

# Business Analyst Interview Questions

The Business Analyst Interview Q&A

## Business Analyst Interview Questions and Answers

### 1. How do you define a requirement?

A requirement is the capability possessed by a solution to solve a problem or achieve an objective.

### 2. How do you define the role of a BA in an organization?

A business analyst is a liaison between different stakeholders in an organization. He acts as a bridge, a connector and helps the complete project team work as a tightly integrated unit. Since stakeholders belong to different domains (e.g. finance, business, marketing) it's very important for a business analyst to be able to sort and balance the needs of these stakeholders while fulfilling the business objectives at the same time.

### 3. What is your requirement elicitation strategy?

The elicitation strategy depends upon the type of the project.

One can take advantage of direct collaboration with client and have facilitated workshops, interviews and observe the end users. In conjunction, we can use techniques that provide us with more precise information like prototype and scenario building.

### 4. What are the best practices you follow while writing a use case?

The following are the best practices that are followed to write a clear and well documented use case:

1. Capture both functional and non-functional requirements in a use case.
2. Include use case diagrams along with the use case.
3. Include the UI details/notes in the use case.

### 5. What do you know about scope creep?

Scope creep, also known as requirement creep is a term that denotes uncontrolled changes/deviation in the project's scope without an increase in the other resources (schedule, budget) of the project.

Scope creep is a risk to the project and is usually caused by poor project management, improper documentation of project's requirements and poor communication between the project's stakeholders.

### 6. What are the skills that a business analyst must possess?

A business analyst must possess fundamental skills such as elicitation skills, problem solving skills, communication and management skills. Alongside, he must have knowledge of IT skills,

Software development understanding and domain knowledge regarding the domain he is working in. For more details read [here](#).

#### 7. How do you avoid scope creep?

Scope creep is a hindrance to the project's success and could be avoided by:

- Clearly document the scope of the project.

- Following proper change management.

- Informing the effects of change to the affected parties before making a change.

- Documenting the new requirements in the project log.

- Refrain from adding additional features to the existing functionalities (also called Gold Plating)

#### 8. How do you deal with difficult stakeholders?

Stakeholders sometimes could be difficult to deal with but we could overcome this situation by:

- Patiently listening to them and being polite.

- Make them understand the situation from a prospective they understand.

- Show a commitment to working with them.

- Make them realize how their interests will be realized when they are more open and collaborative.

- Engage them and make them realize that their contribution is valued.

#### 9. When are you done with requirements?

We consider the requirements are complete when:

- They are elicited from all the stakeholders from all the key stakeholders of the project.

- They align with the project's business case.

- When they could be done with the resources available i.e. attainable.

- When the stakeholders of the project are in consensus with the elicited requirements.

- All the requirements which pass the above four criteria, they are considered to be as formal and final. These requirements are then documented and become a part of the project scope.

#### 10. What is the importance of a flow chart?

Simply, flow chart explains the flow of a process through symbols and text. It is important because it:

- It displays information graphically which is both clearer and easy to grasp.

- Helps in process documentation.

- Helps programmers to write the logic.

- Aids testing and troubleshooting.

## 11. What is UML modeling?

UML (Unified Modeling Language) is a general-purpose modeling language, which is designed to provide a standard way to visualize the design of a system.

A modeling language is any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. The rules are used for interpretation of the meaning of components in the structure.

## 12. Why do we use Activity diagram?

Activity diagram is a graphical depiction/flowchart of actions, representing a stepwise listing of activities. We use activity diagrams for the description of those business processes that describe the functionality of the business system.

## 13. What are some of the common tools that a business Analyst uses?

MS Visio, Enterprise Architect, Rational Requisite Pro, MS PowerPoint, MS Word, MS Excel, DOORS, Balsamiq.

You could learn more about these tools [here](#).

