritisation becomes a discipline rather than an exercise. It's the discipline of asking difficult questions before work enters delivery, resisting the temptation to say yes to every request, and protecting delivery capacity for the initiatives that create the greatest value. Strong BAs understand that prioritisation isn't about ranking everything, it's about deciding what deserves to be built in the first place.

Every request should earn its place in the backlog on the strength of the problem it addresses and the value it is expected to create. It should not simply be added because someone asked for it.

Before agreeing to new work, ask questions including:

- What problem are we trying to solve?
- What value will this deliver?

- Is this essential for the Minimum Viable Product (MVP)?
- What happens if we don't do it?
- How will we know it was successful?

If the answers aren't clear, neither is the priority.

Protecting Teams from Low-value Work

One of the most valuable things a BA can do is protect delivery teams from solving the wrong problems. Every low-value feature consumes time, budget, and attention that could have been invested elsewhere. Great BAs recognise that protecting delivery isn't just about removing blockers. It's about stopping unnecessary work from reaching the team in the first place.

Sometimes a challenge uncovers a simpler solution. Sometimes it reveals a process improvement rather than a technical one. Occasionally, it reveals the problem doesn't need solving at all. Every unnecessary feature removed creates space for something that genuinely matters.

Challenging Stakeholders Constructively

Saying “no” in any relationship, professional or personal, can create friction. The truth is, it's rarely the word “**no**” that damages the relationship. It's the lack of explanation that follows.

One sounds harsh:

“No, we won't build that.”

The other sounds collaborative:

"No, I don't think we should build that because it falls outside our agreed scope and prioritising it would delay higher-value work. Can we first explore the problem we're trying to solve?"

The decision is exactly the same. The difference is the explanation. One closes the conversation. The other fosters understanding.

In my experience, most stakeholders welcome constructive challenge when they understand it's intended to improve the outcome, not block progress. Ultimately, we're all in the same boat, trying to reach the same destination. Sometimes the BA's role is simply to make sure we don't drift off course.

Saying No without Damaging Relationships

Saying "no" doesn't have to mean rejecting an idea forever.

Often it simply means **"not yet."**

Not every idea belongs in the current release, but that doesn't mean it has no value. A well-managed parking lot gives future enhancements a home without distracting from today's priorities.

Questions like these help move the conversation from emotion to evidence:

- What value are we expecting?
- What won't we be able to deliver if we prioritise this?

- Could we prove the assumption with a smaller MVP first?

Most stakeholders are far more comfortable hearing "not now" when they understand why. Good prioritisation isn't about shutting ideas down. It's about making deliberate choices that keep delivery focused on the work that matters most.

The Analyst's Edge

Business analysts don't need authority to build influence.

Influence is built through judgement, knowing when to challenge, when to simplify, and when to protect delivery from work that doesn't align with the agreed objectives.

Great analysis isn't measured by the number of requirements written. It's measured by the quality of the decisions that shape delivery. Because every "yes" has a cost. Strong business analysts understand that saying "yes" to one idea often means saying "no" to another. That's why every decision deserves discipline.

Anurita Ruscoe was a finalist for the BA of the Year Award 2025 and Founder of The Business Analyst Confidence Studio, which empowers BAs to lead with confidence, influence, and purpose. She leads a team of BAs and is passionate about redefining the profession in an AI-driven world.

LinkedIn:

www.linkedin.com/in/anuritaruscoe



Blueshift Part 2: How Business Analysis Can Stop a Cyberattack

Mark Cross

Imagine that you leave the office one day with your head full of user stories and process diagrams but the next day, everything is different. You still have the same work to do but no one can log in. You might learn that the HR systems you are working with are

down, and there is no indication yet of whether any sensitive information has been stolen, or even whether service can be restored in time to run payroll. In the worst case, you may find that all eyes are on *you* to

provide answers that no one else can give.

No one wants to find themselves in this position, but as cyberattacks become more common it is no longer something we can ignore. There is a significant chance that someone reading this article today will find their organisation involved in a cyber incident within the next few years, and when that happens the outcome will probably have been determined weeks or months earlier, often in the little choices we make every day.

This is the third article in a series, following *Redshift: The Art of Asking the Right Security Questions* in the [Q2 2026 edition](#), and *Blueshift: How business analysis can help ensure that transformation is delivered securely* in the [Q3 2026 edition](#).

Introducing the Cyber Kill Chain

Over the last few years I have studied dozens of incidents. I've reviewed testimony shared by the executives of Marks & Spencer and Co-op Group to the UK Parliament, evidence provided by expert witnesses from Transport for London in the prosecution of the two criminals who compromised their systems last year, and the perspective of the CISO of SolarWinds at the AISA Cybersecurity conference. The details of what happens inside incidents are rarely made public, so cases such as these provide extremely important insights.

These cases illustrate a common pattern. The attackers began with reconnaissance to identify weaknesses in their target's defences, then took time to craft

malware or acquire stolen credentials to use in the attack. The attack itself has three further stages. It begins with the first successful compromise of the defender's systems. The attackers then consolidate that access, moving laterally around the target's systems, elevating their level of access, and installing ways to regain entry if they get locked out. The final step is to achieve their objectives, which can range from exfiltrating sensitive information to disabling critical systems and holding them to ransom. Collectively these steps are called the cyber kill chain.

In [my last article](#) I described how business analysis is involved in the first three functions of the [NIST cybersecurity framework](#). Today I am exploring the remaining three: Detect, Respond, and Recover. These sit on the defensive side of the attack and map loosely onto the second half of the kill chain. Defenders must detect the initial compromise, respond faster than the attackers can consolidate their position, and begin recovery before those objectives are achieved.

