

# System Requirements Analysis

System requirements ensure alignment between stakeholders and the technical team.

# Introduction to System Requirements

## 1. Structured Requirements



- System requirements shall align with stakeholder needs.
- They shall provide clarity to the development team.
- They ensure the project stays on track.



## 2. Key Outcomes of Analysis



- System requirements shall define clear goals and objectives.
- They help meet stakeholders' needs.
- Prevent costly revisions and rework.

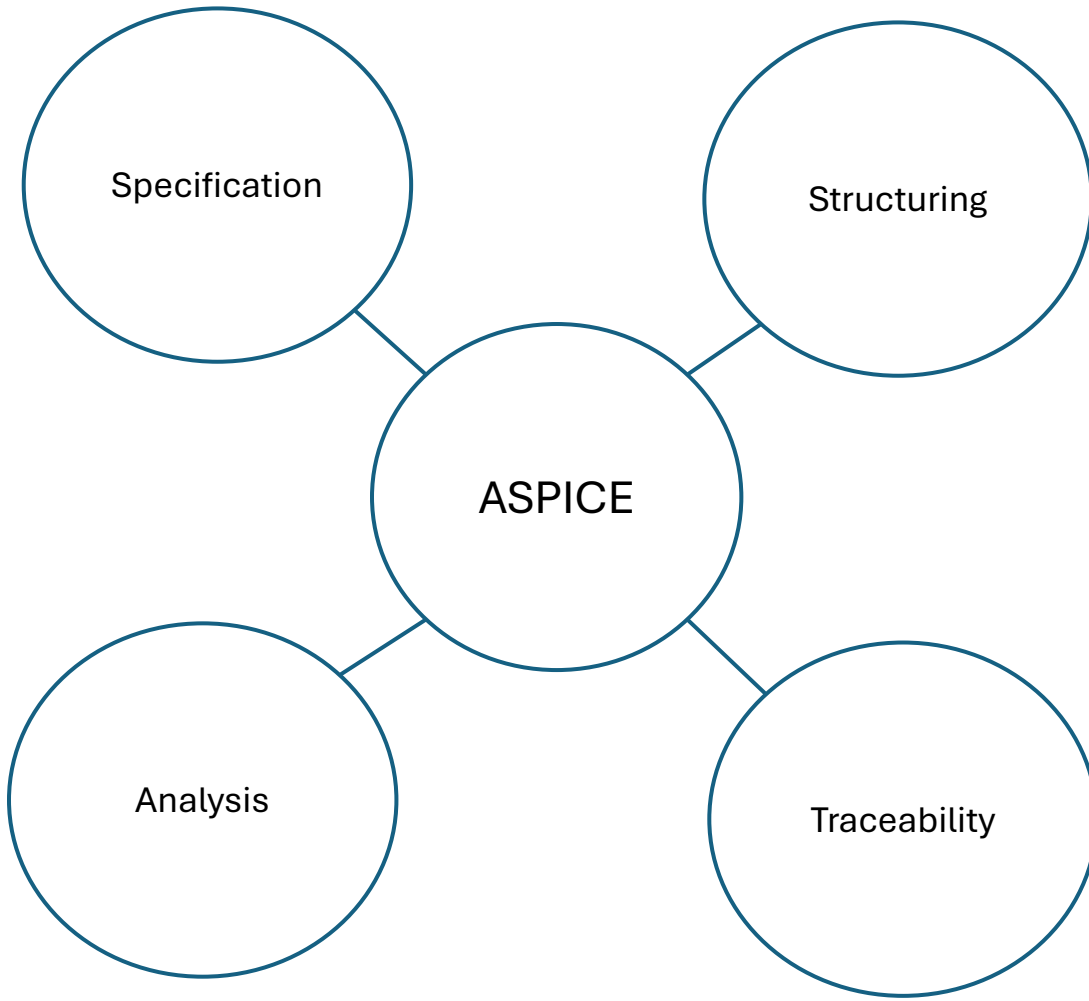


## 3. Critical for Technical Feasibility



- System requirements shall assess whether the infrastructure can support the system.
- They guide developers to create viable solutions.
- Provide a framework for future scalability.

ASPICE connects these practices to improve process capability.