## 14. What documents a Business Analyst should deliver?

- Use case documents
- Process/business flow documents
- Requirement traceability matrix document (RTM)
- Functionality matrix (FM)
- Functional requirement specification document (FRS)
- System requirement specification document (SRS)
- Activity/sequence diagrams
- Business requirement document (BRD)

## 15. How do you manage rapidly changing requirements?

Too many changes can be detrimental to the success of the project and hence requirements should be managed carefully. We could do so by following a strict 'Change control' plan, according to which:

- We document when the change was requested, its description and its severity.
- We assess whether the change is in line with the business objective of the project.
- We then analyze the effects of change on the project constraints. We communicate the tentative schedule, cost and resources expenditure to all the stakeholders.

We implement the change only when all the stakeholders are in consensus with the revised project constraints.

16. What are the non-functional requirements?

Nonfunctional requirements or 'qualities' of a system are the requirements that are used to judge the operation of a system. These requirements define how a system is supposed to 'be'.  
E.g.: Throughput, usability, reliability, scalability and security

17. What do you think is better, the Waterfall Model or Spiral Model?

Each project has got different and unique needs and thus the SDLC phases should be chosen based on the specific needs of the project. In brief:

Waterfall model follows a structured approach with each phase having specific deliverables. But, it has little flexibility and adjusting scope later is very difficult.

In spiral model, estimates of project constraints become more realistic as the work progresses and it involves the developers early in the project. But, it takes more time and high cost to reach the final product.

18. What do you know about a misuse case?

A misuse case is inverse of a use case and documents the scenarios that should not happen within the system. The actions depicted in a misuse case can be performed by any person or entity in order to harm the system.

Thus, misuse case are usually used in the field of IT security and data protection.

19. What are the use of configuration management and version control?

Configuration management is everything that you need to manage in terms of a project. This includes software, hardware, tests, documentation, release management, and more.

Configuration management includes, but is not limited to, version control. Version control is saving files and keeping different versions of them, so you can see the change over time.

20. Describe your understanding regarding high level and low level use cases.

The high level use case usually refers to the entire business process whereas when it is divided into smaller units, the outcome or the sub units are what are then referred to as the low level use case.

21. Please explain the use of SDD.

This is the abbreviation of the term System Design Document; it acts as the mediator between business users and the system developers so as the system developers may

understand the business requirements of the system they are developing in order to know where to put emphasis and end up with a quality and objective based system.

## 22. What is Pareto Analysis?

Pareto analysis is a technique which is used to identify the issue that are causing the most number of defects. The issues and their respective defects are plotted in a bar graph and the issue which is causing the highest amount of defect is addressed first.

Pareto analysis is considered as a creative way of looking at causes of problems as it organize data into logical segments for better analysis, comprehension and communication.

## 23. What can you tell us about BPMN?

BPMN stands for Business Process Model and Notation. It's a global standard for graphically representing business process in the form of a diagram.

BPMN contains a set of graphic elements which are used by business users and developers to create activity flows and processes. BPMN's four basic element categories are: Flow objects: Events, activities, gateways

Connecting objects: Sequence flow, message flow, association

Swim lanes : Pool, lane

Artifacts: Data object, group, annotation

## 24. Explain the difference between a task and an activity with respect to BPMN

Activity is a generic term that is used to denote a process/sub process and is a collection of a task or group of tasks whereas a task is a self-contained piece of work.

## 25. Are you aware of JAD?

Joint Application Development (JAD) consists of a structured workshops session between end user/client, project manager, business analyst, technical team and subject matter experts (SME) to facilitate the design and development of the product.

Applications developed through JAD development approach has higher customer satisfaction and less number of errors as the end user is directly involved in the development process.

## 26. Do you know about the term 'force-field analysis'?

Force-field analysis aids in making decisions by identifying the factors for and against a proposed change to the system. The 'for' and 'against' factors are tabulated and are then analyzed, discussed and evaluated for their impact on the change.

## 27. What are Test cases?

A test case is a document which contains listing of all the possible scenarios that could happen based on a respective use case. Thus, every test case is developed with a use case as a base.