So, where does business analysis fit in?

Detect

Detection is a battlefield. Business analysts are rarely asked to monitor for intrusions, but the people in Security Operations (SecOps) rely on information that we produce. Response cannot begin before detection, and every year attackers are learning to break out of their initial foothold faster. According to the cybersecurity vendor CrowdStrike, [average breakout times](#)

in 2021 were 98 minutes. By 2025 that had fallen to 29 minutes. The metric that matters most here is Mean Time To Detect (MTTD), a non-functional requirement that should sit well ahead of those breakout times if the potential damage is to be contained.

SecOps need two things from us.

Firstly, they need real-time information from the systems we are developing. This usually appears as a functional requirement for logs and notifications from your applications and services to be fed into a Security Information and Event Management (SIEM) tool. Without that integration, detection is slow and inconsistent. Secondly, they need reliable information about the organisation's assets so they can identify abnormal behaviour and assess risk. That means sensitive information such as Personally Identifiable Information (PII) and trade secrets is properly classified, and that alarm profiles are given the right priority. An under-prioritised alarm may be ignored. An over-prioritised one can flood SecOps with hundreds of false positives that distract from genuine threats and lead to a kind of short-term burnout called alert fatigue.

Respond

Response was once a matter of the network perimeter. If an intrusion was detected, you could remove the malware or reconfigure a firewall and the network was secure again. However, our users and our assets are now so widely distributed that an attacker who takes control of a privileged account or service can be extremely difficult to find. The key

metric here is Mean Time To Respond (MTTR). If we can keep the sum of MTTD and MTTR below the breakout time, it's likely we can stop an attack before any significant harm is done, though that becomes harder to achieve as breakout times fall.

Business analysts cannot easily influence how quickly their cybersecurity teams respond. What we can influence is how quickly and easily the attack spreads. If we follow 'Secure by Design' principles and capture requirements that apply secure configuration baselines, internal systems become more resistant to attack. If we ensure that users and services are given only the precise level of access they need to perform their jobs, we limit the harm that can be done when a single account is compromised. We can also learn to apply [zero trust principles](#), which combine compartmentation of the network with least privilege, narrowing the attacker's options for moving through it and drawing attention to their attempts to exceed the permissions they have been given.

Recover

The most important metric of all is the Recovery Time Objective (RTO), the time available to restore all systems to operation following an incident. Recovery from a major incident can take weeks or months. While systems are down, customers are impacted, sales are lost, and your reputation is on fire.

The questions we need to ask here are about which data needs to be protected by immutable backups, what disaster recovery capabilities you need, and, critically, how the

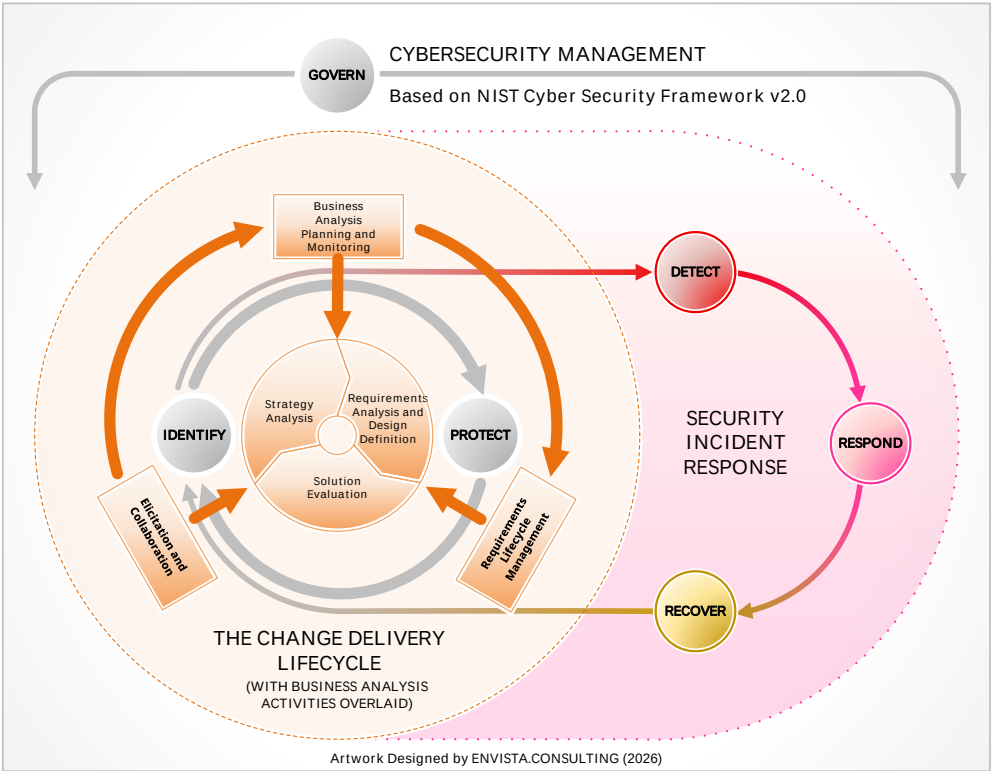
processes for both can be thoroughly tested so that they perform as expected in a crisis. If ransomware is involved, the attackers will be actively trying to sabotage those capabilities by deleting backups and revoking permissions from legitimate users.

