IT Project Guidance -

On Business Requirements Development

Version:

0.1

## Purpose

To provide structured guidance for developing business requirements that inform and precede user requirements in IT projects. This document supports the identification of drivers, motivations, and expected outcomes that justify system change.

## Audience

This guidance is intended for architects, analysts, designers, and governance leads responsible for defining the rationale and scope of system change. It is also relevant to stakeholders validating the strategic intent behind proposed solutions.

## Synopsis

Business requirements precede user requirements and stakeholder mapping. They define the "why"—the motivations and drivers for change—before the "who" and "what" are addressed. Together, these form the foundation for user requirements, which are covered in a separate document: *IT Project Guidance - On User Requirement Development*.

## Contents

[Purpose 1](#_Toc202446130)

[Audience 1](#_Toc202446131)

[Synopsis 1](#_Toc202446132)

[Contents 2](#_Toc202446133)

[Introduction 3](#_Toc202446134)

[What Business Requirements Are 3](#_Toc202446135)

[Business Requirement Vectors 3](#_Toc202446136)

[Registry & Structural Vectors 3](#_Toc202446137)

[Resource & Cost Vectors 4](#_Toc202446138)

[Integration & Interface Vectors 4](#_Toc202446139)

[Market & Value Vectors 4](#_Toc202446140)

[Reputation & Strategic Vectors 4](#_Toc202446141)

[How to Elicit and Classify Business Requirements 4](#_Toc202446142)

[Start with External Drivers: PESTELE 5](#_Toc202446143)

[Assess Internal Capabilities: Capability Mapping 5](#_Toc202446144)

[Use CLEAR to Guide the Engagement Process 5](#_Toc202446145)

[Language Use and Misunderstanding Avoidance 6](#_Toc202446146)

[SMART Business Requirements 6](#_Toc202446147)

[On Embedding MoSCoW Early 6](#_Toc202446148)

[Avoid “All” 7](#_Toc202446149)

[Avoid “Shall” and “Will” 7](#_Toc202446150)

[Use “And” Carefully 7](#_Toc202446151)

[Use “If” Sparingly 8](#_Toc202446152)

[Express Scale Expectations Where Possible 8](#_Toc202446153)

[Business Requirement Statement Template 8](#_Toc202446154)

[Conclusion 9](#_Toc202446155)

[Appendices 10](#_Toc202446156)

[Appendix A - Document Information 10](#_Toc202446157)

[Versions 10](#_Toc202446158)

[Images 10](#_Toc202446159)

[Tables 10](#_Toc202446160)

[References 10](#_Toc202446161)

[Review Distribution 10](#_Toc202446162)

[Audience 10](#_Toc202446163)

[Structure 10](#_Toc202446164)

[Diagrams 11](#_Toc202446165)

[Acronyms 11](#_Toc202446166)

[Terms 11](#_Toc202446167)

# Introduction

Business requirements are the first formal articulation of the rationale for change. They precede both stakeholder mapping and user requirements.

While stakeholder maps clarify roles and responsibilities, and user requirements describe what users must do, business requirements define the outcomes that justify the system’s existence, tying them back to drivers.

This document is part of a broader suite of guidance papers. For the next phase—developing user requirements—refer to *IT Project Guidance - On User Requirement Development*.

# How to Use This Document

This guidance is designed to support practitioners in developing clear, actionable business requirements for IT projects. It can be read end-to-end or used selectively depending on the reader’s role and needs. Analysts may focus on elicitation techniques and language guidance, while governance leads may refer to the rationale and classification frameworks. Each section builds toward a structured understanding of how business requirements shape and inform downstream design and delivery. Use the template and vector reference to draft and validate requirement statements, and refer to the glossary for consistent terminology.

# Drivers: The Foundation of Business Requirements

Business requirements do not arise in isolation. They are responses to drivers—forces that justify or necessitate change. These drivers may be external or internal, and they reflect the organization’s need to change direction, to seize opportunity, mitigate risk, increase strengths, or decrease weaknesses.

A business requirement, then, is a vector: a directional statement of intended change in direction by an amount. An example might be “reduce operational cost by 10%” or “increase market reach by 25%.”

However, before we can define such vectors, we must first uncover what is driving them.

Drivers are not always obvious. They may be strategic, operational, reputational, financial, or regulatory.

They may originate from outside the organization—such as shifts in law, technology, or public sentiment—or from within, such as capability gaps, inefficiencies, or misalignments.

