Explanation of Few Business Analysis Terminologies Using Real World Examples

Created By – Diwakar Kumar Singh

Business Analysis

Imagine you have a lemonade stand. You're trying to make the best lemonade possible and sell as much as you can. But sometimes things don't go as planned, and you want to figure out why and how to improve.

That's where business analysis comes in. A business analyst is like a detective who investigates your lemonade stand. They ask questions and gather information to understand what's happening.

They might talk to you, your customers, and other people involved in your lemonade stand. They'll also look at data, like how much lemonade you sell each day, what the weather is like, and how much money you make.

Once they have all the information, they analyze it to find patterns and clues. They try to figure out why some days you sell more lemonade than others, or why certain customers prefer your lemonade over others.

Based on their analysis, they come up with ideas and recommendations to make your lemonade stand even better. They might suggest changing your recipe, offering new flavors, or improving your marketing to attract more customers. They help you decide which ideas are the best and most realistic for your lemonade stand. They explain the benefits and risks of each option, so you can make an informed decision.

Once you decide on a plan, the business analyst helps you put it into action. They guide you through the changes, making sure everything is going smoothly and that you're on track to improve your lemonade stand.

Finally, they keep checking in to see how your lemonade stand is doing. They measure the results of the changes you made and make adjustments if needed. Their goal is to help your lemonade stand be as successful as possible.

In simple terms, a business analyst is like a detective who helps you understand why your lemonade stand is doing well or not so well, and they give you ideas to make it even better. They're like a secret weapon to help you succeed in your business!

SDLC

Let's take the example of building a house to explain the Software Development Life Cycle (SDLC).

Requirements Gathering: This phase is like understanding the needs and requirements of the homeowner before starting the construction of the house. You would gather information about the number of rooms, desired layout, materials, and any specific features they want in their house.

System Design: In this phase, you would create a blueprint or architectural design of the house based on the gathered requirements. It involves deciding the layout of rooms, the placement of windows and doors, and other structural elements. This step helps visualize the final product before building it.

Implementation: This phase is equivalent to the actual construction of the house. Based on the design, construction workers would start building the foundation, walls, roof, and installing various systems like plumbing and electrical. They would follow the plan and use appropriate materials and techniques to build the house according to the design.

Testing: Once the construction is complete, various tests are conducted to ensure that everything is functioning correctly. Inspections may be carried out to check the quality of construction, the stability of the structure, and the functionality of the installed systems. This is done to identify and fix any issues or defects before handing over the house to the owner.

Deployment: After successful testing, the house is ready for occupancy. The homeowner can move in and start using the house as intended. This phase is equivalent to the deployment of the software, where it is made available for users to utilize.
Maintenance: Just like a house requires regular maintenance to keep it in good condition, software also needs maintenance to address any issues that arise over time. The homeowner would perform routine tasks like cleaning, repairs, and upgrades to ensure the house continues to meet their needs. Similarly, in software development, bug fixes, updates, and enhancements are carried out to keep the software running smoothly and adapt it to changing requirements.

By comparing the SDLC with the process of building a house, we can see how different phases like requirements gathering, design, implementation, testing, deployment, and maintenance are involved in both scenarios. This analogy helps to understand the sequential nature of the SDLC and how each phase contributes to the overall development process.

Project Scope

Let's say you're planning a construction project to build a new office building. The project scope refers to the boundaries and deliverables of the project—the specific work that needs to be accomplished to successfully complete the project.

Here's an example of defining the project scope for building the office building:

Objective: The objective of the project is to construct a five-story office building with a total floor area of 10,000 square meters.

Deliverables: The deliverables of the project include:

- Completed architectural and structural designs
- Obtained building permits and regulatory approvals
- Cleared the construction site and prepared the foundation
- Constructed the building structure, including walls, floors, and roof

- Installed electrical, plumbing, and HVAC systems
- Completed interior finishing, such as flooring, painting, and fixtures
- Set up the necessary security and fire safety measures
- Completed landscaping and external amenities
- Obtained a certificate of occupancy

Exclusions: The project scope should also define what is not included in the project, such as:

- Demolition of existing structures on the site
- Procurement of office furniture and equipment
- Hiring and training of staff for the new office
- Ongoing maintenance and operational activities after construction

Constraints: The project scope should consider any limitations or constraints that might affect the project, such as:

Budget: The total construction cost should not exceed $10 million.
Timeframe: The project should be completed within 12 months.
Regulations: The construction must comply with local building codes and regulations.