A test case contains main flow, positive scenarios, negative scenarios and scenarios covering non-functional requirements also.

A single use case could contain many test cases and these cases are clubbed to make a test script. Test Cases are written in a testing tool like Test Director, but they can be also be written in MS Word. The audience for a test case are the QA testers.

## 28. What are the different testing techniques you use?

The aim of testing is to verify and validate the quality of a developed functionality according to the project requirements. A BA does various types of testing, which are: Black box testing: This is a functional testing where a BA validates that the output generated by the system is as per the requirements/use case

Unit Testing: A BA does unit testing on a developer's machine to make sure the requested functionality is being achieved.

Integration Testing: This type of testing is done when more than one piece of code are integrated to realize a functionality. A BA does integration testing to make sure than the system is performing as expected after different modules are integrated.

Functional Testing: A BA is expected to conduct functional testing to validate that the system is achieving the functionality specified in the use case/functional requirement specification document (FRS).

Acceptance Testing: A BA along with the client, does the acceptance testing to validate that the system is performing as per the business requirements and the product's acceptance criteria.

Regression Testing: Regression testing is done after a modification has been made to the existing system. Its aim is to make sure that all the system functionalities are working as expected.

Beta Testing: A BA along with the testing team, does the beta testing and it is done on a pre-production version of the product. This testing is done to make sure that the functional and non-functional requirements of the system are met.

## 29. Tell me about SaaS

SaaS is Short for Software as a Service and it is a software delivery model under which a software and its associated services are remotely accessed by an end user as a web based service. E.g. Facebook, which is deployed over internet and the users access its services by an internet enabled device.

## 30. What problems a Business Analyst could face during requirements gathering?

Some of the problems faced by a BA during requirements gathering are:

Lack of Clarity in the Scope of the Business requirements

Misalignment of the requirements with the business case of the project

Ill management of Business Requirements

Constantly changing requirements

Unavailability of the key stakeholders

Communication gap between the stakeholders

31. Could you describe the main qualities of a good requirement?

The golden rule to measure the quality of a good requirement is the 'SMART' rule. According to this rule a requirement should be:

Specific: The requirement should be specific so that it could be properly documented

Measurable: We should be able to measure the success criteria of the requirement by different parameters

Attainable: The requirement should be possible to attain with the given resources

Relevant: The requirement should be in line with the project's business case

Timely: The requirement should be posed in time i.e. early in the project life cycle.

32. What are different diagrams that a BA should know about?

There are a couple of different diagrams about which a BA should have concrete knowledge. They are: Entity relationship diagram, data flow diagram, use case diagram, class diagram, activity diagram, sequence diagram, collaboration diagram, component diagrams and deployment diagrams.

33. What are the main responsibilities of a BA?

A business analyst is expected to visualize the 'big picture' and his responsibilities extends towards both the business side as well as the technology side of the project. The major responsibilities that he is expected to fulfill are:

- Ascertain the feasibility of the solution/project/product.

- Analyze, organize and document requirements.

- Liaise and enhance communications with stakeholders.

- Clarify doubts, concerns regarding the solution to be developed.

- Conduct unit testing and verify the development is as per the requirements

- Gain acceptance/approval of the deliverables from the client. Document

- and prioritize change requests from the client.

- Create final product documentations, achieve records and document project lessons learned.

34. What are the different analysis techniques employed by a BA?

The major business analysis techniques used by a BA are: interview, SWOT analysis, facilitated workshop, brainstorming, observation, prototyping and root cause analysis.

35. What is a 100-point method?

The 100-point method is a prioritization method that can be used to prioritize items in a group environment. Each person within the group is given 100 points which they can distribute as votes across the available items.

36. What do you know about 8-omega?

8 Omega is a business change framework to improve the existing business processes. Based on its name, this framework consists of 8 lifecycle phases namely; Discover, Analyze, Design, Integrate, Implement, Manage, Control and Improve. Also, it address 4 key perspectives of business i.e. Strategy, People, Process and Technology.