To surface these drivers systematically, we use frameworks. These help analysts and architects think broadly, avoid blind spots, and ensure that the rationale for change is well-founded.

However, there is no overarching framework - so we use a combination of frameworks.

## SWOT

SWOT (Strengths, Weaknesses, Opportunities, and Threats) is not a framework to find the source of drivers itself, but a way of organizing those found with other frameworks, discussed next.

SWOT provides a balanced view of the organization’s position and helps identify where change is needed.

**Strengths** and **Weaknesses** are internal drivers. One has or has not the resources, appropriate processes, capital to continue on the current path, of go after an opportunity or threat.

**Opportunities** and **Threats** are primarily thought of as external. But there can be internal opportunities as well (possibilities to merge departments, or system, for example)

To populate SWOT meaningfully, we must scan both the external environment and the internal context. This is where PESTELE and internal driver categories come into play.

## External Drivers: PESTELE

To identify external drivers, we use the PESTELE framework: Political, Economic, Social, Technological, Environmental, Legal, and Ecological.

Each category prompts questions about forces that may influence strategy, operations, or reputation. For example:

* Are there new regulations we must comply with? *(Legal)*
* Are economic conditions affecting funding or demand? *(Economic)*
* Are technological shifts creating new service expectations? *(Technological)*

PESTELE is not a classification system—it’s a **lens**. It helps analysts look at an organisation, dept or project, to ensure coverage and avoid overlooking critical external pressures. The insights it produces feed directly into the **Opportunities** and **Threats** in SWOT.

## Internal Drivers: Organizational Concerns

Unlike external drivers, there is no universally accepted framework for internal analysis.

However, we can construct one by identifying key areas that influence organizational capability and readiness.

These areas help analysts surface **Strengths** and **Weaknesses**, and they provide a lens for understanding internal pressures and potential.

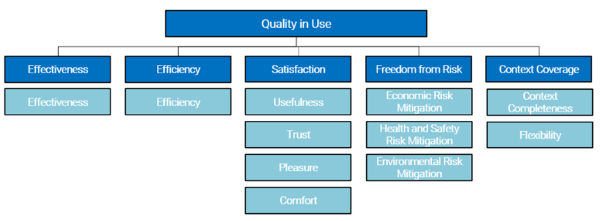
## Internal Drivers

As per SWOT, Strengths and Weaknesses are an internal, whereas Opportunities and Threats are External. That’s exactly right. Opportunities and Threats can also come from internal conditions.

Traditional tools for ferreting out internal weaknesses or strengths, is via a Capability Map of an organisation, so as to work back from services rendered to capabilities needed to do it well, to highlight weaknesses from capability gaps.

However Capability Maps are more about simply binary coverage, are not a tool that highlighting strengths or quality of service. For that, other filters are needed.

ISO-25022 is a standard for measuring user experience of services, and can be used to validate impartially – by end users -- whether internal capabiltlies are translating to external, independent assessment of the service. The qualities it measures is a series of high level categories, Effectiveness, Efficiency, Satisfaction, Freedom from Risk, Context Coverage, with Satisfaction having sub categories of Usefulness, Trust, Pleasure, Comfort, and Freedom from Risk have subcategories of Economic Risk Migration, Health and Safety, Environmental Risk Mitigation, and finally Context Coverage having subcategories of Context Completeness and Flexibility.



So it in turn needs a list of services an organisation offers.

Excellent—here’s the finalized \*\*paragraph-based version\*\* of your internal driver scanning framework, structured for project-level business analysis. Each paragraph introduces a lens for identifying internal drivers, offering practical guidance and examples to help analysts surface relevant concerns without drifting into enterprise architecture territory.

---

### \*\*Obligations\*\*

Obligations are hard requirements—legal, contractual, or policy mandates that must be met. These are non-negotiable drivers for change. Analysts should scan for any formal commitments that the project must uphold, such as statutory compliance, funding conditions, or inter-agency agreements. These often originate from governance bodies or external regulators and form the baseline for acceptable delivery.

---

### \*\*Compliance\*\*

Compliance drivers are softer than obligations but still influential. They include standards, guidelines, and best practices that the organization aims to meet—such as accessibility, privacy, equity, or sustainability. While not always mandatory, failing to meet these can result in reputational damage, stakeholder dissatisfaction, or future risk exposure. Analysts should identify which compliance frameworks are relevant and assess current alignment.