### 1. Process Model Framework

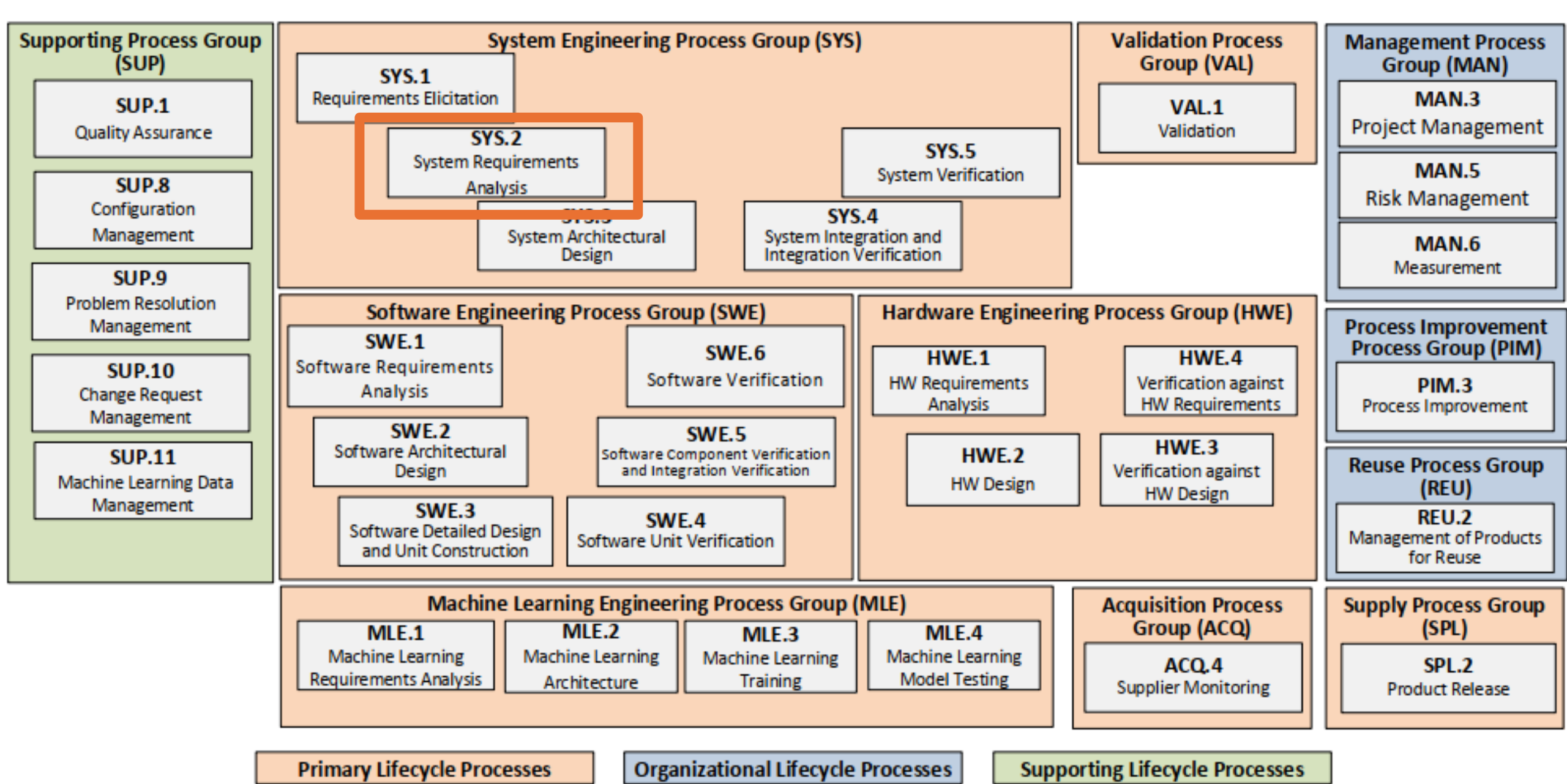
- ASPICE shall ensure quality across the development lifecycle.
- It focuses on continuous process improvement.
- It supports compliance with industry standards.

### 2. Key Base Practices

- ASPICE shall ensure system requirements meet project needs.
- It shall provide structure for documenting functional and non-functional requirements.
- Ensure that all requirements are tested and validated.

### 3. Capability Improvement

- ASPICE shall enhance communication between teams.
- It ensures scalable and sustainable processes.
- Leads to better project outcomes.