37. What is FMEA and why it's used?

FMEA stands for 'Failure Mode and Effects Analysis' and it is used for failure analysis, risk analysis and quality engineering.

It involves reviewing components, systems and subsystems on parameters like functional, design and process to identify failure models. The resulting data is then used for risk management and mitigation.

38. What is a use case ?

A use case is a methodology used in requirement analysis to identify, organize and document the requirements. Following are the main characteristics of a use case:

Contains both functional and non-functional requirements      Describes the flow of events/scenarios

Defines the actors involved in the scenarios

Contains main flow, alternative flows and exceptional flows.

Contains business rules and associated diagrams.

Use cases can be used at various stages of a project and its audiences are both technology and business.

39. Tell us the difference between an alternate flow and an exception flow of a use case?

Alternate flow are the alternative actions that can be performed apart for the basic flow and might be considered as an optional flow whereas Exception flow is the path traversed in case of the error or an exception being thrown. For e.g. on a login page the 'Forgot password' is the alternate flow and system showing '404 error' when correct username and password are entered is exception flow.

40. What is the user of trigger in a use case?

Trigger is an action which will invoke a specific flow which would otherwise have been inactive.

41. What all diagrams are used to visualize a use case?

Use Case Diagram, Activity diagram, Sequence diagram, Communication diagram and State machine diagram.

42. Please explain the term Use Case Points

Use Case Points are normalized unit of measurement used to size and estimate the cost of work that is to be done on a system.

43. What is use case generalization and actor generalization?

In the context of use case modelling, sometimes two or more use cases share a common structure and behaviors. When this happens, we create a new use case that describes the shared parts of its parent use cases.

Similarly, actor generalization is the relationship between two actors in a use case where the child actor inherits the properties of a parent actor.

44. What are the advantages of unit testing?

Unit testing is the type of testing which is done at the developer's desk and if a BA conducts unit testing he is able to find a defect before it gets integrated with other codes. This way, a bug gets identified early and is usually fixed in less duration.

45. Elucidate the difference between assumptions and constraints

Assumptions are scenarios that are considered to be as facts while a product is being designed/developed and constraints are restrictions that are imposed and have to be mandatorily followed.

46. Explain Kano analysis

Kano analysis is a quality measurement process aimed at categorizing and prioritizing the customer requirements in an effort to increase the customer's satisfaction.

47. What is a RACI matrix?

RACI matrix is a type of responsibility assignment matrix used to assign roles and responsibilities within the project team. The acronym stands for Responsible, Accountable, Consulted and Informed.

48. How do you identify the basic flow? What would you do if someone was struggling to determine the basic flow for a use case?

Basic flow for use case can be identified from Business Requirement Documents or Functional Requirement Documents as these use cases are prepared on the basis of these requirements.

49. What is the relationship between use case and test case?

A use case is written from a "user" perspective describing the interaction of a piece of software between the user and the software. These are written in common language typically from the business or user point of view and in enough detail for the developer to create a piece of software. Typically written in a MS Word type tool. Use cases capture the functional requirements of the system. It describes the expected interaction the user will experience, in detail. The audience is the business, for signoff, and technology for development.

A Test Case is written using the use cases for a source. It takes a use case to a deeper level so that software testers can exercise every possible scenario that could occur, negative and positive scenarios. One Use Case can turn into 10 test cases. 10 test cases make up a test script. Typically Test Cases will be written in a testing tool like Test Director, but also can be written in MS Word. The audience is QA testers.

50. Good documentation management systems are highly recommended in system development; briefly describe the factors that contribute to a good documentation management system.

For a documentation system to be considered good, the following factors should be prevalent in it: It should be made in such a way that it can accommodate future changes, including version changes, bearing system security features such as providing access only to the allowed users, i.e. have good authentication features. In general, one should take in data as well as information security measures in place, putting in mind that the documentation should also be able to bend to the changing needs of its users as well as the market conditions.