---

### \*\*Continuity & Risk\*\*

Business continuity and risk exposure are critical internal drivers. Analysts should look for single points of failure, lack of redundancy, or operational fragility that could disrupt service delivery. Risk assessments may reveal vulnerabilities in infrastructure, staffing, or process resilience. These drivers often justify investment in backup systems, failover mechanisms, or process redesign.

---

### \*\*Dependencies\*\*

Dependencies refer to reliance on systems, vendors, teams, or processes that constrain delivery. Analysts should identify any critical dependencies that introduce risk, delay, or complexity. This includes external integrations, shared platforms, or bottlenecked approvals. Redundancy and availability are key considerations—can the project proceed if a dependency fails?

---

### Domains

Domains are high-level areas of business activity relevant to the project. They help analysts scope the context and ensure coverage. For example, a project may touch on education, finance, or health domains. Identifying domains early helps guide capability mapping and stakeholder engagement, ensuring that no critical area is overlooked.

### Capabilities

Capabilities describe what the organization must be able to do. Analysts should identify which capabilities are required for the project, and whether they are present, missing, underdeveloped, or duplicated. Capability maps can help visualize gaps and strengths. Not all capabilities lead to services—but those that do are especially relevant for delivery planning.

### Services

Services are how capabilities are delivered—either internally or externally. Analysts should assess which services are available, accessible, and performing well. Service lifecycle matters: pre-service discovery, active delivery, refund handling, and post-service support all contribute to user experience. Weaknesses in service delivery often translate directly into business requirements.

### Resources

Resources include staffing, tooling, data, and infrastructure. Analysts should evaluate whether resources are sufficient, appropriately allocated, and aligned with project needs. Gaps in resourcing can delay delivery or reduce quality. Over-resourcing may signal inefficiency. Resource analysis should be tied to the capabilities and services being delivered.

### Processes

Processes are the workflows that enable capabilities and services. Analysts should look for inefficiencies, duplications, bottlenecks, or misalignments. Process maturity affects delivery speed, consistency, and adaptability. Mapping current processes can help identify where change is needed and what constraints must be addressed.

### Cost Structure

Cost is a universal driver. Analysts should assess both capital expenditure (CapEx) and operational expenditure (OpEx). Are costs justified by expected outcomes? Are they scalable or optimizable?

Sponsors often view cost as a primary lens—so analysts must be prepared to trace requirements back to cost-related drivers.

### Quality Delivery

Quality delivery refers to how well systems and services meet expectations. ISO/IEC 25010 and 25012 provide useful dimensions—such as reliability, usability, performance, and data quality. Analysts should assess whether current delivery meets these standards, and where improvements are needed. Quality gaps often drive requirements for redesign or enhancement.

### Accessibility

Accessibility ensures that services are usable by all intended users, including those with disabilities or constraints. Analysts should evaluate compliance with accessibility standards (e.g., WCAG), and identify barriers to access. Accessibility is both a compliance and equity concern, and often overlaps with reputation and trust.

### Availability

Availability refers to whether services or systems are online and usable when needed. Analysts should assess uptime, maintenance windows, and responsiveness. Availability gaps can justify investment in infrastructure, scheduling changes, or service redesign.

### Visibility / Discovery

Discovery is the ability of users to find or become aware of a service or product before engagement. Analysts should assess whether discoverability is a barrier—are users aware of what’s available? Are channels effective? Discovery issues often precede engagement problems and may require marketing, onboarding, or interface changes.

### Engagement & Conversion

Engagement and conversion metrics help analysts understand user behavior. Are users enrolling, converting, returning, or disconnecting? What are the inquiry, engagement, and return rates? These metrics can reveal friction points, value gaps, or usability issues that drive business requirements.

### Suite Cohesion / Completeness

Suite cohesion refers to how well services or products work together. Analysts should assess whether offerings are fragmented or unified, complete or partial. Gaps in cohesion can confuse users, reduce value, or increase support costs. Requirements may aim to consolidate, integrate, or expand the suite.

### Value Delivery

Value delivery is about whether users perceive the service or product as worthwhile. ISO/IEC 25022 offers metrics like satisfaction, effectiveness, and context coverage. Analysts should assess whether value is being delivered consistently and meaningfully. Misalignment between perceived and actual value is a strong driver for change.