<b>Process ID</b>
<b>SYS.2</b>
<b>Process name</b>
<b>System Requirements Analysis</b>
<b>Process purpose</b>
The purpose is to establish a structured and analyzed set of system requirements consistent with the stakeholder requirements.
<b>Process outcomes</b>
1) System requirements are specified 2) System requirements are structured and prioritized 3) System requirements are analyzed for correctness and technical feasibility 4) The impact of system requirements on the operating environment is analyzed

<b>SYS.2 System Requirements Analysis</b>	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
<b>Output Information Items</b>						
17-00 Requirement	X	X				
17-54 Requirement Attribute		X	X			
15-51 Analysis Results			X	X		
13-51 Consistency Evidence					X	
13-52 Communication Evidence						X
<b>Base Practices</b>						
BP1: Specify system requirements	X					
BP2: Structure system requirements		X				
BP3: Analyze system requirements			X			
BP4: Analyze the impact on the system context				X		
BP5: Ensure consistency and establish bidirectional traceability					X	
BP6: Communicate agreed system requirements and impact on the system context						X

<b>Base practices</b>
<b>SYS.2.BP1: Specify system requirements.</b> Use the stakeholder requirements to identify and document the functional and non-functional requirements for the system according to defined characteristics for requirements.  <i>Note 1: Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO 26262-8:2018, or the INCOSE Guide For Writing Requirements.</i> <i>Note 2: Examples for defined characteristics of requirements shared by technical standards are verifiability (i.e. verification criteria being inherent in the requirements formulation), unambiguity/comprehensibility, freedom from design and implementation, and not contradicting any other requirement.</i>
<b>SYS.2.BP2: Structure system requirements.</b> Structure and prioritize the system requirements.  <i>Note 3: Examples for structuring criteria can be grouping (e.g. by functionality) or product variants identification.</i> <i>Note 4: Prioritization can be done according to project or stakeholder needs via e.g. definition of release scopes. Refer to SPL.2.BP1</i>
<b>SYS.2.BP3: Analyze system requirements.</b> Analyze the specified system requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates.  <i>Note 5: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.</i> <i>Note 6: Technical feasibility can be evaluated based on e.g. platform or product line, or by means of prototype development or product demonstrators.</i>
<b>SYS.2.BP4: Analyze the impact on the system context.</b> Analyze the impact that the system requirements will have on elements in the relevant system context.
<b>SYS.2.BP5: Ensure consistency and establish bidirectional traceability.</b> Ensure consistency and establish bidirectional traceability between system requirements and stakeholder requirements.  <i>Note 7: Bidirectional traceability supports consistency, facilitates impact analyses of change requests, and supports the demonstration of coverage of stakeholder requirements.</i>
<b>SYS.2.BP6: Communicate agreed system requirements and impact on the system context.</b> Communicate the agreed system requirements, and results of the impact analysis on the system context, to all affected parties.

# System Requirements Analysis

The analysis process aligns system requirements with stakeholder goals and ensures feasibility.

## 1. Purpose of Analysis



- The analysis shall align system outcomes with business needs.
- It reduces risks of scope creep and delays.
- Ensures that technical goals are achievable.



## 2. Process Outcomes



- The analysis shall deliver well-defined system requirements.
- Requirements shall be structured for implementation.
- Feasibility shall be ensured, avoiding technical debt.



## 3. Impact on Environment

- System requirements shall predict necessary infrastructure changes.
- They prevent unnecessary system disruptions.
- Allow planning for long-term maintenance.

# BP1: Specify System Requirements

## System requirements are classified into functional and non-functional types to ensure clarity.

**SYS.2.BP1: Specify system requirements.** Use the stakeholder requirements to identify and document the functional and non-functional requirements for the system according to defined characteristics for requirements.

*Note 1: Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO 26262-8:2018, or the INCOSE Guide For Writing Requirements.*

*Note 2: Examples for defined characteristics of requirements shared by technical standards are verifiability (i.e. verification criteria being inherent in the requirements formulation), unambiguity/comprehensibility, freedom from design and implementation, and not contradicting any other requirement.*

### 1. Defined Characteristics

- Requirements shall use ISO standards (e.g., ISO 29148) for verifiable, unambiguous specifications.
- They shall clearly specify both functional and non-functional requirements.
- Avoid design constraints in requirement specifications.

### 2. Functional and Non-Functional

- System requirements shall define what the system must do (functional).
- They shall specify system properties (non-functional), like performance and security.
- Ensure both types of requirements are clearly differentiated..

### 3. Verifiability and Freedom from Ambiguity

- Requirements shall be measurable and testable.
- They shall be clear and free from ambiguity.
- Ensure they are actionable and realistic.

# Types of System Requirements

## Distinction between functional and non-functional requirements.

### 1. Functional Requirements



- System requirements shall define specific system actions or functionalities.
- Example: The system shall allow users to reset passwords.
- Focus on the user-facing behavior of the system.

### 2. Non-Functional Requirements



- System requirements shall define system properties like performance, usability, and security.
- Example: The system shall handle 10,000 concurrent users.
- Impact system quality rather than functionality.