51. Describe the meaning of the term data mapping.

By definition, the term data mapping is the process by which a system developer creates data element mappings that relates two models of data (databases) in order to assist in data integration. This usually assists in the following manner:

- i) Data mediation or transformation between the source and the destination of data
- ii) Assisting in data lineage analysis by identifying the data relationships
- iii) Assists in data masking by discovering sensitive data
- iv) Assists in data de-identification process
- v) Assists in consolidating multiple databases into one thus identification of redundant columns and advising the developers for consideration or even elimination.

52. Describe the meaning of the following words as used in the use case scenario:

- i) Extends
- ii) Includes

In the use case scenario, the term extends is used to imply that a certain action needs to have taken place in order for the other to take place too whereas includes implies that it is not important, as in the action may take place or as well may fail to take place but the other will still take place.

53. What is JAD session?

JAD session:

1 It brings together business area people (users) and IT (Information Technology) professionals in a highly focused workshop.

2 JAD participants typically include:

- o Facilitator – facilitates discussions, enforces rules
- o End users – 3 to 5, attend all sessions
- o Developers – 2 or 3, question for clarity
- o Tie Breaker – senior manager. Breaks end user ties, usually doesn't attend
- o Observers – 2 or 3, do not speak
- o Subject Matter Experts – limited number for understanding business & technology

3 Advantages:

- o Shortening of the time.
- o Improves the quality of the final product by focusing on the up-front portion of the development lifecycle.
- o Reducing the likelihood of errors that are expensive to correct later on.

#### 54. What is GAP analysis?

Gap Analysis:

- 1 The process of determining, documenting, and approving the variance between business requirements and system capabilities.
  - 2 The process of determining and evaluating the variance or distance between two items' properties being compared.
  - 3 The study of the differences between two different systems or applications, often for the purpose of determining how to get from one state to a new state.
  - 4 A gap is sometimes spoken of as "the space between where we are and where we want to be."
- Gap analysis is undertaken as a means of bridging that space.

#### 55. What is a Communication Diagram?

A UML 2.0 Communication Diagram models the objects or parts of a system, the interactions (or messages) between them, and the sequence in which these interactions occur. There are a lot of similarities between communication diagrams and sequence diagrams in terms of the information they show, but because of how each diagram presents the information, one diagram may be better at conveying or emphasizing specific information over the other.

Communication diagrams use a free-form arrangement of objects or parts of a system. This can be compared to how classes and objects are laid out in UML class and object diagrams. Then the interactions between the objects or parts of the system are shown and labeled to indicate the chronological sequence in which they occur. The free-form arrangement of objects lends itself well to showing the sequenced interactions in a more compact space, allowing the analyst to place objects that have the highest number of interactions with each other near one another. This is the advantage of the communication diagram over the sequence diagram. While showing nearly all of the same information as a sequence diagram, the communication diagram can, at a glance, place a strong emphasis on which objects are interacting with one another. While communication diagrams are formally intended to show system objects and the interactions between them, many analysts choose to create them at a higher level of abstraction. Instead of showing the interactions between objects of a system, larger parts of a system may be represented such as the interaction between web methods, web services, or entire systems.

By using the communication diagram in this way, it shows some similarities to a system context diagram. The primary differences between the two are that a system context diagram places a focus on a single system in context along with which actors and systems outside of the scope of the system interact with it. Additionally, a system context diagram does not show the sequence of interactions.

#### 56. Describe the basic classification of requirements as defined by the BABOK.

The requirements classification schema used by the BABOK (v2.0) places requirements in one of the following categories.

- Business Requirements
- Stakeholder Requirements
- Solution Requirements
- Functional Requirements
- Non-Functional Requirements
- Transition Requirements

Business Requirements define the goals and objectives of the business at the enterprise level. These are requirements that apply to the organization as a whole rather than a specific group within the organization. Business requirements are developed and documented as part of ongoing enterprise analysis activities. Business Requirements, or Enterprise Requirements as I prefer to call them, offer everyone within the organization a common understanding of why certain projects are initiated. They are a compass for the organization providing a clear direction that can be followed. While all requirements ideally should define something in measurable terms, this is even more important with business requirements. Therefore, business requirements should outline a corresponding metric and target that needs to be achieved by the business.