### Reputation & Trust

Reputation and trust are soft but powerful drivers. Analysts should consider stakeholder confidence, public sentiment, and transparency. These are often influenced by quality, accessibility, and value delivery. Requirements may aim to improve communication, reliability, or responsiveness to rebuild trust.

**Strategic & Governance**

* Strategic Alignment
* Policy Mandates
* Governance Obligations
* Investment Justification

**Market & Value**

* Market Positioning
* Segment Coverage
* Brand Perception
* Value Proposition
* Competitive Differentiation

**Product & Service**

* Product Lifecycle
* Service Quality (Pre-service, During, Post-service, Refund)
* Service Availability
* Service Responsiveness
* Service Interconnectivity (Suite cohesion, completeness)

**User & Engagement**

* Discovery (views, impressions)
* Enrollment (sign-ups, inquiries)
* Conversion (engagement rate, conversion rate)
* Retention (return rate, disconnection rate)
* Satisfaction (ISO-25022 dimensions)

**Operational & Delivery**

* Process Maturity
* Delivery Efficiency
* Tooling and Automation
* Resource Allocation
* Dependency Management

**Risk & Resilience**

* Business Continuity
* Redundancy
* Risk Exposure
* Compliance Breadth
* Auditability

**Capability & Structure**

* Capability Gaps
* Underutilized Strengths
* Organizational Structure
* Role Clarity
* Information Availability

**Reputation & Trust**

* Stakeholder Confidence
* Public Trust
* Transparency
* Accountability

# Business Requirement Result from Drivers

Business requirements are based on an prior analyse of the drivers, the motivations for change. The drivers will be external or internal, based on strengths and weaknesses.

These drivers be collated as SWOT (Strength, Weakness, Opportunities, and Threats) based.   
Drivers can be from external sources, or internal sources.   
For external sources it’s common to use PESTELE (Political, economics, Social, Technical, ?, Law, Environmental) to list lenses at which to look for external opportunities and threats.  
Examples might include changes in law, social mores, green awareness, …We’ll talk of PESTELE more in depth in a sec.  
  
As for Internal drivers – strengths and weaknesses, there is no accepted acronym or framework to guide us. Possibly, there’s TOGAF?   
There’s possibly Capability Maps, traditionally used to map capabilitis of an org, which in turn highlights weaknesses due to omissions in resources available, or opportunities to consolidate, versus risk of too much in one basket, risking business continuity, etc.  
  
But that’s at the organisation level. What about the service, product, project level?  
It’s different from the whole org.   
While a capability map can be done on the rpoject’s resources, to highlight delivery weaknesses, I’m not sure that’s the best value.   
Plus, the sponsor is thinking maybe in terms of other dimensions.  
The drivers will be Strategic [Alignment], , Market [segments, depth], Cost [Design, Production, Marketing], Perceived Value [ Currency, Use Cases, and ISO-25022 qualities]…

# What Business Requirements Are

Business requirements are based on an analyse of the drivers, the motivations for change, which are in turn typically expressed as statements of desired increases or decreases of specific business service attributes. Examples include:

* Decrease: design, development, testing costs, marketing spend, operational overhead, development effort, staff requirements, etc.
* Increase: speed to market, feature set, service quality, market reach, stakeholder satisfaction, etc.

These requirements are not activities or tasks. They are outcomes—desired states that justify investment and guide solution design.

# How to Elicit and Classify Business Requirements

Eliciting business outcomes without a framework is risky. When engaging business sponsors, the conversation often gravitates toward immediate risks and long-term vision—leaving the middle ground of operational feasibility and incremental opportunity less explored. Conversely, operational stakeholders tend to focus on current weaknesses and constraints, which can obscure strategic possibilities and future-state aspirations. Additionally, they may be hesitant to surface perspectives that the feel may be at risk of being ad odds with sponsor stakeholders.

This asymmetry is not a flaw in stakeholder thinking—it’s a reflection of their roles. But it does mean that without a structured approach, the resulting business requirements may be incomplete, imbalanced, or biased toward the loudest concerns.

Frameworks help correct this.

Frameworks **force a wider scan** of potential drivers and outcomes, ensuring that both strategic intent and operational realities are captured. They also help analysts and architects **triangulate perspectives**, validate assumptions, and surface latent or politically sensitive drivers that may not emerge organically.

While later phases of system design can rely on standards such as **ISO/IEC 25010**, **25012**, or **25022** to evaluate system and data quality and user experience, the business requirements phase lacks a universally adopted standard.