Assumptions: The project scope may include assumptions made during planning, such as:

- Availability of skilled labor and construction materials
- Stable weather conditions during the construction period
- No major unforeseen obstacles or delays

By clearly defining the project scope, you establish a baseline for what is expected and ensure that everyone involved understands the boundaries of the project. This helps to manage expectations, allocate resources effectively, and ultimately deliver a successful outcome within the defined scope.

Use Case Diagram

Imagine you're planning a road trip with your friends. Before you hit the road, you need to plan various activities and make sure everyone is on the same page. A use case diagram can help you visualize the different actions and interactions involved in this scenario.

In a use case diagram, you have actors and use cases. Actors are the individuals or entities interacting with the system, while use cases represent specific actions or goals.

Let's identify the actors in our road trip example. You and your friends are the primary actors, as you're the ones participating in the activities. However, there can be other secondary actors like the car, GPS system, or even a travel planner.

Now, let's think about the different actions or use cases involved. Here are a few examples:

Plan Trip: This use case represents the activity of collectively deciding the destination, dates, and activities for the road trip. It involves interactions between you and your friends.

Book Accommodation: This use case involves finding and reserving a place to stay during the trip. It may include interactions with a hotel booking website or a travel agent.

Pack Luggage: This use case focuses on the process of packing your belongings for the trip. It doesn't involve any external interactions, as it's something you and your friends do individually.

Navigate: This use case involves using a GPS system or a map to find the best routes and reach your destination. It includes interactions between you and the GPS system.

Stop for Food: This use case represents the need to find places to eat during the trip. It can involve interactions with online restaurant reviews or asking locals for recommendations.

In the use case diagram, you would represent these actors and use cases using simple shapes and lines. The actors would be represented as stick figures, while the use cases would be represented as ovals or rectangles. Lines connecting the actors to the use cases show the interactions or relationships.

Remember, a use case diagram helps you understand the different activities and interactions involved in a system or scenario. It provides a high-level view of how different components work together to achieve specific goals, just like planning a road trip with your friends.

Stakeholder Analysis

Imagine a government planning to construct a new highway connecting two major cities. In this scenario, various stakeholders can be identified:

Government Authorities: The government officials responsible for infrastructure development, transportation departments, and regulatory bodies will be directly involved in the project. Their interests may include economic development, efficient transportation, and environmental impact.

Local Communities: The residents living in the vicinity of the proposed highway will be directly affected. Their concerns may include noise pollution, increased traffic, potential displacement, and the impact on their quality of life. Engaging with them is crucial to address their concerns and ensure their support.

Environmental Groups: Organizations focused on environmental conservation and sustainability will be concerned about the potential impact of the highway on the surrounding ecosystem, wildlife habitats, and air and water quality. They may advocate for measures to mitigate environmental damage.

Construction Companies: Private contractors involved in the construction process have an interest in securing the contract and ensuring a profitable project. They will be concerned about the project timeline, budget, and compliance with safety regulations.

Local Businesses: Business owners, such as restaurants, hotels, and gas stations, located along the existing roads or in proximity to the proposed highway, will be interested in understanding how the new infrastructure will affect their operations. They may want assurances that their businesses will not be negatively impacted.

Transportation Providers: Public transport operators and logistics companies might have a stake in the project due to potential changes in transportation routes, connections, and competition. They may want to ensure that the new highway doesn't disrupt their operations or create unfair advantages for certain players.

Once these stakeholders are identified, a stakeholder analysis would involve assessing their level of influence, their level of interest or concern, and their potential impact on the project. This analysis helps in prioritizing stakeholders, understanding their expectations, and developing appropriate strategies for engagement and communication. It enables project planners to address concerns, build support, and make informed decisions that consider the interests of all relevant stakeholders.

Requirement Gathering

Imagine you're throwing a party and you want to order a cake from a bakery. Before you can place the order, you need to gather the requirements for the cake. Here's how the requirement gathering process might go:

Identify the purpose: You start by clarifying the purpose of the cake. Is it for a birthday, an anniversary, or some other celebration? This helps set the context and expectations.

