**BUSINESS ANALYSIS FUNDAMENTALS**

***BUSINESS ANALYSIS IS NOT LIMITED TO ONE ROLE. THE FIELD OF BUSINESS ANALYSIS IS FULL OF DIFFERENT SPECIAL SKILLS ACROSS INDUSTRIES AND TEAMS***

DATA ANALYST

UX ANALYST

PROCESS

ANALYST

REQUIREMENT ANALYST

SYSTEMS ANALYST

BI ANALYST

AUTOMATION ANALYST

PRODUCT ANALYST

**TOP BUSINESS ANALYST SKILLS**

6 KEY SKILLS

ORGANIZATION

PROBLEM SOLVING

COMMUNICATION

SYSTEMS ANALYST

BI ANALYST

PRODUCT ANALYST

**APPROACHING CHANGE**

***ORGANIZATION CAN ADAPT DIFFERENT APPROACHES TO INTRODUCE CHANGE.***

ADAPTIVE APPROACH

PREDICTIVE APPROACH

* Embrace change, delivering small increments frequently.
* Offer flexibility and early value but can be difficult budget and may prolong project completion
* Open for new requirements.
* Plan and execute change in distinct phases
* Aim to foresee and deliver a complete solution
* Struggles with changing requirements
* Gathering all the requirement before moving on

implementation.

COMMON PREDICTIVE FRAMEWORK: **WATERFALL MODEL**

PRODUCT UNDERGOES TESTING

SOLUTION IS CREATED/CONSECUTED

DESIGN/BLUEPRINT CREATED

REQUIEMENTS ARE ANALYZED & DOCUMENTED

TESTING

CREATION

DESIGN

ANALYSIS

FINALIZED SOLUTION IS IMPLEMENTED

IMPLEMENTATION

|  |  |
| --- | --- |
| ADVANTAGE OF WATERFALL FRAMEWORK | |
|  | |
| 1. LINEAR PROGRESSION | It follows a linear progression from idea to implementation, making it easier to track status and identify potential bottlenecks. |
| 1. RESOURCE LOCATION | Waterfall only uses the people needed for each step, which avoids extra work and makes sure teams focus on their own tasks. |
| 1. DEFINED CHECKPOINTS | Each step ends with a checkpoint where the team and sponsors review the work. This makes sure everything is on track and still valuable. |
| 1. PREDICTABLE TIMELINE | Because Waterfall is planned from the start with set schedules, the release date is known ahead of time. This helps organizations prepare for the change. |

LIMITATIONS: Most notably, the Waterfall framework is very rigid and does not allow much flexibility or adaptability to changing requirements.

**WATERFALL** has advantage and limitations. It can be the best framework depending on certain industries and business context.

COMMON PREDICTIVE FRAMEWORK: **V MODEL (VERIFICATION & VALIDATION MODEL)**

ENSURE USER ACCEPTANCE

STAGE 1

STAGE 6

DEFINE REQS

TEST S0LUTION

DESIGN SOLUTION

STAGE 2

STAGE 5

STAGE 3

STAGE 4

TEST MODULES

DEVELOP SOLUTION

* **Define requirements connects to ensure user acceptance,** as it provides the testing criteria and understanding of what satisfying those needs or ultimately what success looks like.
* **Design solution connects to test solution,** providing the system test criteria, confirming all components work together seamlessly.
* **Develop solution connects to test modules,** providing the unit test criteria and understanding of the expected functions.

*The V-model is like an improved version of Waterfall that focuses on checking and testing at every stage.*

*Each development step is matched with a testing step, making sure everything is verified and working properly before moving forward.*

**Popular Agile Frameworks**

Five popular agile frameworks used in business analysis: **Scrum, Kanban, Scrumban, Scaled Agile Framework (SAFe), and DevOps.**

**Scrum Framework**

* Most widely used agile framework (50%+ of organizations).
* Best for complex product development.
* Built on **3 pillars**:
  + **Transparency** – everyone shares the same understanding.
  + **Inspection** – check if progress is on track.
  + **Adaptation** – adjust plans based on feedback.
* Work is divided into short cycles called **sprints**.
* Teams self-organize, meet daily (**daily scrum**) to share updates, and adjust as needed.
* At the end of each sprint, teams deliver value and immediately start the next sprint.

**Kanban Framework**

* Used by ~20% of agile organizations.
* Focuses on **visualizing work** and **limiting work in progress**.
* Uses a **Kanban board** with cards that move across columns (e.g., To Do → In Progress → Done).
* Encourages finishing tasks before starting new ones.
* Flexible and adaptable to different industries.