Instead, practitioners often turn to widely recognized but non-standard frameworks such as PESTELE, SWOT, TOGAF, and ISO/IEC 25022 to guide their thinking.

These frameworks help avoid critical outcomes—such as trust, buy-in, continued use, customer retention, and churn reduction—are not forgotten or left undocumented. These are essential vectors that may otherwise be underrepresented in early documentation.

## Start with External Drivers: PESTELE

*Before* engaging stakeholders, begin by scanning external change forces using the PESTELE framework (Political, Economic, Social, Technological, Environmental, Legal, Ecological). These external pressures often shape strategic intent and justify investment.

Use PESTELE to guide your thinking and document as many potential threats and opportunities as possible—these may evolve into drivers for change.

When done, engage stakeholders to validate these early assumptions, listening *very* carefully for omissions in your prior analysis.

## Assess Internal Capabilities: Capability Mapping

While capability mapping is traditionally a business architecture tool, it can be scaled and adapted for use at the **department, program, or project level**—especially when business requirements are expected to influence or respond to internal strengths and gaps.

Use capability mapping to identify:

* **Gaps**: Capabilities that are missing or underdeveloped and may hinder delivery.
* **Surplus**: Existing strengths or resources that are underutilized and could be leveraged.

Before engaging stakeholders, analysts should develop a preliminary capability map based on available documentation, observations, and strategic context. This reduces the cognitive load on stakeholders by shifting their role from creators to validators for the most part. The goal is to present hypotheses for confirmation, correction, or completion—not to ask stakeholders to build the map from scratch.

However, remain highly attentive during engagement. Listen carefully for omissions, misclassifications, or latent capabilities that may not have surfaced in your initial analysis.  
  
Note: Refer to *IT Project Guidance – On Developing Capability Maps*.

## Use CLEAR to Guide the Engagement Process

When engaging stakeholders, consider following a **CLEAR** process:

* **Collaborative**: Engage across roles and levels to triangulate perspectives.
* **Limited**: Keep scope focused to avoid dilution.
* **Evaluated**: Validate assumptions and refine based on feedback.
* **Appropriate**: Ensure relevance to the domain and context.
* **Resource-conscious**: Respect time and effort constraints of participants.

This approach helps ensure that elicited requirements are not only comprehensive but also grounded in reality and stakeholder consensus.

# Examples Business Requirement Vectors

Business requirements are not abstract ideals—they are directional intentions to change something measurable or observable. These changes often relate to cost, quality, reach, or structure, and can be expressed as **vectors**: increases or decreases in specific attributes. While some vectors align with external drivers (e.g., from PESTELE), many are internal and operational. The following categories offer a reference set of common vectors to consider when drafting or validating business requirements.

## Political

Increase Supply Chain Options by x% by when: decrease reliance on a single region or country.

Increase reliance on local comparable capabilities:

## Economical

## Social

Increase trust in service

## Technological

Decrease number of components: decrease cost and delay of assembly.

**Dependency Count**: Reduce the number of dependent components or systems.

Increase use cases possible.

## Legal

## Registry Vectors

* **Registry Value Count**: Increase or decrease the number of entries in a registry (e.g., enrolments, records, cases).

## Design and Build Vectors

* **Component Count**: Consolidate or expand the number of functional components.
* **Component Quality**: Improve reliability, maintainability, or performance of components.

## Resource & Cost Vectors

* **Parts Cost**: Reduce cost of physical or digital components.
* **People Cost**: Reduce staffing requirements or optimize resource allocation.
* **Support Cost**: Minimize ongoing support and maintenance overhead.
* **Training Cost**: Reduce onboarding or enablement effort for users or staff.

## Integration & Interface Vectors

* **Integration Points**: Reduce or streamline the number of system interfaces.
* **Dependency Depth**: Flatten or simplify dependency chains.
* **Interoperability**: Increase compatibility across platforms or domains.
* **Automation Coverage**: Expand automation to reduce manual effort.

## Market & Value Vectors

* **Increase Value:** increase perceived value.
* **Sales Price**: reduce sales price to improve competitiveness.
* Increase
* **Market Reach**: Expand into new segments or geographies.
* **Service Quality**: Improve responsiveness, reliability, or user satisfaction.
* **Time to Market**: Accelerate delivery of new services or features.

## Strategic Vectors

* **Strategic Alignment**: Increase alignment with organizational goals or mandates.