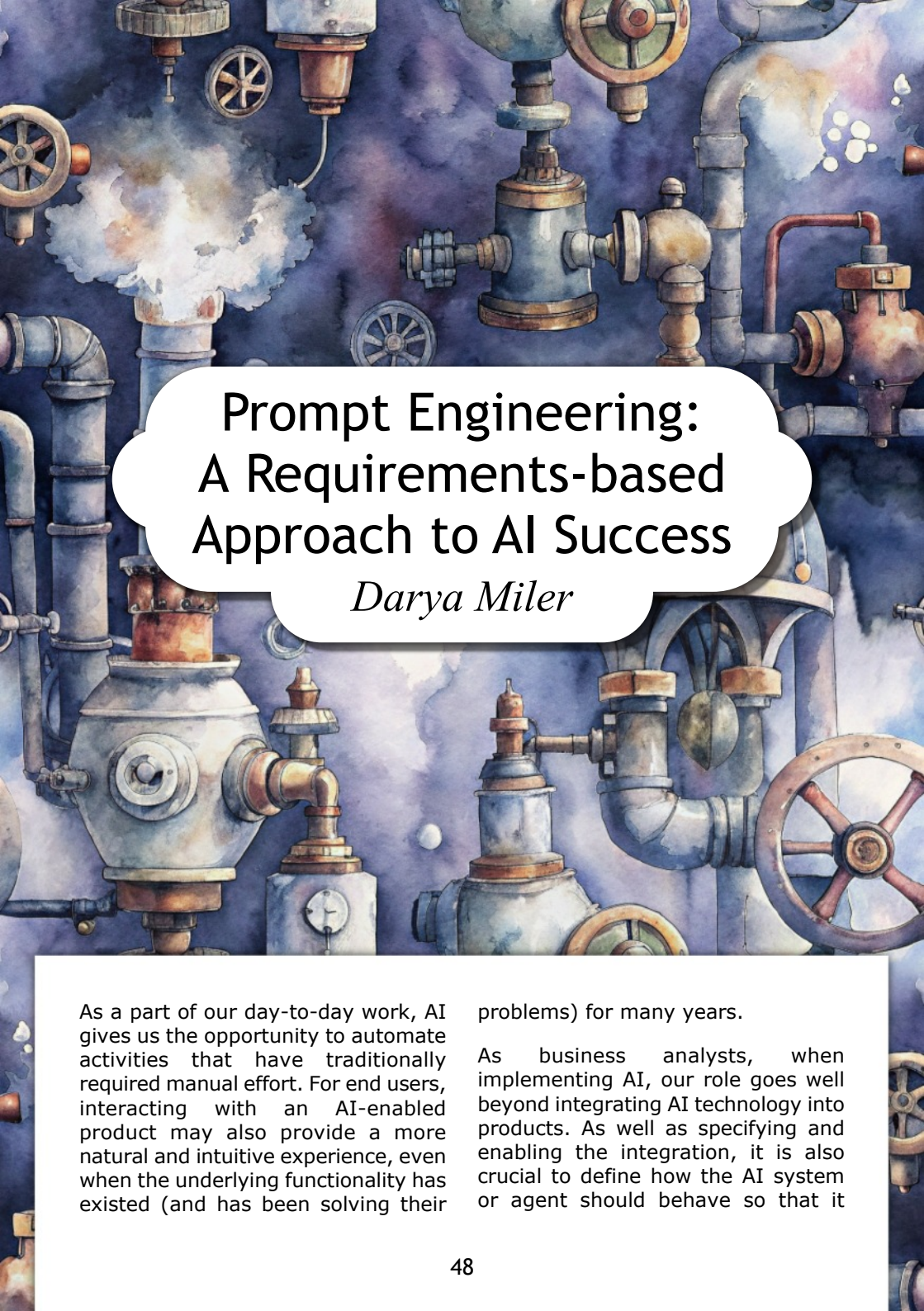
Conclusion

It isn't our job to replace cybersecurity teams, but those teams cannot do their jobs effectively without our skills and product knowledge. By applying a critical mindset to transformation, business analysts can identify risks, help to get stakeholders aligned, and shape

robust controls for new capabilities through the project lifecycle. My own view is that the case for business analysts to be fluent in cybersecurity has never been stronger or more urgent. If you have questions, or experiences of your own to share, please do reach out. I would be glad to hear from you.

Mark Cross is a cybersecurity consultant, business analyst, and speaker. He is the founder of Envista Consulting and chairs the IIBA's Cyber Special Interest Group. Connect with him on LinkedIn at www.linkedin.com/in/markcross.





Prompt Engineering: A Requirements-based Approach to AI Success

Darya Miler

As a part of our day-to-day work, AI gives us the opportunity to automate activities that have traditionally required manual effort. For end users, interacting with an AI-enabled product may also provide a more natural and intuitive experience, even when the underlying functionality has existed (and has been solving their

problems) for many years.

As business analysts, when implementing AI, our role goes well beyond integrating AI technology into products. As well as specifying and enabling the integration, it is also crucial to define how the AI system or agent should behave so that it

maximises value for the end user. In this context, a production prompt, that is, the prompt embedded within a live AI-enabled product, can be viewed as an executable form of product requirements. It translates business intent, user needs, operational constraints, and quality expectations into instructions an AI system can follow consistently.

Requirement Representation inside the Prompt

Requirements, user stories, business rules, and other requirements artefacts can inform or be converted into elements of a prompt which will subsequently be used within the product. However, this relies on requirements and stories being well-written and complete. For example, a requirement which states simply “summarise customer feedback” is incomplete because it does not define the audience, decision to be supported, source material, level of detail, etc.