Stakeholder Requirements describe the goals and objectives of a particular group within an organization. Like Business Requirements, they are intended to provide a higher level direction for the stakeholder group, but often they are developed while considering the contending goals and objectives of other areas of the organization that interact with each other. So, the stakeholder requirements of various groups need to reflect an overall balance across the organization to support and achieve the Business Requirements in the best possible way. This tighter coupling and dependency between requirements causes Stakeholder Requirements to change more frequently. Stakeholder requirements do not define what needs to be supported by a particular solution (whether the solution is a business process or system solution), rather they span the gap between Business Requirements and more specific Solution Requirements.

Solution Requirements describe the various characteristics of a solution that must be met. The solution may be a process solution or a system solution. Solution requirements should be written in a way that they also support and align with the Stakeholder and Business Requirements. Solution requirements are defined throughout the requirements analysis process. They can be further classified into two sub-categories:

Functional Requirements describe the behavior and information that the solution will manage. In the case of a non-system solution, the behavior typically refers to a workflow and the information refers to the inputs and outputs of the workflow. Additionally, the requirements describe how the data will be transformed and by whom. In the case of a system solution, the functional requirements describe the features and functionality of the system as well as the information that will be created, edited, updated, and deleted by the system.

Non-functional Requirements describe the qualities of the process or system. Instead of describing what the solution must do non-functional requirements describe how well the solution must do something. Non-functional requirements often describe qualities of a process or system such as its repeatability, usability, reliability, interoperability, scalability, extensibility, etc.

Transition Requirements describe any capabilities of the solution that aren't permanent but instead exist only to facilitate the transition from the current state to the future state. Once the process or system has been developed and the transition of users and information from the current solution to the new solution has occurred, these capabilities will no longer be needed or supported. Transition requirements cannot be developed until both the current state and the future solution have been defined.

57. What are the advantages and disadvantages of using screen mockups in the requirements gathering process of a system solution?

Screen mockups can support the requirements gathering process when introduced at the right time, but if introduced too early they can become problematic. Here are a few key points that an analyst should remember.

1) Mockups are nice because they help the business representatives or clients visualize the functionality of the system. This can be a big advantage to help analysts and stakeholders identify problems early on. However, if introduced too soon in the process the natural tendency is for the business reps/clients to try and be screen designers. Instead of stating that the system shall support “x”, they begin saying that they need a dropdown to capture “y” and a button to do “z”. The client is not a UI designer, in fact few business analysts truly are, so this can lead to a screen design which does not have an appropriate emphasis on usability. Similarly, specifying the controls needed on a screen detracts from the true requirements of the system and often results in an inadequate level of discussion around why a system must support certain functionality.

2) When requirements are captured in screen mockups with no supporting requirements list, it becomes impossible to know whether an early screen design decision was made because it supports a necessary requirement or if it was made for some other reason. How can the analyst and developers know whether they can eliminate or alter the screen feature without losing an important requirement. Questions like, “Do we really need to have the control on this screen, or can we capture the data at a later point in the process?” becomes unanswerable without going back to the original stakeholders. And, on complex projects no one stakeholder may be able to answer the question.

3) Screen mockups alone cannot capture the flow through the system. Often analysts will accompany screen mockups with a written description of what happens when certain buttons are clicked or when certain values are entered within a field or dropdown. These descriptions are helpful, but they fall short of describing the end to end processes that the system must support. Further document such as process flows or use cases are required, but often overlooked when too much emphasis is placed on screen mockups during the requirements gathering process. While analysts and stakeholders who are involved in the screen mockup process may have a basic understanding of the processes supported, developers and testers will not.