## Reputation Vectors

* **Reputation**: Improve public trust, stakeholder confidence, or brand perception.

## Risk and Compliance Vectors

* **Reduce Compliance Risk**: Reduce exposure to legal or regulatory penalties.
* **Increase Compliance Breadth**: more compliance frameworks (accessibility, privacy, security, financial)

# Language Use and Misunderstanding Avoidance

Clear, actionable language is essential in requirement development. Vague or overly prescriptive terms can lead to misinterpretation, unrealistic expectations, or unimplementable obligations. This section outlines common language pitfalls and offers guidance on how to avoid them.

## SMART Business Requirements

While the full SMART framework (Specific, Measurable, Achievable, Relevant, Time-bound) may not always apply cleanly to business requirements, several elements remain useful.

Business requirements should be Singular in focus—each statement should reflect one driver or one intended change. Whether it’s the driver or the action that is singular depends on the framing, but clarity improves when each requirement addresses one change vector.

Requirements should be Achievable, with caveats. For example, Dependencies are acceptable as long as they are acknowledged. Unrealistic or aspirational goals should be flagged and scaled appropriately.

Relevant requirements are those that align with the strategic intent of the project or program. Avoid including requirements that are tangential or speculative.

Time/resource limitations are useful to include where known. Even if not precise, indicating urgency or delivery windows helps shape expectations.

Measurability is the most difficult to apply early on. While exact metrics may not be available, directional goals and indicative targets (e.g., “reduce by 10%”) are still more valuable than not expressed.

## On Embedding MoSCoW Early

While premature embedding of MoSCoW (Must, Should, Could, Won’t) risks overcommitment, early framing of Obligations and Prohibitions removes ambiguity. Avoid Permissions and Recommendations, which often lack follow-through. Focus on what must be done or must not be done.

Whether an organisation uses MosCoW or RFC 2119 is not an important decision: both are acceptable to map Obligations, Recommendations, Permissions, Prohibitions.

## Avoid “All”

“All” implies universality, which is rarely achievable or testable. In business requirements, it can overstate the scope of change or create unrealistic expectations.

Replace “all” with quantifiable or bounded terms. If universality is truly required, define the scope explicitly.

**Example**:  
“All services must be improved to meet customer expectations.”

**Improvement**:  
“Improve customer-facing services that contribute to 80% of support volume, with a focus on reducing response time and increasing satisfaction.”

## Avoid “Shall” and “Will”

These terms are often used to imply obligation but lack clarity on enforceability or priority.

Use obligation terms from MoSCoW (Must, Should, Could, Won’t) or define obligations and prohibitions explicitly. Avoid “will” unless describing future states or intentions outside the scope of requirements.  
  
**Example:**“The program shall deliver benefits to all partner agencies.”

**Improvement**:  
“The program must deliver measurable benefits to partner agencies involved in frontline service delivery, prioritizing those with high-volume interactions.”

## Use “And” Carefully

“And” can conflate multiple requirements into one, making them harder to test or implement.

Split compound statements into discrete, testable requirements.

**Example**:  
“The initiative must reduce cost and increase reach and improve equity.”

**Improvement**:

* “Reduce operational cost by consolidating overlapping service channels.”
* “Increase reach by expanding eligibility criteria to include underserved groups.”
* “Improve equity by aligning service delivery with regional demographic needs.”

## Use “If” Sparingly

Conditional logic can obscure the core requirement and introduce ambiguity.

Use “if” only when conditional behaviour is essential. Prefer role-based or state-based framing.

**Example**:  
“If uptake is low, marketing efforts will be increased.”

**Improvement**:  
“Marketing efforts must be scaled to achieve a minimum uptake threshold of 60% within the first quarter of launch.”

## Express Scale Expectations Where Possible

Even if aspirational, expressing scale helps set realistic expectations.

Use directional goals (increase/decrease) with indicative targets. Avoid vague terms like “improve” without context.

**Example**:  
“Improve access to digital services.”

**Improvement**:  
“Improve access to digital services by increasing mobile usage by 25% among rural users within 12 months.”

## Risk of Over-Structuring Early Requirements

While this guidance encourages early structure (e.g., SMART, MoSCoW) it risks premature rigidity if not balanced with iterative validation.

Early structure should guide—not constrain—exploration. Requirements may evolve as stakeholder understanding deepens.

## Distinguishing Business Requirements from User Requirements

It is awfully common to find blurred lines between business requirements and user requirements—especially in early project phases or when stakeholders express needs in terms of outcomes rather than actions.