To avoid incompleteness like this, the outcome that is sought through using AI for each specific task should be considered. Once the outcome is clear, prompt components such as the following should be outlined:

- **Role and Context:** Define the AI’s role, audience, purpose, and product environment. (Business and stakeholder requirements)
- **Task and Scope:** Specify the required action and clearly define what is excluded. (Business, functional, and scope requirements)
- **Inputs:** Describe input data, format, sources, and possible missing information. (Data, interface, and integration requirements)
- **Output Contract:** Define the required structure, format, length, tone, and language. (Functional and NFRs)
- **Decision Rules:** Explain prioritisation, classification, thresholds, and conflict-resolution logic. (Business rules and functional requirements)
- **Boundaries:** State prohibited actions, assumptions, safety limits, and escalation requirements. (Security, compliance, safety, and scope constraints)
- **Evaluation Tests:** Define acceptance criteria and tests for quality and consistency. (Acceptance criteria and NFRs)
- **Edge-Case Instructions:** Explain handling of incomplete, conflicting, unusual, or invalid inputs. (Business requirements and NFRs)

This decomposition supports requirements traceability and structured explanation of what the user expects as an output.

Tips and Edge Cases (How to Avoid Unexpected Outputs)

Once the prompt is decomposed, the BA can validate its content, refine the wording, and apply the following tips.

To avoid unexpected results you can use clear action verbs to avoid vague instructions such as 'help user'. You can use verbs such as write/compose/generate if the prompt should generate or create something; evaluate/compare/classify for analytics; translate/summarise/simplify for making text understandable and concise, and so forth.

The context should act as a single source of truth that can be updated without rewriting the entire prompt. It should contain only the information needed by the AI, avoiding unnecessary detail or conflicting instructions. Depending on the product, the context might include approved terminology, business definitions, policies and business rules.

The output contract defines how the AI's response should be displayed to the end-user. This will include format and channel (email, report, summary, etc.) It should also include factors such as:

- Fields and their meanings
- Permitted values
- Required language
- Length limits
- Ordering
- Whether explanations or citations are required
- Behaviour when information is missing

The output contract may also include templates or worked examples containing placeholders to be

populated with the relevant data. The rules governing how those placeholders should be completed should also be specified.

One aspect that deserves particular attention here is the evaluation of prompts against associated acceptance criteria. Just as business analysts define acceptance criteria for product features, prompts should be evaluated against clearly defined success measures.

Evaluation should consider several dimensions including: task accuracy, rule compliance, correct use of context, output format validity, hallucination rate, tone, cost, consistency across repeated runs.

Prompt engineering should therefore be viewed as an iterative cycle, rather than a one-off activity: Requirements → Prompt → Test cases → Evaluation → Prompt revision → Production monitoring

Non-functional Requirements for the Prompt

Prompt-specific non-functional requirements can be grouped into four main categories. Reliability and consistency requirements ensure the model produces stable results across similar inputs, wording changes, different input lengths, and repeated runs. That can include output consistency, semantic stability, context robustness, evaluation repeatability, etc.

Accuracy and transparency requirements cover hallucination rate, factual support, and uncertainty

handling. The model should rely on approved sources, avoid unsupported claims, and clearly communicate when information is missing.

Control and safety requirements cover instruction adherence, prompt-injection resistance, format compliance, and refusal quality. These requirements ensure the model follows priorities, maintains the expected output structure, resists malicious instructions, and rejects inappropriate requests safely.

Finally, efficiency and compatibility requirements cover token efficiency and model portability. The prompt should achieve the required quality without unnecessary tokens and

continue to perform acceptably when models or model versions change.

Prompt engineering is a new way of writing requirements. Business analysts already understand the basics: how to define the business context, gather requirements, and separate functional from non-functional needs. These activities are already part of our daily work. The next step is simply to adapt these requirements for a new audience—the AI.

Darya Miler, is an experienced Technical Program Manager and certified Product Owner. You can contact her via www.linkedin.com/in/darya-miler

What if BAs could help shape demand, rather than just respond to it?

Internal BA Consultancy: Shaping Demand

Practical one-day training
delivered in-house for BA teams.

I Don't Want to Be in a Meeting Room with Technical People

Rakesh Patange



The title of this article is an actual sentence uttered by a business process owner in a recent project meeting. Fortunately, it was not a meeting I was involved in. Before explaining further, let me give you some context.

The Project Environment

The framework I work under places the business process owner in one government agency, while the technical team belongs to another. Both agencies work on a mutually agreed project timeline and interact

closely throughout the project life cycle. Both agencies have their own BAs. In such an environment, everyone tries to work professionally and collaboratively. It is rare to hear statements that are as troubling as the one in the title.

Voicing Concerns

In a recent project meeting, a project manager asked all key business process stakeholders to attend a solution demonstration so they could see the system's capabilities firsthand. One of the process owners

responded promptly, saying she does not want to be in that meeting room. She went on to explain further: “Those technical people speak a language that no one understands in our business team. When they explain a feature, they don’t contextualise it with the supporting business process. They speak fast, and assume that every one of us is as technically oriented as they are. As a result, we hardly get any value out of such meetings.”

Is This Reaction Justified?

Was this comment too harsh? Should softer wording have been used?

Maybe.

However, if we go beyond the words and try to understand and empathise with the underlying sentiment, it reveals a darker truth.