**Scrumban Framework**

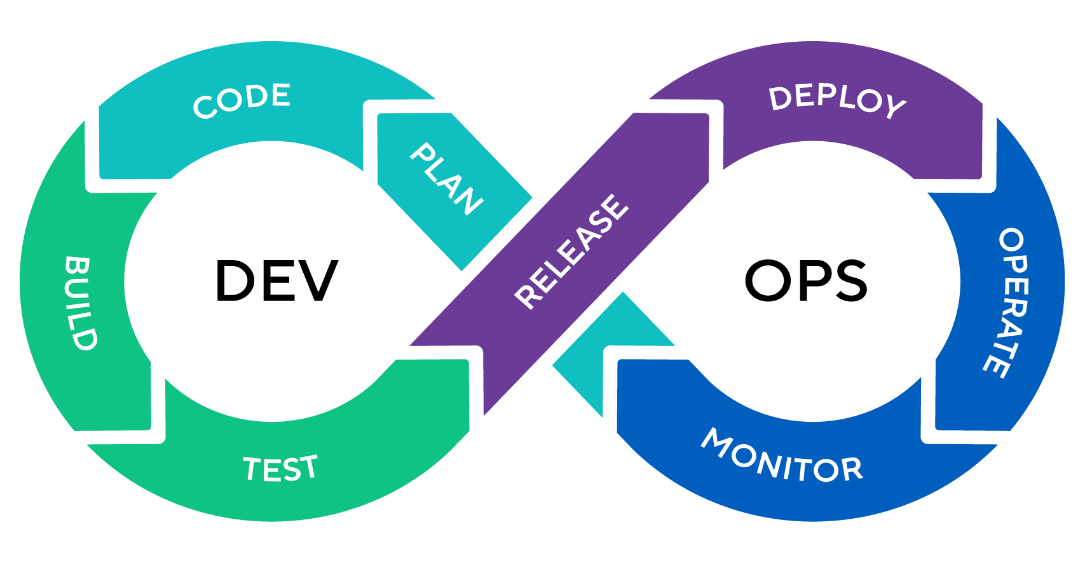
* A **hybrid of Scrum and Kanban**.
* Uses **Scrum events/roles** + **Kanban board visualization**.
* No fixed rules — organizations customize it to their needs.

**Scaled Agile Framework (SAFe)**

* Designed for **large organizations** with many agile teams.
* Builds structure and coordination across multiple teams.
* Uses Scrum, Kanban, or both at team level.
* Critics say it adds too much structure, but when applied correctly, it helps align enterprise-wide goals.

**DevOps Framework**

* Focuses on collaboration between **developers** and **operations teams**.
* Goal: **deliver high-quality software faster and more reliably**.



**DevOps Cycle:**

* 1. Planning – define requirements.
  2. Coding – write the solution.
  3. Building – prepare deployable packages.
  4. Testing – automated checks for quality.
  5. Deployment – release changes to production.
  6. Monitoring – track performance and fix issues quickly.

**Scrum**: Iterative sprints, focus on transparency, inspection, and adaptation.

**Kanban**: Visual boards, limits on work in progress.

**Scrumban**: Flexible mix of Scrum and Kanban.

**SAFe**: Scales agile to enterprise level.

**DevOps**: Combines development + operations, focuses on automation and speed.

**Understanding the Business Objective**

Many business analysts skip over understanding the objective and goals at the start of a project. They dive directly into gathering requirements. It is essential to slow down and first understand the business objective or goal of the project before identifying stakeholders or talking to users.

There are three key questions to ask at the beginning of every project to clarify the objective:

* What is the purpose of the project? Why was it approved and budgeted?
* What are the goals and objectives? Where is the return on investment?
* What does success look like and how will it be measured?

**What is a Stakeholder and How to Identify Them**

**What is a Stakeholder and How to Identify Them**

Stakeholders are people or groups who are directly or indirectly affected by a project. They can influence project success — either positively or negatively. Identifying and engaging them early is key to delivering successful outcomes.

**What is a Stakeholder?**

* Anyone impacted by the project (directly or indirectly).
* Can include team members, customers, suppliers, employees, executives, government agencies, or communities.

**Common Categories of Stakeholders**

* **Project Team Members** – Project manager, business analyst, IT staff, etc.
* **Customers** – Internal or external end users.
* **Suppliers** – Provide goods or services tied to the project.
* **Employees & Executives** – From frontline staff to CEO/CFO.
* **Community / City** – Public stakeholders for community-related projects.
* **Professional Organizations** – Regulators, unions, or industry groups.
* **Individuals Impacted or Support Teams** – Landowners, IT support, designers, etc.

**Why Identify Stakeholders?**

* **More Ideas & Input** – Broader perspectives reduce risks.
* **Varied Perspectives** – Each group sees the project differently.
* **Gaining Buy-In** – Early engagement builds support.
* **Increased Credibility** – Shows careful planning and inclusivity.

**Engaging Stakeholders**

* Communicate project goals.
* Ask stakeholders who else may be impacted.
* Prevents surprises later by uncovering hidden stakeholders early.

**Assigning Responsibilities to Stakeholders using a RACI Matrix**

The RACI Matrix ensures that everyone knows their role and responsibilities throughout the project. It also enables the business analyst to understand all stakeholder roles clearly before the project begins, facilitating better coordination and accountability.

**Definitions of RACI Roles**

* **Responsible**: The person or people who will perform the task and ensure the task is completed.
* **Accountable**: The person who **owns the task** and is ultimately answerable for its success or failure. They approve and validate the work.
* **Consulted**: Stakeholders who provide input, suggestions, or recommendations but do not make decisions.
* **Informed**: Those who need to be kept updated on progress, such as executives or management but don’t take part in the work.

**Requirements Basic**

***One of the most crucial aspects of being a business analyst is knowing what requirements are, how to elicit them, and how to document them.***

A requirement is simply something a product must do or a quality it must have. There are different categories of requirements:

FUNCTIONAL REQUIREMENTS

NON – FUNCTIONAL

REQUIREMENTS

CONSTRAINTS

Functional requirements specify actions the product must take or things it must do. Functional and data requirements specify what the product must do and how data will be manipulated.

Nonfunctional requirements describe how a system should look rather than what is should do.