Discuss the basic details: You talk to the bakery to get a sense of the basic requirements. How many people will attend the party? Do you have any specific flavors or dietary restrictions? This helps define the overall scope of the cake.

Gather specific preferences: You ask the attendees about their preferences. Some might prefer chocolate, while others might like vanilla. Some may have allergies or dietary restrictions that need to be considered. This helps gather specific details and constraints.

Define design elements: You discuss the design of the cake. Is there a particular theme or color scheme for the party? Are there any specific decorations or personalized messages? This helps capture the visual and creative aspects of the cake.

Confirm delivery logistics: You finalize the logistics with the bakery. When and where should the cake be delivered? Are there any special instructions or requirements? This ensures a smooth delivery process.

By going through this process, you've gathered all the necessary information about the cake: the size, flavors, design, and delivery details. This information helps the bakery understand exactly what you need and ensures that they can deliver a cake that meets your requirements.

In the same way, requirement gathering in software development or any other project involves identifying the purpose, discussing the basic details, gathering specific preferences, defining design elements, and confirming logistics. It ensures that everyone involved understands and agrees on what needs to be done, leading to successful project completion.

Business Process Modelling

Business Process Modeling is a way of visually representing how a business operates and how its activities are connected. It helps people understand and analyze the various steps, tasks, and interactions involved in achieving a specific goal within a business.

Let's take a simple real-world example to explain this concept. Imagine you are running a small coffee shop. The process of serving a customer can be broken down into several steps, such as:

Greeting the customer: When a customer enters your coffee shop, the first step is to greet them with a warm welcome.

Taking the order: The next step is to ask the customer for their coffee order. You may have a menu board or a point-of-sale system to help with this.

Preparing the coffee: Once the order is received, your barista starts preparing the coffee based on the customer's preferences. This involves grinding the coffee beans, brewing the coffee, adding milk or sugar if requested, and so on.

Serving the coffee: After the coffee is ready, it needs to be served to the customer. This can involve placing it on a tray, adding any extras like a napkin or a small cookie, and delivering it to the customer's table.

Payment: Once the customer has enjoyed their coffee, they need to pay for it. You may have a cashier or a self-service payment system for this step.

Farewell: Lastly, you bid the customer goodbye, thanking them for their visit and inviting them to return.

Now, with business process modeling, you can represent these steps visually using symbols and diagrams. For example, you could use boxes to represent each step and arrows to show the flow between them. This way, you can see the sequence of activities, identify any bottlenecks or inefficiencies, and make improvements to streamline the process.

Business process modeling helps you analyze the steps involved, identify areas for improvement, and make informed decisions to enhance efficiency, reduce errors, and deliver a better customer experience. It provides a clear overview of how your business operates and serves as a guide to ensure smooth operations.

Feasibility Analysis

Feasibility analysis is a process of evaluating whether a particular idea, project, or plan is realistic and achievable. It involves assessing the potential benefits, costs, and risks associated with the idea to determine if it is worth pursuing.
Let's take the example of opening a small coffee shop. Before starting the coffee shop, you would need to conduct a feasibility analysis to see if it is a good idea. Here's how it would work:

Market Analysis: You would research the local market to understand if there is a demand for coffee shops in the area. You might consider factors like the number of competitors, the population density, and the demographics of the target customers. If the market is already saturated with coffee shops or there is not enough demand, it may not be feasible to open a new one.

Financial Analysis: You would analyze the costs involved in starting and operating the coffee shop. This includes expenses such as rent, equipment, supplies, employee salaries, and utilities. You would also estimate the potential revenue based on the number of customers you expect to serve and the price you plan to charge for your products. If the costs outweigh the potential earnings, it may not be financially feasible.

Location Analysis: The choice of location is crucial for a coffee shop. You would evaluate different locations based on factors like foot traffic, accessibility, and competition. A location with high visibility and easy access for customers would be more favorable. If you can't find a suitable location that meets your requirements, it may affect the feasibility of the coffee shop.

Operational Analysis: You would consider the day-to-day operations of the coffee shop. This includes tasks like sourcing ingredients, preparing coffee, managing inventory, handling customer service, and maintaining cleanliness. You would assess if you have the necessary skills, resources, and support to run the coffee shop effectively. If the operational requirements are too complex or you lack the necessary resources, it may not be feasible to proceed.