The Persistent Challenge of Communication

Today, AI helps BAs in many ways, from developing business requirements and defining scope boundaries, to developing user acceptance criteria, right up to refining solution delivery tasks and beyond. But when it comes to face-to-face interactions, no AI command can replace the human touch. When business stakeholders interact with technical teams, confusion can set in rapidly, and emotions can flare up. I engaged in introspection, recalling past events from my career. My introspection revealed some interesting insights, which I’ll expand upon below.

Fundamental Differences in Focus

Business requirements originate from a business problem or pain point. That translates into a set of features needed in a solution—often an IT-enabled one—and becomes a suite of user stories or other requirements artefacts that define what needs to be built or changed. This then narrows down to technical details. The Key Performance Indicators (KPIs) for a business process owner are clearly different from the KPIs of a development-oriented technical team member. A process owner will be tracking end-to-end business outcomes, while a development team will likely be focusing on the elements of the solution that have been shipped.

During the final stages of solution demonstrations, I have seen a gap between the way technical team members *present* the solution, and the key business representatives’ understanding of the solution’s capabilities.

For example, one solution was a client expense claims portal. It had configurable background parameters (such as claim entry limits, approval limits, and repeat claim thresholds). This configuration was structured as a list of technical jargon, rather than in business-friendly language. Presenting these parameters in technical language is problematic, as they exist to ensure the solution is flexible enough to avoid coding updates when business processes or agency policies change. This is a critical technical aspect that business

process owners must understand for effective operation.

If business stakeholders develop the view that their interaction with the solution is overly technical, they will likely disengage. This is an example of the emerging trend in which a specific aspect of technology is exposed to business to gain greater control over outcomes, but highlights how a lack of common understanding of that element of the technology leads to suboptimal outcomes.

Bridging the Gap: Training or Mediation?

So, what is the best approach here? Should the technical people be trained to speak business language, or should the business process owners be trained to understand the technicalities? And importantly, what role do we, business analysts, play in such business-versus-technology situations?

Back to First Principles

I would recommend focusing on first principles, which I believe are:

- The human mind understands the user interface of the newly built/enhanced solution.
- The human mind understands pictures and does not like to read too much over a short time.
- The human mind does not like a situation in which something explained by another person is incomprehensible, because it generates feelings of inferiority or inadequacy.

A Layered Approach to Engagement

Based on these foundational principles, I follow a layered approach:

Layer 1: Addressing Business Pain Points

At the first layer, I create a presentation that outlines the business pain points that drive the business case and how each is addressed in the newly built/enhanced solution. This layer contains a high-level requirement statement tied to a set of system steps. The objective here is to demonstrate that the new/enhanced system has addressed every business pain point. Some key business process owners do not wish to go beyond this stage, as their operational staff are responsible for executing the subsequent steps within the system.

Layer 2: Focusing on Operational Staff

The second layer is for the operational staff who need to understand the system as an integral part of their work. This layer is more about the **how** than the **why** of the solution. The **why** part is already addressed in the first layer. I ensure that solution demonstrations happen under my supervision at this layer, so that any technical jargon, if thrown around, can be quickly contextualised by tying it to the business requirement statements I originally documented in the specification.

Layer 3: Engaging Technically Oriented Stakeholders

The third layer consists of business stakeholders who are more technically oriented and keen to understand how things work in more detail. It is precisely those technical details and tasks that I mentioned earlier that are generally assigned to such people. For this group, I ensure that a detailed technical explanation and demonstration are conducted in close collaboration with the technical team.

Conclusion

The term "Divide & Conquer" may not

seem appropriate in the current era, but that is essentially what I do to avoid statements from my stakeholders, such as "I don't want to be in a meeting room with the technical people".

Rakesh is currently working as a Business Analyst with the Department of Veterans Affairs in Australia and is engaged in various projects under the 'Legislative Reform Program'. Each project involves changes and improvements to current business processes, supported by the design of ICT systems.

You can contact him on www.linkedin.com/in/rakeshpatange



BECOME AN AI-READY BA

Live learning and trusted guidance from Angela Wick



FREE DOWNLOAD

BABOK v3 IN AN AI WORLD

A practical guide to tasks, techniques, and competencies



SCAN TO GET YOUR
FREE GUIDE

5 LIVE/INTERACTIVE AI-READY BA COURSES

- ✓ BA Role in the Age of AI
- ✓ AI Literacy For BAs

- ✓ AI-Enabled Systems & Process Design
- ✓ Analysis for Agentic AI
- ✓ AI-Accelerated Business Analysis

**SAVE 15%
WITH CODE
BADIGEST**

Each course stands alone. Choose one or bundle all five.
2-4 days | \$499-\$999 each | All 5 for \$2,395 | Group of 3+?
Save even more!

EXPLORE COURSES
www.maven.com/angela-wick

Human-in-the-loop (HITL): How AI Is Accelerating the Transformation of a Business Analyst Role

Pratik Pitale

Throughout my career as a business analyst, I have often been surprised by how much time we spend documenting rather than analysing. Process maps, user stories, acceptance criteria, gap analyses and stakeholder presentations are all important, but together they can consume a significant proportion of the working day. When I first gained access to AI tools through my organisation, I was curious about the capabilities of AI, given the hype that surrounds it. I began experimenting with AI tools and working to integrate their use cases into my day-to-day work. What followed were months of building, testing, adjusting and implementing AI within organisational guidelines.

The result: Surprised by the scale, it has transformed my abilities as a BA by adopting the *Human-In-The-Loop* concept.