Limitations or restrictions that affect the design and development process.

Requirements are essential to guide the design and ensure the solution meets business needs.

**SMART Requirements**

Aligned the desired goal with the broader business goal

Set a realistic but challenging goal that is reasonable to achieve

**Specific**

**Measurable**

**R**

**T**

**Relevant**

**Attainable**

**M**

**S**

**A**

**Traceable**

Be as specific as possible with the desired goal.

A requirement must be traceable throughout the project lifecycle.

Set a measurable goal to track and quantify the process

**Specific**

A requirement must be **clear and concise** so that everyone interprets it the same way. It should use consistent terminology and avoid vague words.

To validate specificity, ask yourself:

* What is it doing?
* Why is it doing that?
* Who is doing it?
* Where is it happening?
* Use consistent language across all requirements.
* Avoid terms like *some, several, many*.
* Always include units with numbers (e.g., 7 seconds, not just “within seven”).
* Visuals such as flowcharts or mockups can reduce confusion.

**Measurable**

Requirements must be **quantifiable** so their success can be validated.

Check with questions:

* How much?
* How many?
* How will I know when this is accomplished?
* Define exact metrics (e.g., “response within 7 seconds”).
* Include measurements during requirements gathering, not later in testing.
* Ensure testers can say clearly: *met* or *not met*.

**Attainable**

Check with questions:

* Is there a known solution?
* Can we solve it?
* Has this done before?
* Are there any constraints?

Requirements must be **realistic and feasible** given the project’s scope, budget, time, and technical ability.

* Rule out “wish list” or impossible ideas.
* Validate feasibility with experts when in doubt.
* Consider constraints like environment, location, or system limits.

**Reasonable**

Even if a requirement is attainable, it must also be **worth the effort**.

Check with questions:

* Is the worthwhile
* Is the timing right?
* Does this match other efforts and needs?
* Compare cost vs. benefit — does the value justify the effort?
* Ensure the timing is right (avoid duplication with other projects).
* Align with business priorities and current needs.

**Traceable**

Requirements should be **trackable across the project lifecycle**, from design to implementation to testing.

Check with questions:

* Can I ensure this requirement has been met in the design solution?
* Can I trace it to the implementation
* Can I trace it to the testing.
* Connect each requirement back to its origin and business justification.
* Note any dependencies, assumptions, or priorities.
* Ensure every requirement can be validated through testing.

Traceable means every requirement has a clear link from start to finish, so the team can prove it was addressed correctly in the final product.

**Phases of the Requirements Process**

This phase is primarily about extracting requirements from various sources such as documentation and stakeholders. Techniques include interviewing, surveys, requirement workshops, and others.

**REQUIREMENT ELICITATION**

Requirements Approval involves obtaining business, technical, and executive sponsor or steering committee approvals.

This phase focuses on categorizing requirements by determining which are functional, non-functional, supplemental, or constraints. It also involves setting attributes and priorities for the identified requirements.

This phase ensures that all requirements are complete. It involves modeling the requirements in a manner that both business and technical users interpret consistently. The analysis phase is crucial to make the requirements readable and to eliminate all ambiguity.

**REQUIREMENT ANALYSIS**

**REQUIREMENT APPROVAL**

**REQUIREMENT SPECIFICATION**

**BUSINESS RULES**

A **business rule** is a clear instruction that tells a business what it must or must not do, always resulting in either true or false.

For example, “Every course must have at least one instructor” or “A quote must be created before an invoice.” Business requirements are then written to support these rules, such as making sure the system allows assigning instructors or linking quotes to invoices.

Business rules should be kept simple, written with their supporting requirements, and stable so they don’t change often because changing them later can cause big problems for the project.

**Requirement Elicitation**

**Elicitation** is the process of gathering information from stakeholders or other sources to understand their needs and requirements.

**Elicitation Techniques**

Some common ways to conduct elicitation include:

* **Interviews** – Asking stakeholders targeted questions.
* **Observation** – Watching how people work to uncover hidden needs.
* **Brainstorming** – Generating ideas collaboratively with a group.
* **Workshops** – Structured sessions to refine and prioritize requirements.
* **Surveys/Questionnaires** – Gathering input from a larger audience.

**Planned vs. Unplanned Elicitation**

* **Planned:** Sessions are scheduled and prepared in advance. For example, before an interview you define the goal and write down the questions.
* **Unplanned:** Happens spontaneously. For instance, in a meeting unrelated to your project, you might hear information that affects your work and ask follow-up questions.

**Why it’s important?** Strong elicitation skills allow analysts to uncover the right requirements, reduce misunderstandings, and ensure the final solution aligns with business needs.

**Brainstorming as an Elicitation Technique**

**What is Brainstorming?**

Brainstorming is a collaborative way to generate ideas, solve problems, or build consensus. It is one of the most common elicitation techniques used by business analysts to quickly gather perspectives from multiple stakeholders.

**Types of Brainstorming**

1. Individual – One person generates ideas (rarely used, but useful for unique roles).
2. Open – Most common; group discussion where ideas build on one another.
3. Structured – Each participant shares ideas one by one, ensuring equal input.