This confusion is understandable: both types of requirements describe change, and both are essential to shaping a solution. However, they serve distinct purposes and must be treated accordingly.

Business requirements articulate the drivers and motivations for change. They describe the desired outcomes—such as reducing cost, increasing market reach, or improving service quality. These are strategic in nature and often expressed as directional vectors (e.g., “reduce operational cost by 10%”).

User requirements, on the other hand, describe the specific tasks, interactions, or capabilities that users need to perform or access to support those outcomes. They are operational and actionable, often framed in terms of roles, personas, and use cases (e.g., “a regional coordinator must be able to generate a monthly performance report”).

# Business Requirement Statement Template

Below is an example of a business requirements composed from a set of analysed observations and expectations:

* ID: BR-001    
  Title: Reduce Operational Cost    
  Statement: Must Reduce operational cost by 10% within 6 months.   
  Rationale: Current operational overhead exceeds budget targets.
* Developed from the following aspects or parts:
  + Driver: Efficiency improvement in service delivery
  + Motivation: Current operational overhead exceeds budget targets
  + Vector Direction and category: Reduce, operational cost
  + Scale: By 10% within 6 months.
  + PESTLE: Economic pressure from budget constraints
  + SWOT: Opportunity to streamline redundant processes

# Conclusion

Business requirements are the foundation upon which successful IT projects are built. They articulate the rationale for change, grounded in strategic drivers and operational realities. By focusing on outcomes rather than activities, and by using structured frameworks to elicit, classify, and validate these requirements, analysts and architects can ensure that the solutions developed are both relevant and achievable.

This guidance emphasizes doing the analytical groundwork upfront—scanning external pressures, assessing internal capabilities, and preparing hypotheses—so that stakeholder engagement becomes a process of confirmation rather than creation. It also stresses the importance of clear, singular, and scalable language, avoiding common pitfalls that lead to ambiguity or overreach.

Ultimately, well-formed business requirements enable better decision-making, clearer scope definition, and more effective alignment between business intent and technical delivery. They are not just a step in the process—they are the compass that guides it.

Appendices

Appendix A - Document Information

Authors & Collaborators

* Sky Sigal, Solution Architect

### Versions

* 1. Initial Draft

### Images

**No table of figures entries found.**

### Tables

**No table of figures entries found.**

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Russell Campbell, Technical Manager |  |
| Carrie Buckmaster, Senior Analyst |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Acronyms

API

: [Application Programming Interface](#Term_ApplicationProgrammingInterface).

DDD

: Domain Driven Design

GUI

: [Graphical User Interface](#Term_ApplicationProgrammingInterface). A form of [UI](#Acronym_UI).

ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. [IT](#Acronym_IT) is a subset of ICT.

IT

: acronym for Information, using Technology to automate and facilitate its management.

PESTELE

:

* **Political**: Government policies, regulatory changes, and political stability.
* **Economic**: Market conditions, funding availability, inflation, and cost pressures.
* **Social**: Demographic shifts, cultural expectations, and public sentiment.
* **Technological**: Emerging technologies, innovation cycles, and digital disruption.
* **Environmental**: Operational sustainability concerns such as energy use, waste management, and carbon footprint—typically tied to compliance or efficiency.
* **Legal**: Laws, regulations, and contractual obligations that affect operations or service delivery.
* **Ecological**: Broader ecosystem impacts including biodiversity, land use, climate resilience, and long-term environmental stewardship—often tied to strategic or reputational considerations.

SMART

: Singular, Measurable, Achievable, Relevant, Time/Resource constrained.

SWOT

: Strengths, Weaknesses, Opportunities, Threats

UI

: User Interface. Contrast with [API](#Acronym_API).

### Terms

Refer to the project’s Glossary.

Application Programming Interface

: an Interface provided for other systems to invoke (as opposed to User Interfaces).

Capability

: a capability is what an organisation or system must be able to achieve to meet its goals. Each capability belongs to a domain and is realised through one or more functions that, together, deliver the intended outcome within that area of concern.

Domain

: a domain is a defined area of knowledge, responsibility, or activity within an organisation or system. It groups related capabilities, entities, and functions that collectively serve a common purpose. Each capability belongs to a domain, and each function operates within one.