Understanding The Human-in-the-loop Concept

I first came across the modern "Human-in-the-loop" AI paradigm [through the work of Fabio Massimo Zanzotto](#). He formalised the concept as a crucial framework ensuring humans remain active participants in the process. This closely relates to the work carried out by Norbert Wiener, the MIT mathematician. In his book, [Cybernetics or Control and Communication in the Animal and the Machine](#) Wiener argued that the most effective systems rely on continuous feedback loops where output is monitored, corrected, and fed back into the system to improve the next cycle.

It is my view that these principles map almost perfectly onto how a BA

should interact with AI; not as a passive recipient of outcomes, but as an active participant at every stage of the loop. The human is evaluating the output, providing feedback, and helping refine and improve the model and its output.

Before realising this, my initial interaction with AI was something like:

- 1. Copy requirements from a static document
- 2. Paste requirements into the chat (an AI tool/LLM etc.)
- 3. Write a simple prompt
- 4. Hope for the best outcome

However, when following this approach the responses were generic. They lacked specificity, missed information, and to improve the output required much more effort and thus negating the time saving I had hoped for. I'm sure many readers will relate to this scenario.

This is a trap where many users might stop exploring AI capabilities and return to their usual tried and tested approaches.

However, I remained curious and continued. What worked for me was utilising a structured workflow with the BA involvement at every step of the AI interaction. Not just at the end, but throughout the interaction and the very first thing that must be refined is the prompt.

I have found that an effective prompt should include items such as:

- Role and persona
- Primary goal

- Knowledge hub with additional context (Repository of documents that defines standards and quality required for the output)
- Reasoning guidelines (To limit the scope)
- Outcome format

Below is my interpretation of a Human-In-The-Loop framework, specifically for AI in business analysis:

Human	Sets the context: Defines prompt, provides references
AI	Drafts: Creates initial draft of appropriate documents/content
Human	Refines: Refines the output, provides follow-up prompts, clarification, corrections and data cleaning
AI	Elaborates: Updates the outputs, refines the content
Human	Governance: Reviews, validates and examines the outcome

Transition from Creator to Orchestrator

Before I started using AI in my work, I used to spend hours drafting the requirement documents, refining them to make sure they met the standard. This felt natural because I felt this is what BAs are supposed to do: BAs were, in my mind, a traditional creator role. A challenge arises when the creator becomes a bottleneck. When the volume increases, in order to keep up, there's a risk that the quality of the

documentation or other artefacts deteriorates.

What I built with AI shifted how I related to my work. Rather than being the one who creates the document, I set the standards and structure, review the output *iteratively*, refining the prompt every step of the way until I get the desired outcome. In other words, I became an orchestrator.

Building the Workflow in Practice

The specific workflow I built was around the most time-intensive recurring artefacts in my area of responsibility. I used the requirement analysis I performed over a period and drafted a prompt sequence that summarised and categorised the requirement analysis which I used as an input to the AI.

At each stage, I validated, refined and corrected the feedback prompt before moving to the next stage, hence putting humans in the loop. The result is not just the AI produced document, rather it is a document produced by a BA who used AI to work at pace and volume which otherwise would be challenging.

What This Means for the BA Community and Organisations

The Human-In-The-Loop model has the potential to reshape the BA role by combining human judgement with AI assistance. In practice, this can help significantly reduce the time required for routine tasks while allowing analysts to focus more of

their effort on analysis, stakeholder engagement and decision-making.

As a BA, I usually spend around 2–3 hours per user story (depending on the complexity) across multiple refinement sessions. Mock-ups and gap analysis can sometimes take weeks. By adopting the HITL model, I was able to cut short almost 50% of the time required to write user stories and around 80–90% of the effort needed for preliminary gap analysis and wireframes. Overall, it has helped me translate a lot of manual effort into faster and more productive output.

What AI did exceptionally well was high volume routine work, i.e. structural drafting, initial elaboration etc.

What a human BA brings, and no AI can substitute is domain knowledge and judgment, stakeholder relationship management, organisational value alignment, requirement prioritisation and accountability along with the governance.

These are not peripheral activities; rather, these are the core activities that a BA is known for. Human-In-The-Loop protects them and amplifies them by freeing up labour-intensive manual tasks.

Conclusion

The Human-In-The-Loop is a core differentiator.

Over the past few months in a genuinely complex domain, working on initiatives involving digital transformation, the HITL AI model delivered fundamental change in the

BA ways of working. It measurably reduced the manual effort required to improve the consistency and the completeness of analysis documents, and improved stakeholder engagement through the process. The outcomes were real and quantifiable.

But the important finding from this implementation was not just the quantifiable increase in productivity. It was that every meaningful quality was driven by the business analyst remaining actively involved throughout the process. Every user story that reached development without ambiguity, every gap identified before it became a defect, and every stakeholder session that ran smoothly because the documentation was ready reflected

the contribution of a human who was genuinely in the loop.

The BA was not simply reviewing AI-generated outputs or providing a final sign-off. They were directing, refining, challenging and governing the work at every stage.

Pratik Pitale is an experienced BA working in insurance, driving large-scale digital transformation initiatives with a strong focus on Agile delivery and process improvement. He's passionate about AI and believes in using ethical AI to improve productivity, efficiency, and business outcomes. You can connect with him on LinkedIn at:

www.linkedin.com/in/pratik-pitale-960116a4/

UPCOMING
WEBINARS

#BACommunity
Webinar Series

EVERY REQUIREMENT IS A NEGOTIATION

WITH
CHRISTELLE
GOVENDER

 OCTOBER 15 6:30 PM – 7:30 PM BST

Blackmetric.