**Purposes of Brainstorming:**

* Solve problems at any project stage (early, mid, or late).
* Generate new ideas or creative solutions.
* Build consensus by encouraging collaboration.

|  | **Open Brainstorming** | **Structured Brainstorming** |
| --- | --- | --- |
| **Format** | Free-flowing discussion where participants share ideas openly. | Each participant shares ideas one by one in rounds. |
| **Participation** | May be uneven — strong voices dominate, quieter participants may hold back. | Equal participation — everyone gets a turn. |
| **Energy Level** | High-energy, fast-paced, ideas build off one another. | Slower-paced, more controlled and balanced. |
| **Best Used When** | Team is comfortable sharing openly and collaboration is strong. | Sensitive topics, presence of executives, or strong personalities in the room. |
| **Advantages** | - Generates lots of ideas quickly.  - Builds momentum and excitement. | - Ensures all voices are heard.  - Reduces dominance from louder participants. |
| **Disadvantages** | - Risk of going off-topic.  - Some voices may be overshadowed. | - Less dynamic and energetic.  - Takes more time to collect ideas. |

**Key Roles**

* Facilitator – Guides discussion (often the BA).
* Scribe – Documents all ideas (preferably digitally, visible to group).
* Timekeeper – Ensures session stays within schedule.

**Managing the Session**

* Limit brainstorming to 30–45 minutes per session to avoid fatigue.
* Use multiple shorter sessions for complex topics.
* Group and categorize ideas before wrapping up.
* Consider voting/prioritization (e.g., each participant gets 3 votes) to identify top ideas.

| **Type** | **Description** | **When to Use** | **Strengths** | **Challenges** |
| --- | --- | --- | --- | --- |
| **Formal Requirement Workshop** | Highly structured session with strict agenda and selected stakeholders. | When requirements must be clearly defined, refined, and finalized. | • Produces documented, approved requirements  • Ensures accuracy & validation | • Needs strong facilitation  • Can be time-consuming |
| **Business Process Improvement (BPI) Workshop** | |  | | --- | | Looks at the current process (“as-is”) and designs an improved process (“to-be”), often using Post-it notes to map steps. |  |  | | --- | |  | | When processes need redesign or improvement. | • Visualizes workflows clearly  • Identifies bottlenecks and inefficiencies | • May surface cross-department conflicts  • Requires more time for mapping |
| **Agile Requirement Workshop** | Informal, conversation-driven workshop emphasizing backlog & scope over documentation. | At the start of Agile projects to define backlog and sprint goals. | • Encourages collaboration & adaptability  • Faster and lighter approach | • Less documentation  •Risk of missing details if not followed up |

**Elicitation Technique: Requirement Workshops**

Requirement workshops are organized meetings where different groups like end users, experts, managers, decision-makers, and IT come together to discuss and agree on project requirements.

Instead of collecting input from each group separately and trying to combine it later, everyone talks in real time, clears up confusion, and reaches agreement on the spot.

*Requirement workshops are very useful for big projects that involve many different teams. They take careful planning and a good facilitator, but they help everyone work together in a structured way to collect clear requirements, agree on decisions, and build a strong starting point for the project’s success.*

**Introduction to Interviewing in Requirement Elicitation**

**What is Interviewing?**

Interviewing is a discussion technique used to extract requirements quickly and document the ideas and thoughts of stakeholders.

**Benefits of Interviewing**

* Reduces misunderstandings by enabling direct communication, often face to face.
* Allows clarification of vocabulary or acronyms that might be confusing.
* Encourages stakeholders to ask questions comfortably, which might not happen in larger group settings.
* Can be conducted one-on-one or in small groups, typically no more than two people to maintain balanced conversation.

|  | **Formal Interviews** | **Informal Interviews** |
| --- | --- | --- |
| **Structure** | **Highly structured, follows a set agenda and protocol.** | **Flexible and conversational, not strict.** |
| **Purpose** | **To document official requirements, align with strategic goals, and maintain accountability.** | **To gather honest feedback, uncover issues, and capture practical day-to-day needs.** |

**Understanding Stakeholder Perspectives**

Interviews allow you to see processes and requirements from the interviewee's perspective, helping to identify what works, what doesn't, and potential improvements for current or future systems.

**Types of Interviews**

| **Type** | **What It Is** | **Best For** | **Strengths** |
| --- | --- | --- | --- |
| **Personal Interview** | Asking scripted, mostly open-ended questions to multiple stakeholders in the same group. | Validating responses and spotting conflicts. | * Consistent questions across users * Clearer comparisons |
| **Job Shadowing** | Observing a user or group performing daily tasks in real time. | Understanding real workflows and hidden challenges. | * Reveals issues not mentioned in interviews * Provides real context |
| **Customer Site Visit** | Visiting the customer’s environment to see real conditions and workflows. | When consultants or external analysts need context. | * Shows environmental factors * Builds stronger client understanding |
| **Task Analysis** | Users explain their tasks step by step, often away from normal work. | Identifying inefficiencies and process improvements. | * Deep dive into processes * Exposes bottlenecks |

**Advantages and Disadvantages of Interviewing**

| **Advantages** | **Disadvantages** |
| --- | --- |
| Builds rapport and strengthens relationships through interactive discussions. | Requires stakeholders’ time and willingness to participate. |
| Helps identify conflicts or gaps in requirements. | Preparing good, scripted questions can be time-consuming. |
| Encourages participation and reveals nonverbal cues (e.g., discomfort, hesitation). | Stakeholders may focus only on current processes, not future needs. |
| Allows immediate follow-up for clarification and better understanding. | Documentation may be misinterpreted by the interviewer or readers. |

