More Than Just Gathering Requirements

When people think about a Business Analyst's role, **gathering requirements** is often one of the first things that comes to mind. And yes, it's a key part of the job. But for me, it falls under a bigger task: **using critical thinking to analyze and organize information.** 🧠📝

Why? Because **good requirements come from asking better questions**...and this applies to others and also to yourself. *The goal isn't just to collect information but to extract valuable, high-quality insights that actually make a difference.* And usually, that comes from thinking through all the possible scenarios that could trigger an event or result from one. ↪️⤴️⤵️

🚨 Just a heads up!! Fun fact... or not-so-fun, depending on the situation! This way of thinking doesn't just apply to work—it starts creeping into every aspect of life. You'll find yourself analyzing every decision, predicting outcomes, and spotting patterns everywhere—even when you're just trying to order food or pick a movie. 🤔 What if...? Why?

Anyhow...In the section below, I'll share some techniques that help me think differently and build a mindset for getting meaningful information.

Now, let's dive in. 🚀

How to Ask Better Questions

Just a quick reminder—*this is based on my experience and what works for me. It might not be the same for you, and you may need to find your own method based on your natural strengths. 🦸 But hopefully, this gives you some ideas you can adapt in your own way.*

When I first started, I thought gathering requirements was just about listening to stakeholders and writing things down. But I quickly realized that people didn't always tell me everything I needed to know. Sometimes, they assumed I already understood; other times, they didn't even know what they were missing.

I've always been a very visual person—I understand things better when I can see them. So instead of just taking notes, I learned to visualize requirements using different models. These models helped me structure my thinking, spot gaps, and, most importantly, ask better questions.

Let me show you how.

🧩Key Models That Help Frame Questions

Using structured models or techniques—such as user stories, writing Gherkin scenarios, or process flows—can help you identify missing pieces of information and guide your questions more effectively.

Here are some different models that help me do this in different ways:

1. User Stories – Focusing on the Who and Why

One of the simplest but most powerful tools I use is the **User Story format:**

📝 *As a [user], I want [feature] so that [value/benefit]*

I like this format because it forces me (and the stakeholders) to think about **who** the feature is for and **why** it matters.

📅 Example: Online Booking System

A stakeholder tells me:
💬 *"We need an option to reschedule users appointments."*

Instead of just writing that down, I start asking:

- *Who exactly needs this?* Customers? Admins? Both?
- *Why is this important?* Are they missing too many appointments?
- *Are there any limits?* Can they reschedule anytime, or only with 24-hour notice?
- *What happens if they reschedule last-minute?* Does it create an empty slot?

By asking these questions, we can avoid unnecessary features that don't add real value or uncover hidden needs that stakeholders might not even realize they have. Also drive innovation! by exploring better ways to solve the problem.

---

2. Exploring Scenarios using Gherkin - Focusing on the What If

When I need to dig into different situations and edge cases, I rather use the **Gherkin format:**

📝 *Given* [starting condition], *When* [action happens], *Then* [expected outcome]

This method is great for figuring out **what should happen in different scenarios.**

🏧 Example: ATM Cash Withdrawal

I first break *"The user withdraws money"* down like this:

✅ **Given:** The user inserts their ATM card.
✅ **When:** They enter their PIN and request $500.
✅ **Then:** The ATM dispenses the cash and updates the balance.

And here's where you think through **all possible scenarios**—asking the **What if?** questions:

- *What if they enter the wrong PIN?* (How many attempts before lockout?)
- *What if they don't have enough funds?* (Does the ATM suggest a lower amount?)
- *What if the ATM is out of $20 bills?* (Cancel transaction? Offer different bills?)
- *What if the card is expired?* (Show an error? Allow deposits but not withdrawals?)

By thinking through these scenarios, you help prevent last-minute surprises and ensure a smoother user experience. After all, **nobody wants to see an angry user yelling and kicking an ATM machine!** 😅

---

3. Process Models – Visualizing the Flow

When I need to understand **step-by-step processes**, I like flowcharts or swimlane **diagrams.** Makes it **so much easier** to spot missing details.

💹 **Example: Online Order Process**

A company wants to streamline their **online order process**. Instead of just writing down steps, I **map it out** and ask:

- *What triggers an order?* Clicking 'Buy Now'? Adding to cart?
- *When does payment get processed?* Before or after confirmation email?
- *What happens if an item is out of stock?* Does it go on backorder, or does the order get canceled?
- *Are 'these steps' always in the same order? Can they happen in parallel?*
- *Can users edit orders after checkout?* Or do they have to call support?

This approach helps me **see gaps** in the process as missing steps, unclear transitions, or shows if a step can happen independently, avoiding unnecessary waiting times (no one likes being stuck in a virtual queue!).

---

4. Context Models – Understanding Data & Dependencies

Whenever I'm dealing with multiple systems, I use **Context Diagrams** to map out **what connects to what.**

This helps me quickly understand:
✅ **Who interacts with the system** (users, external systems, third parties).
✅ **Where the data comes from** (input sources).
✅ **Where the data goes** (outputs, reports, notifications).

How This Helps in Real Life 🌍

Let's say I just started a new job or switched projects, and the team is already deep into the work. I need to **get up to speed fast**. Instead of reading endless documentation or sitting through hours of meetings, I can **build a Context Diagram** to get a high level overview of how everything connects.

👩‍🏫 **Example: Student Registration Portal**
If I join a project for a university's online registration system, I'll ask questions like:

- *Who uses this system? (Students, Admins, Professors?)*
- *What other systems does it connect to? (Payment systems, Course Catalog, Email Notifications?)*
- *Where does the student data go after registration? (CRM, Reports, Dashboards?)*
- *What external constraints should I consider? (Compliance, Data Privacy Regulations?)*

By **visually mapping this out**, I can **ask better questions**, spot **gaps or missing connections**, and **get a clear understanding of the big picture much faster**—without drowning in scattered information.

🗺️ It's like **getting a map before exploring a new city**—you don't need every single detail upfront, but knowing the main roads and key landmarks helps you navigate efficiently. 🚀

To Sum Up – How to Ask Better Questions

✅ I make information visual whenever possible
✅ I use different models to structure my thinking
✅ I ask "WHAT IF?" and "WHY?" often
✅ Focus on WHO and WHY, not just WHAT

At the end of the day, it doesn't matter how great a product is if no one needs it or uses it.