#BACommunity
Webinar Series

WHEN ARTIFICIAL INTELLIGENCE MEETS BUSINESS ANALYSIS INTELLIGENCE

WITH
ANURITA RUSCOE

 NOVEMBER 10 6:30 PM – 7:30 PM GMT

Blackmetric.

Click For
More Info

BA Back Stories

Debra Paul



Interviewed by Christina Lovelock

After 20 years leading specialist training and consultancy company [AssistKD](#), Dr Debra Paul has moved into a non-exec director role. Debra has been instrumental in advancing the BA profession globally, is an author of [many books](#), created qualifications and established events and communities including the [BA Manager Forum](#). Additionally, she's a founding member of the [BA Conference Europe](#) committee, and has been central to the conference's

success and continued growth.

Debra has just celebrated the launch of the new [BCS Service Design book](#) and is planning what's coming next, which might include new books and new projects.

How did you get started in your career?

I joined the Civil Service, which is brilliant at giving people learning and career opportunities. On my first day



in the Civil Service I encountered computing. Before that, I had not used it very much as I didn't have it at home and didn't use it much at school. When I encountered computing, I thought it was absolutely wonderful. So, I knew I always wanted to work in something aligned with technology. I moved through a few roles in the Civil Service, ending up as a systems analyst. I had worked on a lot of implementation projects and had done a lot of work with people who were unsure, nervous and apprehensive about technology.

I also worked on projects where the system was fine but the processes wrapping around it were poor. So, when I became a systems analyst, I was very aware of business context. I loved systems analysis. I was chief analyst on a project, and we did things that were very forward-looking. We worked collaboratively with the users and the developers, sitting alongside each other—this was very successful.

I also had an affinity with understanding the law, the policy, and what it meant in practice. So, I kept pushing at that door, asking “How is this going to work?”, I talked a lot with user representatives about policy and law.

When I left the Civil Service, I moved into a consultancy role, working at a more strategic level. This allowed me to understand the overall business context.

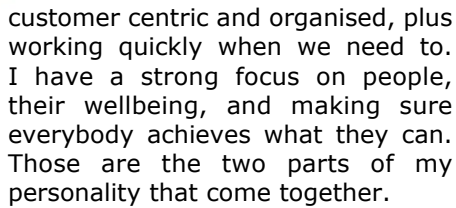
I started putting forward that it wasn't enough to just understand the technology and pushed this everywhere I could. I was on the

board of the Structured Systems Analysis and Design Method (SSADM) user group and spoke to various people about this. There were others championing this idea, but not many at that time. *Systems* analysis was the focus but I kept saying, “There's also this area called *business* analysis! That is where we should be going. We should encompass systems analysis skills within the business context.” I was working with BCS at the time. I wrote a syllabus for a module that is still a module in the [BA diploma](#). That was how it all started for me. I worked on projects as an independent consultant by that time and looked at *the use of* technology, not just the technology itself. Everything went from there!

Is there a skill or trait that's particularly helped you in your career?

Resilience. And my children say persistence is my superpower! In terms of skills, I'm very organised and I can multitask. It has been commented on how many plates I can spin at the same time. I can flip between things, but I am very organised. I don't get easily distracted. I'm very focused. This has been a big factor in helping my career.

Many years ago, I took an occupational personality questionnaire test at work. The person reviewing it with me said, you've got an unusual profile because you have analytical, logical, organised traits, but you also have empathy and people aspects. You don't usually see two aspects together. I didn't think about it at the time, but I have since. While I want things to be run well, I also want everything to be focused,



I am always looking for how we can improve. BCS used to have the strap line, 'making IT good for society'. I think my own strapline would be 'making change work for everyone'. But also looking at new ideas, new ways of modelling, new ways of using technology that can benefit the situation and having that outcome focus. If we're doing anything, my question is, "Why are we doing it? How do we get from A to B? What's possible? How can we innovate this? How can we make it good for people and use technology effectively?" That has been my approach through everything I've done.

Yes, it's quite depressing after all these years! The recognition of business analysis at a senior or middle manager level in organisations is often less than ideal. It's disappointing when you constantly see problems in how organisations work. A good business analyst could put in place clear processes, and training for people, so that a lot of issues are avoided. A good business analyst could sort so many things out.

Sometimes, business analysts are working in a comfort zone of: "I know my systems in my company, therefore I can look at requirements and change requests." It's that lack of questioning and challenging that concerns me.

I have spent a lot of my career trying to move this conversation outside what I call the 'BA bubble', and help get that recognition that business analysis is important and really helps organisations.

There are many more business analysts! It's interesting that we have this sense that it's not sufficiently recognised, and then you find there are hundreds of thousands, if not millions, of business analysts around the world. Organisations are investing in business analysts for a reason. They must understand what they can do, but they don't always seem to utilise them effectively. There have been situations where organisations will suddenly think business analysis isn't working and have made whole BA practices redundant, which is such a shame to see.

understanding within the BA community of what we can offer and what we need to be in terms of [services and skill set](#). But you still get senior management flip-flopping. If they're not going to understand it and back it, how are we ever going to get recognition? How are organisations ever going to use the skills effectively?

There is a lot of guidance, but we still haven't broken through as effectively as I would like. I would like to see more BAs at board level. I think roles such as head of transformation on a board should always have business analysis knowledge and skills.

Is there anything you wish you'd learned earlier in your career?

Not to get too stressed when things look like they're going wrong, because things turn around.

I wish I had called out situations rather than tending to think "It's me, I'm not doing a good enough job". We've all been in situations where we've thought "It's me." In hindsight, we should have called it out for what it was.