By conducting a feasibility analysis, you can assess the viability of your coffee shop idea before investing time and money into it. If the analysis indicates that the idea is feasible, you can proceed with confidence. If it suggests that the idea is not feasible, you can either modify your plans or explore other opportunities.

Functional Requirements

Functional requirements are the specific tasks or actions that a system or product needs to be able to perform in order to fulfill its intended purpose.

Imagine you are designing a mobile banking app. The functional requirements for this app would include the following:

User Registration: The app should allow users to create an account by providing their personal information, such as name, email address, and password.

Login and Authentication: Once registered, users should be able to log in to their accounts using their email and password. The app needs to verify their credentials to ensure only authorized users can access their accounts.

Account Balance and Transaction History: After logging in, users should be able to view their account balance and transaction history. This means the app needs to securely fetch and display this information from the bank's database.

Funds Transfer: Users should be able to transfer money from their account to another account within the same bank or to external accounts. The app should provide a user-friendly interface to input the recipient's details, such as account number or email address, and initiate the transfer securely.

Bill Payment: The app should allow users to pay bills, such as utilities or credit card bills, directly from their bank account. This requires integrating with payment gateways and securely processing the transactions.

Notifications: The app should send notifications to users for important events, such as successful transactions, low balance alerts, or security-related information.

Account Settings: Users should be able to update their account information, such as changing their password, updating contact details, or enabling additional security features like two-factor authentication.

These are some of the functional requirements that the mobile banking app needs to meet in order to provide a useful and convenient banking experience for its users.

Non-Functional Requirements

Non-functional requirements, also known as quality requirements, are the criteria that describe how a system should behave rather than what it should do. They focus on aspects such as performance, reliability, usability, security, and maintainability. Let me give you some simple real-world examples to help you understand:

Performance: Imagine you're buying a new smartphone. You might have a non-functional requirement that it should have fast processing speed so that apps open quickly and the device doesn't lag or freeze when you're using it.

Reliability: When you book a flight ticket online, you expect the system to be reliable. In this case, a non-functional requirement could be that the system should have high availability, meaning it should be accessible and operational almost all the time without frequent crashes or downtime.

Usability: Suppose you're using a website to buy clothes. A non-functional requirement related to usability could be that the website should be user-friendly and easy to navigate, with clear instructions and intuitive interfaces, so that customers can quickly find the items they want to purchase.

Security: When you use online banking services, you want your personal and financial information to be protected. A non-functional requirement in this case would be that the system should have strong security measures in place to safeguard your data from unauthorized access or hacking attempts.

Maintainability: Consider a software application used in a company for managing employee records. A non-functional requirement for maintainability could be that the application should be easy to update or modify when new features are needed or when changes in regulations require adjustments.

These are just a few examples of non-functional requirements in everyday situations. Non-functional requirements are important because they ensure that a system not only performs its intended functions but also meets the necessary quality standards to provide a satisfactory user experience and fulfill other important aspects, such as security and reliability.

Gap Analysis

Gap analysis is a way to understand the difference between where you are right now and where you want to be in the future. It helps you identify the "gaps" or differences between your current situation and your desired goal.

Let's say you want to improve your health and fitness. You have a goal to run a marathon, but right now you can only run for 10 minutes without getting tired. Here's how gap analysis can help you:

Identify your current state: You start by understanding your current situation. You assess your fitness level, which is being able to run for 10 minutes.

Define your desired state: Next, you set your desired goal, which is to run a marathon. Running a marathon requires a much higher level of endurance and stamina than you currently have.

Analyze the gap: Now comes the gap analysis. You compare your current state (10 minutes of running) with your desired state (running a marathon). The difference between the two is the "gap." In this case, the gap is quite big because you need to significantly improve your endurance and stamina.

Plan for bridging the gap: Once you've identified the gap, you can create a plan to bridge it. You might start by gradually increasing your running time each week, following a training program, incorporating strength training, and improving your diet. These actions will help you close the gap and get closer to your goal of running a marathon.

Monitor progress: As you implement your plan, you regularly monitor your progress. You track how long you can run, measure your improvement, and see how close you are to reaching your goal. This helps you stay motivated and make any necessary adjustments to your plan along the way.

By using gap analysis, you can break down a big goal into smaller, actionable steps. It helps you understand where you are, where you want to be, and what you need to do to get there.