Driver

: a motivating force or condition that justifies the need for change. Drivers may originate from strategic goals, external pressures, or operational inefficiencies. They are the “why” behind a business requirement and often reflect directional intent (e.g., reduce cost, increase reach).

Entity

: an entity is a core object of interest within a domain, usually representing a person, place, thing, or event that holds information and can change over time, such as a Student, School, or Enrolment.

Function

: a function is a specific task or operation performed by a system, process, or person. Functions work together to enable a capability to be carried out. Each function operates within a domain and supports the delivery of one or more capabilities.

Motivation

: the contextual reason or rationale for a specific business requirement. While a driver reflects the broader intent, motivation explains the immediate condition or challenge that makes the requirement relevant (e.g., budget overruns, service delays, stakeholder dissatisfaction).

Outcome

: a desired future state resulting from the implementation of a business requirement. Outcomes are directional and aspirational, such as improved service quality, reduced cost, or expanded market reach. They are not tasks or deliverables, but the impact of change.

Output

: a tangible deliverable or result produced by a system, process, or initiative. Outputs are typically measurable and concrete (e.g., a report, a service channel, a data feed), and may contribute to achieving one or more outcomes.

Person

: a physical person, who has one or more Personas. Not necessarily a system User.

Persona

: a facet that a Person presents to a Group of some kind.

Quality

: a quality is a measurable or observable attribute of a system or outcome that indicates how well it meets expectations. Examples include reliability, usability, and performance. Refer to the ISO-25000 SQuaRE series of standards.

[Business] Service

: a service delivered by an organization to meet business needs or fulfil stakeholder expectations. Business services are distinct from automated system services—they may involve human processes, policy decisions, or multi-channel delivery. They represent the functional capabilities of the business, not just the technical implementation.

User

: a human user of a system via its UIs.

User Interface

: a system interface intended for use by system users. Most computer system UIs are Graphics User Interfaces ([GUI](#Acronym_GUI)) or Text/Console User Interfaces (TUI).

Appendix B – Example Requirements

Clear, Measurable, and Standards-Aligned Requirements for Service Improvement

# Market Reach and Segmentation

* Increase the total number of active users by X% within 12 months of release.
* Expand market segmentation coverage by 25%, measured by the addition of at least two new user segments (e.g., Early Learners, Primary, Secondary, Tertiary, Professional, Senior).

# Service Availability

* Extend consumers service availability from 9:00–17:00 to 5:00–24:00 (local time),
* ensuring 99.95% availability during all operating hours as measured quarterly.

# Equitability and Accessibility

* Improve accessibility compliance to meet WCAG 2.2 AA standards for visual, auditory, physical, and cognitive disabilities by the next major release.
* Increase the percentage of users with disabilities who report successful task completion by 30%, as measured in post-release usability studies.

# User Roles and Breadth of Access

* Expand supported user roles from Teachers and Students to include Parents and Caregivers, allowing for role-based access and tailored user interfaces.

# Functional Capability

* Increase the number of core system capabilities by 20% to support additional key business processes, tracked through functional coverage analysis.

# Usability and Support

* Reduce customer support request volume by 40% within six months post-launch, as an indicator of improved usability, accessibility, and feature recognizability.

# Operational and Environmental Efficiency

* Achieve a 25% reduction in business-as-usual (BAU) hosting costs, including a 15% decrease in energy usage, contributing to organizational environmental goals.

# Security and Privacy

* Reduce the number of flagged security and privacy issues per release by 50%, as measured by vulnerability assessments and privacy audits.
* Implement enhanced privacy controls that enable users to manage and share consents, meeting new regulatory and political requirements.

# Development Efficiency

* Lower the average cost of developing new features by 20% over the next development cycle through reuse, modular design, and improved tooling.

# Integration and Interoperability

* Increase the number of external systems integrated with the core platform by 30% within a year, unlocking better data and workflow connectivity.
* Expand support for interoperability standards and APIs to include at least three additional protocol or data-exchange methods.

# Social and Organizational Outcomes

* Enable broader user participation by expanding the set of community and support roles, fostering increased engagement and mutual assistance.

# PESTLE Considerations

* Address Political and Legal drivers by ensuring compliance with emerging privacy legislation and social equity requirements.
* Respond to Economic and Environmental pressures by optimizing operational costs and energy efficiency, as above.
* Adapt to Social trends by facilitating broader access and user empowerment, supporting inclusion and diversity in participation.
* Leverage Technological advances to support better integration, interoperability, and system scalability as detailed above.