Collaboration and community are important to you. Why is that so valuable?

Collaboration is at the heart of good work. Analysis is not a solitary activity. Whenever I've collaborated with somebody, the end result has been exponentially better.

I've been lucky to work with excellent collaborators in my career: James Cadle, Paul Turner, [Christina Lovelock](#), [Jonathan Hunsley](#), to name just a few of the people who have

been wonderful people to collaborate with, and there are so many others. When you've had collaboration that's effective, you know what a difference it makes. It is amazing what people can contribute, what we can achieve together. Collaboration is so important. Some people like to do everything themselves but I feel they don't understand what they're missing. Having collaborated effectively for decades, I know what I would be missing if I didn't collaborate.

And community is where you find your collaborators. It's where you find people who you think, "that was a brilliant conversation, we've got to take that further!" I think of the people from our community who I've worked with over the years and how fabulous it's been. Community and collaboration really are the heart of everything.

You have co-written books in business architecture and now service design. What disciplines should BAs be looking at to expand their toolkits?

When I first started looking at enterprise and business architecture, I thought it was very interesting. I was running a small company, and the more I looked into business architecture I realised this is an invaluable skill set that really knits together with business analysis. I believe that if you don't understand the essence of *capability*, then I'm not sure your analysis works as well as it could.

Something like a value stream or value chain diagram helps you achieve clarity of thought. As an

analyst, you need clarity. In previous projects, for example I did projects for a large bank, I can look back and see that although I hadn't used formal architectural techniques, I was using the thinking. On one project, when I presented information concepts, it actually stopped the project because they hadn't thought through information capture, storage and retention underpinning the whole project. Looking at it architecturally gave me a step forward.

I've written the BCS book on service design with Jonathan Hunsley. When I started looking at service and the research, it helped me to formalise areas. When you understand the service mindset, it opens up your thinking as an analyst and ensures you are outcome focused. These three skill sets bond together and are excellent ways for business analysts to develop and grow.

Other, more technical areas include detailed modelling, technology advances, thinking about how we can use AI, and also where we need to understand factors regarding security and cybersecurity. As a business analyst, the clue is in the word *business*. It's holistic. We have to have skills in different directions.

What do you wish more business analysts and change professionals were talking about?

I wish they would question 'received wisdom' more. For example, by talking about how the shiny toys are not always so shiny once you start to investigate and challenge them. If they were to do this, they would stop

running towards what seem like shiny new things that everyone needs. They would give themselves permission to take a step back, research, analyse, think. That would benefit them, our profession and their organisations.

Is there something you want to give a shout out to? Something that has inspired you?

I remember reading about the concept of [service and service design](#) many years ago and my brain going on fire because it made a lot of things make sense. As an analyst, it really made me think. So, if I was going to encourage people to look at something, it would be that.

But my biggest shout out would be: go and look at subjects in depth. Because too often we take the superficial. We take the words and accept them. Service science is my example, but my big shout out is don't accept, always question, look at what is out there and finally, give yourself time to think. If you're an analyst, you should be thinking. Read. Think. Question. Take the time to be interested and find out more.

Connect with [Debra via LinkedIn](#)

About the interviewer:

Christina Lovelock is a digital leader, author and coach. She is active in the Business Analysis professional community and champions entry-level roles. She is the author of two BCS books: [Delivering Business Analysis](#) and [Careers in Tech, Data and Digital](#). Connect with [Christina on LinkedIn](#)

***Thanks to all the authors who kindly
allowed us to use their content.***

Agnieszka Balcerzak
Brett Srawley
Line Karkov
Kent J. McDonald
Sherry McMillan
Erivan de Sena Ramos
Olabisi Odusanya
Adrian Reed
Nuno Santos
Li Yang
Nick de Voil
Anurita Ruscoe
Mark Cross
Darya Miler
Rakesh Patange
Pratik Pitale
Debra Paul & Christina Lovelock

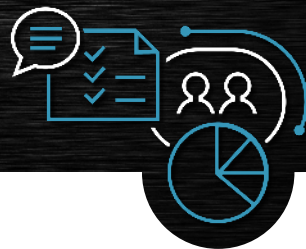
Layout & Design

Matt Smart

Licensed Artwork & Photography

Image Credit: © Julien Eichinger— [stock.adobe.com #171655696](https://stock.adobe.com/#171655696)
Yayimages.com
wordclouds.co.uk

Please note: The views expressed by the authors, contributors and advertisers are their own, and may not always reflect the views of Blackmetric Business Solutions Ltd/BA Digest. Products and services mentioned in advertisements are the responsibility of the advertiser, and queries around these should be directed to the relevant advertiser. This publication contains hyperlinks to external websites, Blackmetric/BA Digest do not own and are not responsible for the content on other sites. Trademarks, whether acknowledged in-line or not, are the property of their respective owners.



Practical, live business analysis training

We run practical, interactive training for in-house BA teams,
delivered **in person in the UK** or **live online worldwide**.

Popular courses include:



Pragmatic Benefits
Management



Mastering the
Requirements Process



Business Analysis
Planning & Monitoring



Pre-Project
Problem Analysis



Pragmatic BPMN



Systems Thinking
for BAs



Non-Functional
Requirements Workshop



Business Analysis:
Leading from the Middle



Introduction to
Business Analysis



Discovery in Practice:
Understanding Context,
Outcomes and Options

> ...and many more



Looking for training for your team?
Explore our full range of courses at

blackmetric.com/training