**Interviewing Best Practices**

|  | **Best Practices** |
| --- | --- |
| **Choosing the Type of Interview** | - Select the interview type based on the problem or challenge.  - Options: individual, group, job shadowing, client site visit, task analysis.  - Don’t proceed without knowing the right type. |
| **Preparing for the Interview** | - Prepare structured, scripted questions.  - Avoid “winging it”—looks unprofessional and risks missing details.  - Good prep reduces the need for re-interviews. |
| **Scheduling & Punctuality** | - Schedule in advance so stakeholders can plan.  - Arrive 5–15 minutes early to review questions and background.  - Being late = disrespect, hurts rapport. |
| **Matching the Interviewee’s Pace** | - Adjust your speaking speed to theirs.  - Fast talkers cover more quickly; cautious speakers need more time.  - Matching pace improves communication. |
| **Checking Understanding** | - Summarize and repeat back key points.  - Confirm accuracy often.  - Saves time by avoiding misunderstandings later. |
| **Informing Interviewees About Use** | - Be clear on why information is gathered.  - Reinforce they are the experts on processes.  - Show how input will be captured in requirements/design. |
| **Scribe or Recording** | - Use a scribe if possible.  - Otherwise, record with permission.  - Ensures accuracy and helps review later. |
| **Examples & Screenshots** | - Ask for examples/screenshots of workflows, system screens, or errors.  - Provides clarity and context. |
| **Interviewing Multiple People & End-Users** | - Interview 2–3 per role for diverse views.  - Always include end-users (not just managers).  - Compare perspectives to spot gaps. |
| **Gratitude & Follow-Up** | - Thank participants in invitations and after sessions.  - Send a follow-up email showing how input will be used.  - Builds trust and engagement. |
| **Debrief Time** | - Don’t schedule back-to-back.  - Allow 15 minutes after each interview for notes.  - Prevents detail loss and handles overruns. |

**Elicitation Technique: Surveys**

**What is a Survey?**

A survey traditionally consists of questions asked to stakeholders to quantify their thoughts. For example, "Do you like the current system? Rate from one to five." These questions are typically closed-ended and presented in a document or online.

**Type of Survey Questions**

| **Type** | **Description** | **Strengths** | **Challenges** |
| --- | --- | --- | --- |
| **Open-Ended** | Free-text responses (e.g., “What do you think about the current system?”) | Rich, detailed feedback | Hard to quantify, requires manual/algorithmic interpretation |
| **Closed-Ended** | Predefined answers (e.g., rating scales, multiple choice) | Easy to analyze, scalable, statistical insights | Limited depth, may miss unique perspectives |

**Types of Closed-ended Questions**

Common closed-ended question formats include:

* **Rankings:** Asking respondents to rank items by importance, frequency of use, or pain points.
* **Multiple choice:** Selecting the best option from a list of tasks or preferences.

**Pros and Cons of Survey**

| **Advantages** | **Disadvantages** |
| --- | --- |
| * Surveys require limited time from stakeholders, often only a few minutes. * Scalable across locations/time zones * Cost-effective with online tools * Validates assumptions from other techniques | * Surveys often have low response rates unless supported by management. * Poorly worded or leading questions can invalidate results. * Open-ended questions require more complex analysis. * Administering surveys requires training on survey tools and domain knowledge to craft effective questions. |

**Elicitation Technique: Document Review**

**What is Documentation Review?**

Documentation review involves examining existing documentation. This can include user guides, documentation from previous projects or implementations, lessons learned, and records of what went wrong or went well.

**When to Use Documentation Review**

Use documentation review early in the project and again about midway through requirements solicitation. This allows you to examine what currently exists, validate it against user input, and make informed determinations based on the requirements gathered so far.

**Advantages and Disadvantages**

| **Advantages** | **Disadvantages** |
| --- | --- |
| Provides early context before engaging stakeholders | May be outdated or inaccurate |
| Helps understand business operations & systems | Requires domain knowledge to interpret correctly |
| Saves time in framing discussions | Can be time-consuming if too much documentation |
| Useful for building glossaries of acronyms & terms | Risk of being misled if relied on exclusively |

**Requirement Analysis**

**Purpose of Requirements Analysis**

Requirements analysis involves stating requirements in multiple ways to accomplish three main objectives:

* First, to ensure all stakeholders understand the requirement.
* Second, to enable the business or customer to prioritize requirements by understanding the need and purpose of each requirement.
* Third, to allow the design and development teams to design and implement solutions that meet those requirements.

**Visual Modeling Concepts**

**Introduction to Visual Modeling**

Visual modeling is a technique used to simplify complex information by representing it graphically. Instead of relying solely on blocks of text, visual modeling transforms data into diagrams, process flows, or matrices, making it easier to understand and analyze.

**Benefits of Visual Modeling**

| **Benefit** | **Simplified Summary** |
| --- | --- |
| **Easier Understanding of Complex Information** | Visual models make complicated ideas easier to understand than long text. |
| **Stakeholder Involvement** | Visual tools help engage stakeholders during discussions and workshops. |
| **Efficient Requirements Elicitation** | Modeling speeds up and clarifies the process of gathering requirements. |
| **Identification of Underlying Problems** | Diagrams can reveal hidden issues that text might miss. |
| **Analysis of What-If Scenarios** | Models help explore different situations to ensure all possibilities are considered. |

**Business Models**

A screenshot of a computer

AI-generated content may be incorrect. **Organizational Chart**

An organizational chart is a visual model that shows the hierarchical structure of a company or organization. It clearly outlines the chain of command, displaying who is in charge and the reporting relationships between different departments and individuals.