### 3. Measurable and Verifiable



- Both functional and non-functional requirements shall be measurable and verifiable.
- Ensure requirements can be tested effectively.
- Specific criteria should be set for validation.



System requirements are classified into functional and non-functional types to ensure clarity.

**SYS.2.BP2: Structure system requirements.** Structure and prioritize the system requirements.

*Note 3: Examples for structuring criteria can be grouping (e.g. by functionality) or product variants identification.*

*Note 4: Prioritization can be done according to project or stakeholder needs via e.g. definition of release scopes. Refer to SPL.2.BP1*

### 1. Grouping by Functionality



- Requirements shall be structured by functionality to ensure clarity.
- Grouping helps align system objectives.
- Structure requirements to meet project or stakeholder needs.

### 2. Prioritization by Needs



- Prioritize requirements according to stakeholder or project needs.
- Ensure high-priority functionalities are addressed first.
- Use prioritization to meet critical milestones.

### 3. Variants and Release Scope



- Requirements shall be grouped according to system variants or release scope.
- Align them with project phases for easier implementation.
- Ensure the system can evolve with future releases.

# BP2: Structure System Requirements

# Examples of Structuring System Requirements

## System requirements can be grouped to match project phases or critical functionalities

### 1. Functionality Grouping Example

- Group system requirements related to user authentication into a single functionality group.
- Example: Login, password reset, and two-factor authentication are grouped under "User Authentication."
- This ensures clarity and avoids duplication of requirements across other groups.

### 2. Release-Based Structuring

- System requirements can be grouped by release cycles, separating MVP from future features.
- Example: Basic login functionality is developed in the MVP, while advanced security features are reserved for later releases.
- Prioritizing based on release cycles ensures critical functionalities are delivered first.

### 3. Clarity in Grouping

- Well-structured requirements ensure smoother implementation and reduce redundancy.
- Grouping allows teams to focus on specific feature sets without confusion.
- It simplifies traceability and ensures all related requirements are tracked together.

# BP3: Analyze System Requirements

## Analyzing system requirements ensures their technical feasibility and alignment with project goals

**SYS.2.BP3: Analyze system requirements.** Analyze the specified system requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates.

*Note 5: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.*

*Note 6: Technical feasibility can be evaluated based on e.g. platform or product line, or by means of prototype development or product demonstrators.*

### 1. Correctness and Feasibility

- System requirements shall be analyzed for technical feasibility.
- Ensure they are implementable and aligned with project goals.
- Avoid conflicting requirements that hinder progress.

### 2. Interdependencies

- Analyze how system requirements affect each other.
- Ensure no conflicts between requirements.
- Clarify interdependencies for successful implementation.

### 3. Impact on Estimates

- Analyzing requirements helps provide accurate project estimates.
- Requirements shall be realistic within time and cost constraints.
- Accurate estimates prevent resource misallocation.

# Analyzing requirements helps prevent conflicts and ensures correct implementation

## Examples of Analyzing System Requirements

### 1. Feasibility Example

- A performance requirement shall be analyzed to ensure it aligns with available hardware capacity.
- Example: A system must support 10,000 concurrent users, but current hardware only supports 5,000.
- Identify if hardware upgrades are required or if performance optimizations can be made.

### 2. Dependency Analysis Example

- A user interface requirement may depend on a security module that must be developed first.
- Example: Implementing multi-factor authentication requires the security module to be developed.
- Identify dependencies early to avoid delays in implementation.

### 3. Correctness Check

- Requirements must be checked for correctness and alignment with business rules.
- Example: A functional requirement may conflict with an existing business logic, causing discrepancies.
- Analyzing requirements for correctness ensures they align with overall project goals.

# BP4: Analyze Impact on the System Context

## System requirements impact multiple elements in the operating environment

**SYS.2.BP4: Analyze the impact on the system context.** Analyze the impact that the system requirements will have on elements in the relevant system context.

### 1. System Context Analysis

- Analyze how system requirements impact the broader system environment.
- Identify potential dependencies and constraints.
- Prevent disruption to the system during integration.

### 2. Contextual Dependencies

- System requirements shall consider the full operating context.
- Analyze impacts on related systems or infrastructure.
- Adjust system architecture as needed.

### 3. Plan for Adaptation

- Ensure the system context can adapt to new requirements.
- Plan for future modifications and enhancements.
- Minimize negative impacts on other system components.

# Examples of Impact on System Context Analysis

Understanding the system context helps identify impacts on key components.

## 1. Example: Network Impact



- System requirements for enhanced security may impact network performance.
- Example: Adding encrypted communication for sensitive data increases bandwidth use.
- Ensure that network infrastructure can handle the additional load.