Ultimately, the introduction of UI mockups can be very helpful, but this should only occur after an exhaustive list of features and usage scenarios (what business process flows need to be supported by the system) have been documented. Only then can the UI mockups be generated without introducing major pitfalls.

58. What is a Context Diagram and what are the benefits of creating one?

The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (systems, organizational groups, external data stores, etc.).

Another name for a Context Diagram is a Context-Level Data-Flow Diagram or a Level-0 Data Flow Diagram. Since a Context Diagram is a specialized version of Data-Flow Diagram, understanding a bit about Data-Flow Diagrams can be helpful.

A Data-Flow Diagram (DFD) is a graphical visualization of the movement of data through an information system. DFDs are one of the three essential components of the structured-systems analysis and design method (SSADM). A DFD is process centric and depicts 4 main components.

Processes (circle)

External Entities (rectangle)

Data Stores (two horizontal, parallel lines or sometimes an ellipse)

Data Flows (curved or straight line with arrowhead indicating flow direction)

Each DFD may show a number of processes with data flowing into and out of each process. If there is a need to show more detail within a particular process, the process is decomposed into a number of smaller processes in a lower level DFD. In this way, the Context Diagram or Context-Level DFD is labeled a "Level-0 DFD" while the next level of decomposition is labeled a "Level-1 DFD", the next is labeled a "Level-2 DFD", and so on.

Context Diagrams and Data-Flow Diagrams were created for systems analysis and design. But like many analysis tools they have been leveraged for other purposes. For example, they can also be leveraged to capture and communicate the interactions and flow of data between business processes. So, they don't have to be restricted to systems analysis.

A sample Context Diagram is shown here.

A Context Diagram (and a DFD for that matter) provides no information about the timing, sequencing, or synchronization of processes such as which processes occur in sequence or in parallel. Therefore it should not be confused with a flowchart or process flow which can show these things.

Some of the benefits of a Context Diagram are:

Shows the scope and boundaries of a system at a glance including the other systems that interface with it

No technical knowledge is assumed or required to understand the diagram

Easy to draw and amend due to its limited notation

Easy to expand by adding different levels of DFDs

Can benefit a wide audience including stakeholders, business analyst, data analysts, developers

59. What are the contents that go into the object model and domain model during GAP analysis. how are the AS-IS and To-BE system documentation prepared?

When performing analysis between an existing system (AS-IS) and its desired future state (TO-BE) you could take into account all aspects/dimensions.

Having said that, I realize that this approach may not always be practical nor necessary. So you should probably determine the focus of your gap analysis by answering the following questions:

- What are the key characteristics of the system?
- What do you know about the existing system?

For example, if your system is workflow centric then it would probably be very beneficial to focus on the differences in workflow and document them by creating a workflow diagram which focuses on the differences.

On the other hand – if the only documentation you have of the existing system is a list of requirements then you should start by identifying the requirement gaps.

So getting back to the domain model...

If you are doing a Domain Model gap analysis focus on and document the following:

- Differences in business entities (new entities, entities no longer needed, etc.)
- Differences in entity relationships (diffs in types of relationships, diffs in multiplicities, etc.)
- Differences in attributes for a given entity (new attributes, attributes no longer needed, etc.)
- Differences in methods/behavior

60. How would you influence people when you do not have decision making authority?

Few business analysts have the final authority to make critical decisions on projects. That's why it's so important for business analysts to polish their influencing skills.

The process used to influence people can be a formal, well thought out presentation, or it can be an informal conversation. Either way, it never hurts to think through a standard framework by which to structure your thoughts before attempting to influence someone. The following is a concise 5-step framework that can be used for both formal and informal communications that involve influencing another person or audience.

- Define the What, Why, and Who

- Prepare your case

- Deliver your message

- Obtain a commitment

- Agree to a specific action plan

These 5 steps provide a framework to structure and plan your communication to maximize your influence over a person or audience, but it's the details of each step that will determine how influential you will be