A screenshot of a computer

AI-generated content may be incorrect. **Competitive Comparison Matrix**

The competitive comparison matrix is a matrix that breaks down different products, services, or companies based on various requirements or attributes important to that product, service, or company. This visual helps determine which product, company, or service best meets your needs.

A diagram of a dough-licious system

AI-generated content may be incorrect. **Stakeholder Map**

A stakeholder map is a visual tool used to organize and understand the people and groups involved in a project. This model helps a business analyst or project manager easily see where everyone fits, which is crucial for activities like gathering requirements.

**Use Case Diagram**

A diagram of a person's process

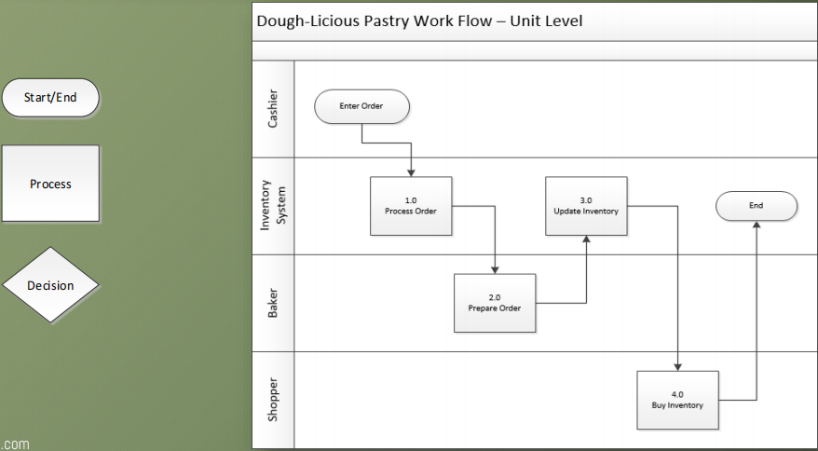
AI-generated content may be incorrect.

A use case diagram is a visual model that shows how different users (called actors) interact with a system and its functions (called use cases).

It helps to clarify who will use the system and for what purpose.

**Process Flow Diagram**

A Process Flow Diagram is a visual representation of a business process or system. It uses standard symbols to show the sequence of steps, decisions, and actions, making a complex process easy to understand.



**Technical Models**

Technical models are generally used less by business analysts and more by IT business analysts, designers, and developers.

**System Context Diagram**

A diagram of a patient

AI-generated content may be incorrect.

The system context diagram is a high-level diagram illustrating a system and the people or systems interacting with it, along with the nature of those interactions.

**Data Flow Diagram (DFD)**

A diagram of a company

AI-generated content may be incorrect.

A Data Flow Diagram is a visual model that shows how data moves through a system or a business process. Unlike a process flow diagram, which focuses on the sequence of steps and decisions, a DFD specifically tracks the flow of information from its origin to its destination.

**State Diagram**

**A diagram of a payment system

AI-generated content may be incorrect.**

A State Diagram is a visual model that shows the different states or conditions a system or object can be in. It also illustrates the events that cause it to move from one state to another, which are called transitions.

**A diagram of a computer

AI-generated content may be incorrect.Entity Relationship Diagram (ERD)**

An Entity-Relationship Diagram (ERD) is a visual model that shows how different pieces of information, or entities, relate to each other within a system. It's like a blueprint for a database.

**BPM VS UML**

**A diagram of a computer program

AI-generated content may be incorrect.**

**A diagram of a cash flow

AI-generated content may be incorrect.**

* **UML**, the older standard, is primarily for **software engineering**. It's used to model the structure and behavior of software systems and is best suited for technical audiences like developers and architects.
* **BPMN** is specifically for **business process modeling**.Its symbols and structure are designed to be easily understood by both business and technical stakeholders, focusing on the flow of activities and events within a process.

**Key Differences**

|  | **BPMN** | **UML (specifically, Activity Diagram)** |
| --- | --- | --- |
| **Primary Focus** | Business processes & workflows | Software system design & behavior |
| **Target Audience** | Business Analysts, managers, all stakeholders | Software developers, system architects |
| **Readability** | High, uses intuitive symbols like intermediate events | Lower, more complex for non-technical users |
| **Complexity** | Excellent at modeling complex business logic, including exceptions and parallel flows. | Can become confusing when modeling complex, multi-path business logic. |
| **Recommendation** | Learn this if you're a business analyst, as it's the current industry standard. | Learn this if you're a software developer or solution architect. |

While they can model similar concepts like activities, events, and gateways, **BPMN is generally recommended for business analysts** because its notation is specifically designed for business processes, making it easier to communicate with a wide range of stakeholders.

‘

**Requirement Specification**

**CATEGORIZING REQUIREMENT**

**ASSIGNING REQUIREMENT**

**DERIVING REQUIREMENT**

Attributes are additional details tied to each requirement that support **clarification** (adding context like origin or importance), **filtering** (organizing by type, priority, or urgency), and **validation** (ensuring requirements meet business goals).

Common attributes include **unique identifier, acceptance criteria, author, complexity, ownership, performance, urgency, business value, status, type, priority, and source.** Among these, the six most important to assign early are **unique identifier, acceptance criteria, complexity, business value, status, and source.**

Attributes are usually tracked in tools like spreadsheets and later included in requirements documentation. While all attributes are useful, type and priority are often determined later in the process.