## 2. Example: Hardware Impact



- System performance requirements often necessitate hardware upgrades.
- Example: The system shall process 10,000 transactions per second, requiring a faster processor.
- Evaluate the existing hardware to ensure compatibility.

## 3. Example: Software Impact



- Introducing new software features can impact third-party integrations.
- Example: Implementing a new API requires compatibility with the CRM system.
- Check for software compatibility issues before deployment.

# Bidirectional traceability ensures consistency across stakeholder and system needs.

**SYS.2.BP5: Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish bidirectional traceability between system requirements and stakeholder requirements.

*Note 7: Bidirectional traceability supports consistency, facilitates impact analyses of change requests, and supports the demonstration of coverage of stakeholder requirements.*

## 1. Consistency in Requirements

- System requirements shall remain consistent with stakeholder needs.
- Ensure alignment between technical and stakeholder objectives.
- Avoid conflicts that may arise from changing requirements.

## 2. Bidirectional Traceability

- Maintain bidirectional traceability between stakeholder and system requirements.
- Ensure all system requirements trace back to stakeholder inputs.
- Trace changes in system requirements back to the source.

## 3. Change Management

- Traceability shall support effective change management.
- Identify impacts of requirement changes on the system.
- Ensure traceability at every level to prevent misalignment.

# BP5: Ensure Consistency and Establish Bidirectional Traceability

# Example of Ensure Consistency and Establish Bidirectional Traceability

Traceability ensures that every stakeholder need is reflected in the system design

## 1. Example: Traceability of Functional Requirements

- A user story about login functionality traces to system requirements specifying authentication methods.
- Example: "As a user, I want to reset my password"  
→ "The system shall allow password reset via email."
- Traceability ensures this user need is fully implemented.

## 2. Example: Traceability of Non-Functional Requirements

- A stakeholder requirement for security traces to system requirements defining encryption standards.
- Example: "The system shall encrypt all user data" links to stakeholder needs for data privacy.
- Traceability guarantees these requirements are met.

## 3. Example: Managing Change Impact

- A change in user experience requirements triggers updates in system requirements for UI and performance.
- Example: "The system shall allow for quick password reset" changes to "The system shall reset passwords within 30 seconds."
- Bidirectional traceability ensures change impacts are tracked.



# BP6: Communicate Agreed System Requirements and Impact on System Context

## Communicating the impact of system requirements on the environment ensures stakeholder alignment

**SYS.2.BP6: Communicate agreed system requirements and impact on the system context.** Communicate the agreed system requirements, and results of the impact analysis on the system context, to all affected parties.

### 1. Clear Communication with Stakeholders

- System requirements shall be clearly communicated to stakeholders.
- Ensure all stakeholders understand the impact of system requirements.
- Confirm agreements on system requirements before implementation.

### 2. System Context Impact Analysis

- Communicate how system requirements will affect the operating environment.
- Ensure stakeholders are informed of potential impacts on system performance.
- Adjust system plans based on feedback from stakeholders.

### 3. Agreements Documentation

- Document stakeholder agreements on system requirements and their impact.
- Keep records of all approved requirements and their impacts.
- Ensure these documents are accessible for future reference.

# Example of Communicate Agreed System Requirements and Impact on System Context

Communicating and documenting system impacts ensure alignment and understanding.

## 1. Example: Communication of Performance Requirements



- Communicate how a system performance requirement affects network infrastructure and server capacity.
- Example: "The system shall support 10,000 concurrent users" requires network upgrades to support additional bandwidth.
- Ensure these impacts are communicated early to avoid bottlenecks.

## 2. Example: Documenting Security Impact



- Document how new security requirements will impact the system's authentication flow.
- Example: "The system shall enforce multi-factor authentication" affects login and user sessions.
- Ensure stakeholders are aware of the added complexity.

## 3. Example: Alignment of New Features



- Communicate how adding new features might impact existing ones.
- Example: "The system shall introduce a new payment gateway" must be compatible with the existing system.
- Inform stakeholders of potential challenges with backward compatibility.

# Summary and Q&A

## System requirements, traceability, and communication are essential for project success

### 1. System Requirements are Critical

- Align system requirements with stakeholder needs for successful project outcomes.
- Well-structured, clear requirements prevent project delays and errors.

### 2. Traceability is Key

- Bidirectional traceability helps track changes and maintain consistency.
- Tools like Jira and Confluence simplify managing traceability across the project.

### 3. Effective Communication is Vital

- Clearly communicate system requirements and their impacts to stakeholders.
- Early management of impacts ensures smoother integration and fewer issues.