Root Cause Analysis

Root cause analysis is a method used to find the main reason behind a problem or issue. It helps us understand why something went wrong so that we can fix it effectively and prevent it from happening again in the future. Let me explain it with a simple example:

Imagine you have a leak in your bathroom. Water is dripping from the ceiling, and it's causing damage to the floor below. You want to fix the leak, but you also want to make sure it doesn't happen again. So, you decide to perform a root cause analysis.

First, you observe the problem and gather information. You notice that the leak is coming from the area around the shower. However, you don't stop there. You keep asking "why" until you reach the root cause.
You start by asking yourself, "Why is there a leak?" The answer might be, "Because water is escaping from the shower area." Then you ask, "Why is water escaping from the shower area?" The answer could be, "Because the shower faucet is not closing properly." Then you ask, "Why is the shower faucet not closing properly?" The answer might be, "Because the rubber seal inside the faucet is worn out." Finally, you ask, "Why is the rubber seal worn out?" The answer could be, "Because it has been used for many years and hasn't been replaced."

By asking "why" multiple times, you've identified the root cause of the problem: the worn-out rubber seal. Now that you know the main reason for the leak, you can take appropriate actions to fix it. In this case, you would replace the rubber seal in the shower faucet.

The purpose of root cause analysis is to dig deeper and find the underlying cause of a problem. By addressing the root cause, you can implement effective solutions that prevent the issue from recurring.

Requirement Prioritization

Requirement prioritization is the process of deciding which tasks or features are most important to work on first. It helps teams focus their efforts and resources on the most valuable or critical aspects of a project.

Imagine you have a limited amount of time and resources to work on a school project. You have several tasks to complete, such as researching, writing, creating visuals, and practicing your presentation. You can't do everything at once, so you need to prioritize.

Let's say your teacher has told you that the most important aspect of the project is the content of your presentation. This means that even if your visuals are not perfect or you don't have much time to practice, as long as your content is strong, you'll do well.
In this scenario, you would prioritize researching and writing your presentation over everything else. Once you have completed the most important task, you can allocate your remaining time and resources to the secondary tasks, like creating visuals and practicing.

In another example, let's say you're a software developer working on a new mobile app. Your team has identified several features that the app should have, such as user registration, a messaging system, and a photo-sharing feature. However, due to time and resource constraints, you can't develop all the features at once.

To prioritize, you may consider factors like market demand and user needs. If your research shows that most users prioritize messaging functionality above everything else, you would focus on developing that feature first. Once it's completed, you can move on to the next most important feature, like user registration, and so on.

By prioritizing tasks or features based on their importance and impact, you can ensure that you're making the best use of your limited time, resources, and effort. It helps you deliver the most valuable aspects of a project early on, increasing the chances of success.

Use Cases

Let's use the example of a smartphone to explain use cases in simpler terms. Communication: A smartphone allows you to make calls, send text messages, and use various messaging apps to stay connected with your friends, family, and colleagues. It's like having a portable telephone that you can carry with you wherever you go.

Internet Access: With a smartphone, you can access the internet anytime and anywhere. This means you can browse websites, search for information, watch videos, and use social media apps like Facebook or Instagram. It's like having a mini computer in your pocket that connects you to the vast world of information and entertainment.

Navigation: Smartphones have built-in GPS capabilities that can help you navigate and find your way around. You can use apps like Google Maps to get directions, find nearby places like restaurants or gas stations, and even track your location in real-time. It's like having a personal guide that helps you reach your destination without getting lost.

Entertainment: Smartphones offer a wide range of entertainment options. You can listen to music, watch movies or TV shows, play games, and even read books using various apps. It's like having a portable entertainment center that keeps you entertained during your free time or when you're traveling.

Productivity: Smartphones come with productivity apps like calendars, notes, and task managers that help you stay organized and manage your daily activities. You can set reminders, create to-do lists, schedule appointments, and even work on documents or presentations on the go. It's like having a personal assistant that helps you stay productive and on top of your tasks.

Photography: Most smartphones today have high-quality cameras that allow you to capture photos and videos. You can use different camera modes, apply filters, and instantly share your moments with others through social media or messaging apps. It's like having a pocket-sized camera that lets you capture memories and share them with the world.

These are just a few examples of the many use cases of a smartphone. It's a versatile device that combines various functionalities to make your life easier, more connected, and more enjoyable.