When you're organizing project requirements, you should sort them into three groups: **functional**, **non-functional**, and **constraints**.

Functional requirements define what the system must do, while non-functional ones describe its qualities, such as speed or performance. This process ensures all essential details are considered before you begin building the system.

Deriving requirements is the process of adding clarity, detail, and removing ambiguity from requirements you have already gathered. This is done using four key techniques:

* **Parsing:** Breaking down a broad or complex requirement into multiple, more specific ones, especially those with the word "and."
* **Interpreting:** Clarifying vague or subjective terms to make the requirement precise and unambiguous.
* **Focusing:** Combining overlapping or redundant requirements into a single, concise statement.
* **Qualifying:** Adding a measurable and verifiable condition to a requirement to ensure it can be tested.

**VALIDATING REQUIREMENT**

**PRIORITIZING REQUIREMENT**

Prioritizing requirements is essential because every project has more desired features than available time, budget, and resources.

**Key Factors for Prioritization**

* **Value to the business** – benefits the company will gain.
* **Value to the customer** – impact on the end user.
* **Minimize cost to develop** – choose cost-efficient solutions.
* **Time to implement** – how quickly benefits can be realized.
* **Ease of technical implementation** – feasibility and complexity.
* **Ease of business implementation** – how smoothly the organization can adopt it.
* **Obligation to external authority** – compliance with laws or regulations.

**Three-Step Process**

1. **Define Usefulness to the Business**
   * *Critical*: Must have; without it, the project cannot proceed.
   * *Important*: Valuable but not essential for project continuation.
   * *Nice to have*: Enhancements, not mandatory.
2. **Estimate Cost**
   * Rate requirements on a 1–5 scale (low to high).
   * Example for a 6-week project:
     + 1 = <4 hours
     + 2 = 4–8 hours
     + 3 = 9–20 hours
3. **Estimate Timeframe**
   * Use a 1–5 scale.
   * 1 = shortest timeframe; 5 = longest timeframe.

Combining usefulness, cost, and timeframe helps determine the top priorities: critical, low-cost, quick-to-implement requirements should be handled first.

Validating requirements is different from approving them.

* **Approval** = obtaining consent from stakeholders and technical teams (done in the next phase).
* **Validation** = checking requirements to ensure they are accurate, useful, and implementable.

**How to Validate Requirements**  
Use the **SMART technique**:

* **Specific** – clearly defined and unambiguous
* **Measurable** – can be tested or verified
* **Attainable** – realistic to achieve
* **Reasonable** – aligned with business goals and constraints
* **Traceable** – linked back to its source and business value

**Additional Tips**

* Requirements should always be **clear, concise, and free from ambiguity**.
* Review earlier resources (Requirement Basics, SMART Requirements lecture, and validation tips) to ensure consistency.

**BUSINESS REQUIREMENTS DOCUMENT**

**Business Requirements Document (BRD)**

The BRD is a formal document that contains all project requirements, business rules, use cases, version history, and stakeholder information. It serves as the central source of truth for requirements and becomes the foundation for later phases of the project.

**Responsibility for Preparing the BRD**

* The business analyst is responsible for creating and maintaining the BRD.
* It must be complete, accurate, and approved before moving to subsequent phases.

**Purpose and Uses of the BRD**

* Stores all requirements for the project.
* Lists and documents all stakeholders.
* Used to obtain stakeholder approval of requirements.
* Guides solution design to meet business needs.
* Sets the foundation for testing.
* Acts as the groundwork and foundation of the entire project.

**When to Fill Out the BRD**

* Filled out mainly during the **analysis** and **specification** phases.
* During **elicitation**, requirements are documented informally.
* During **validation**, approvals on the BRD are obtained.

**Key Sections of a BRD**

| **Section** | **Description** |
| --- | --- |
| **Cover Page** | Title, project information, version number, and last modified date. |
| **Version & Approvals** | Tracks changes, modifications, and approvals by stakeholders. |
| **Table of Contents** | Provides organized navigation through the document. |
| **Project Details** | Overall project information, often sourced from the project charter. |
| **Overview** | States the purpose of the BRD, its use, and project summary. |
| **Document Resources** | Lists stakeholders and resources, may include a stakeholder map. |
| **Glossary of Terms** | Defines acronyms and key terms for common understanding. |
| **Project Overview** | Outlines dependencies and provides a stakeholder list. |
| **Key Assumptions & Constraints** | Identifies key assumptions and constraints relevant to the project. |
| **Use Cases (Optional)** | Visual models describing user and system interactions. |
| **Business Requirements** | Core section: includes priorities, descriptions, and rationale for requirements. |
| **Appendices** | Contains process flows, diagrams, business rules, and other supporting models. |

**Requirement Approval**

**Three Main Steps to Gaining Approval**

There are three main steps to gaining approval:

1. **Business Team Approval**: This involves validating that all documented requirements are accurate and complete according to the business team's needs.
2. **Technical Team Validation**: The technical team must confirm they understand the requirements and believe they can design a solution that is technically feasible and compatible with the known architecture.
3. **Project Sponsor Approval**: The project sponsors or committee, who provide funding, must approve the high-level project, design, and solution to move forward officially.

**Gaining the Business Team Approval**

Gaining business approval ensures that requirements are accurate, aligned with business needs, and free of missing or unnecessary items. This step confirms the requirements before moving forward with design, development, and testing.

| **Section** | **Details** |
| --- | --- |
| **Scheduling the Approval Process** | Hold multiple short review sessions. If many business units are involved, schedule separate sessions to keep discussions focused. |
| **Session Duration & Engagement** | Keep sessions under 4 hours (best: 2.5 hours). The first 20–30 minutes are often less productive, so shorter sessions work better. |
| **Inclusion of SMEs** | Invite subject matter experts (SMEs) who helped define requirements. Do not add new people who were not part of the earlier process. |
| **Keeping Sessions Relevant** | Focus on what matters to each group. Example: managers care about reports, while users care about processes. |
| **Preparing a Meeting Agenda** | Make a simple agenda with topics and time slots. This helps participants know what to expect and when breaks will happen. |
| **Conducting the Approval Meeting** | Start with the purpose and agenda. Do introductions if needed. Remind everyone of the project goals, then review requirements step by step for approval. |
| **Reviewing Requirements** | Go through requirements one by one. Explain clearly, answer questions right away, and record any needed updates. |
| **Managing Changes & Updates** | Update requirements during the session if possible. Only add new requirements if they are critical—otherwise, plan a separate follow-up. |
| **Setting Expectations** | At the start of the meeting, clarify that the purpose is to approve existing requirements, not to define new ones. |

**Gaining the Technical Team Approval**

| Details | |
| --- | --- |
| Scheduling Sessions | Two sessions:  1) High-Level Review (early stage, check feasibility and make/buy decisions),  2) In-Depth Review (detailed validation with developers). | |
| Participants | Include technical SMEs such as IT business analysts, technical leads, developers, UX designers, and architects. | |

| **Aspect** | **Session One: High-Level Review** | **Session Two: In-Depth Technical Approval** |
| --- | --- | --- |
| Purpose | Provide business context, check overall feasibility, and prepare team for deeper review. | Validate technical feasibility in detail and confirm readiness for design phase. |
| Agenda | Explain purpose, review project objectives, touch on requirements, and highlight critical components. | Reiterate purpose, review objectives briefly, and go through requirements one by one. |
| Focus | Identify any major concerns not addressed during analysis/specification. | Flag troublesome/high-cost requirements, confirm sufficient detail for design, and finalize decisions. |
| Engagement | Reengage technical team by answering questions and clarifying early concerns. | Validate alignment, resolve technical challenges, and confirm feasibility. |
| Output | Provide all requirements for further review; team prepares for detailed session. | Finalize make/buy decisions, update cost estimates, and grant technical approval. |

**Gaining Sponsor or Committee Approval**

The sponsor approval meeting is the final step before moving the project into the design phase. The goal is to confirm project alignment, validate business value, and gain official sign-off.

**Key Steps in Preparing for Sponsor Approval**

1. Prepare the Presentation

* Create a high-level presentation (not detailed BRD).
* Focus on business value and objectives rather than technical details.
* Include visuals (charts, graphs, summaries) to make information easy to grasp.

2. Update Project Information

* Revise project schedule, costs, and risks based on finalized requirements.
* Ensure timelines and risk assessments reflect the current state.

3. Summarize Business Requirements

* Provide a high-level overview of business requirements.
* Highlight sections and purpose rather than listing every detail.

4. Define the Recommended Solution

* State whether the solution will be built or bought.
* Present the major components of the recommended approach.

5. Explain the Transition Plan

* Outline the business transition:
  + Training needs
  + Policy or process changes
  + Job aids and support for end-users

6. Schedule the Meeting

* Invite only essential attendees (sponsor, technical lead, business lead, possibly one SME).
* Avoid inviting the entire project team.

7. Create & Share an Agenda

* Send the agenda in advance with duration and topics.
* Allow the sponsor to request changes before the meeting.

Conducting the Approval Meeting

1. Start with the purpose and agenda.
2. Review the project objectives and reinforce business value.
3. Deliver the high-level presentation:
   * Focus on benefits and business outcomes.
   * Keep technical details minimal (except when discussing risks).
4. Address questions and concerns immediately with support from key leads.
5. Gain official sign-off from the sponsor as the stage gate to proceed.

Meeting Duration & Considerations

* Aim for 1 hour (can extend to 1.5–2 hours for larger projects).
* Respect the sponsor’s limited time (executives and senior managers).
* Ensure enough time for Q&A to strengthen approval confidence.

**Project Review**

A project review is conducted to bring together the project team and stakeholders to reflect on the project, discussing both what went well and what did not. The review ensures lessons are captured and improvements identified for future projects.

Best Practices for Project Reviews

There are three best practices for conducting project reviews:

* **Multiple Meetings**: While one meeting is preferred, large groups (50, 60, or 100+ people) may require splitting into multiple meetings to ensure everyone can contribute.
* **Core Project Team**: Ensure the core project team meets with different stakeholder groups separately if needed, such as departments or upper management.
* **Pre-Meeting Survey**: Send out a survey ahead of time to gather feedback, so the meeting is not conducted blindly and to facilitate discussion.

Sending the survey immediately after project completion captures immediate lessons learned, as waiting too long may result in faded memories and reduced engagement.

Without reviewing what went well and what did not at the end of each project, mistakes will recur, causing frustration and inefficiency.

*Take time to review, analyze, and create an action plan to execute improvements, thereby increasing the success of future projects.*