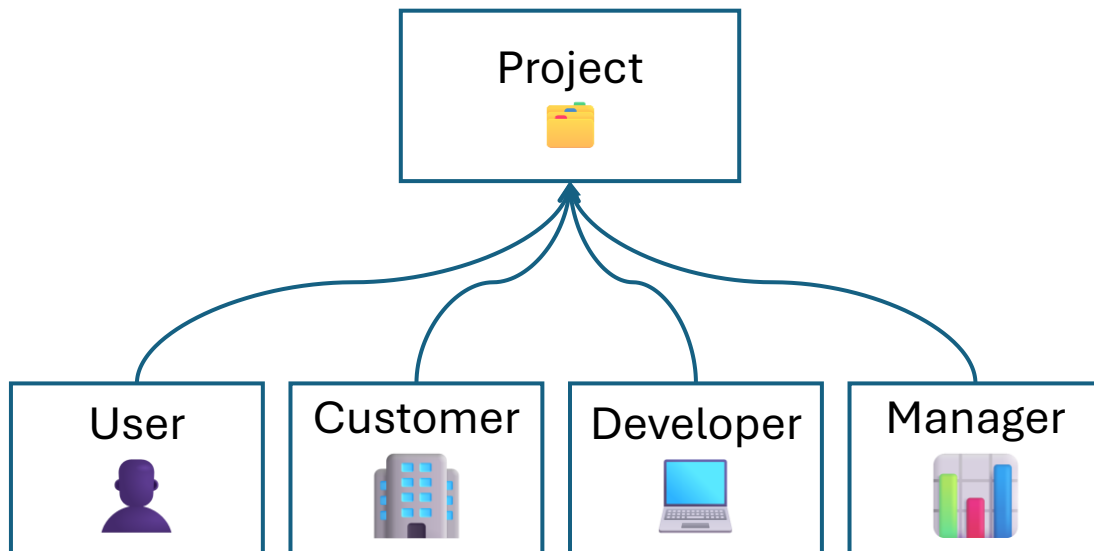


# Stakeholder Requirements and Scenario-Based Requirements

# Introduction to Stakeholder Requirements

Stakeholder requirements are derived from the diverse needs and expectations of all project stakeholders.



## 1. Definition of stakeholder requirements:

- Statements that capture the needs, expectations, and constraints of stakeholders to be addressed by the system or product.

## 2. Types of stakeholders:

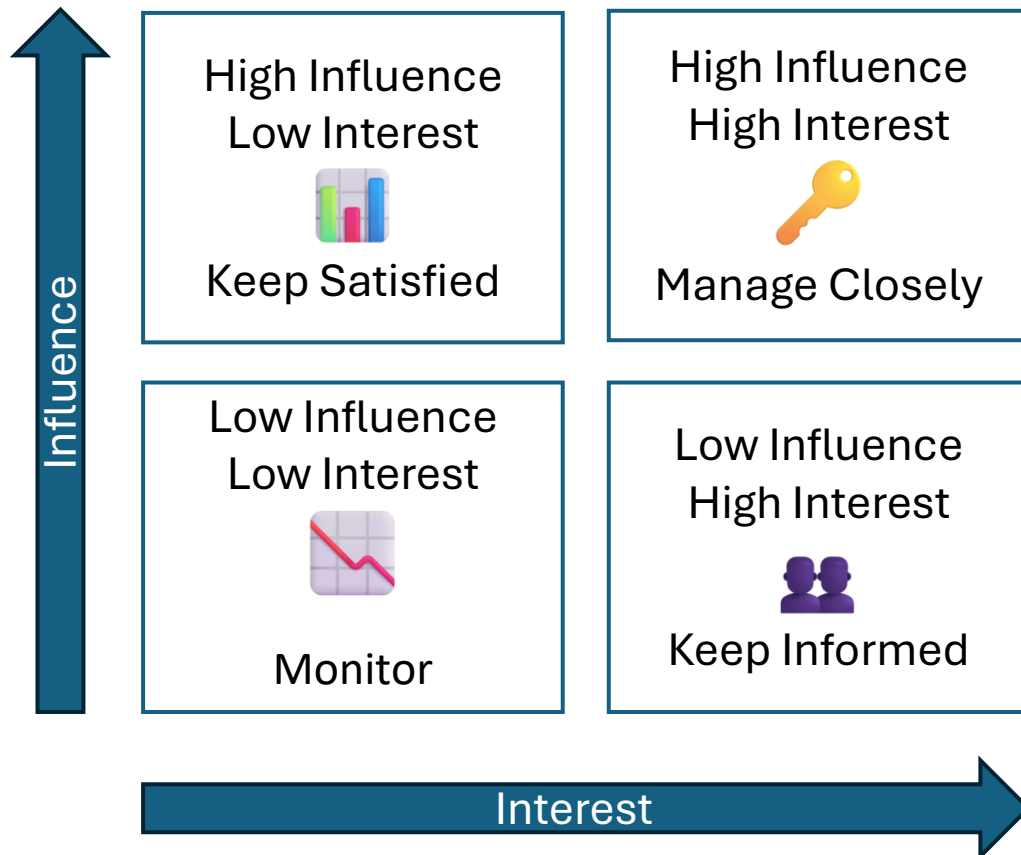
- **Customers:** Individuals or organizations purchasing or using the product.
- **End-Users:** People who directly interact with the system or product.
- **Developers:** Technical team responsible for building the product.
- **Managers:** Individuals overseeing project execution and resource allocation..

## 3. Importance of identifying and understanding stakeholder needs:

- Ensures alignment between project deliverables and stakeholder expectations.
- Helps prioritize features and functionalities based on user needs.

# Identifying Key Stakeholders

Stakeholder requirements are derived from the diverse needs and expectations of all project stakeholders.



## 1. Techniques for Identifying Stakeholders:

- **Stakeholder Mapping:** A visual method to identify and analyze stakeholders based on their influence and interest.

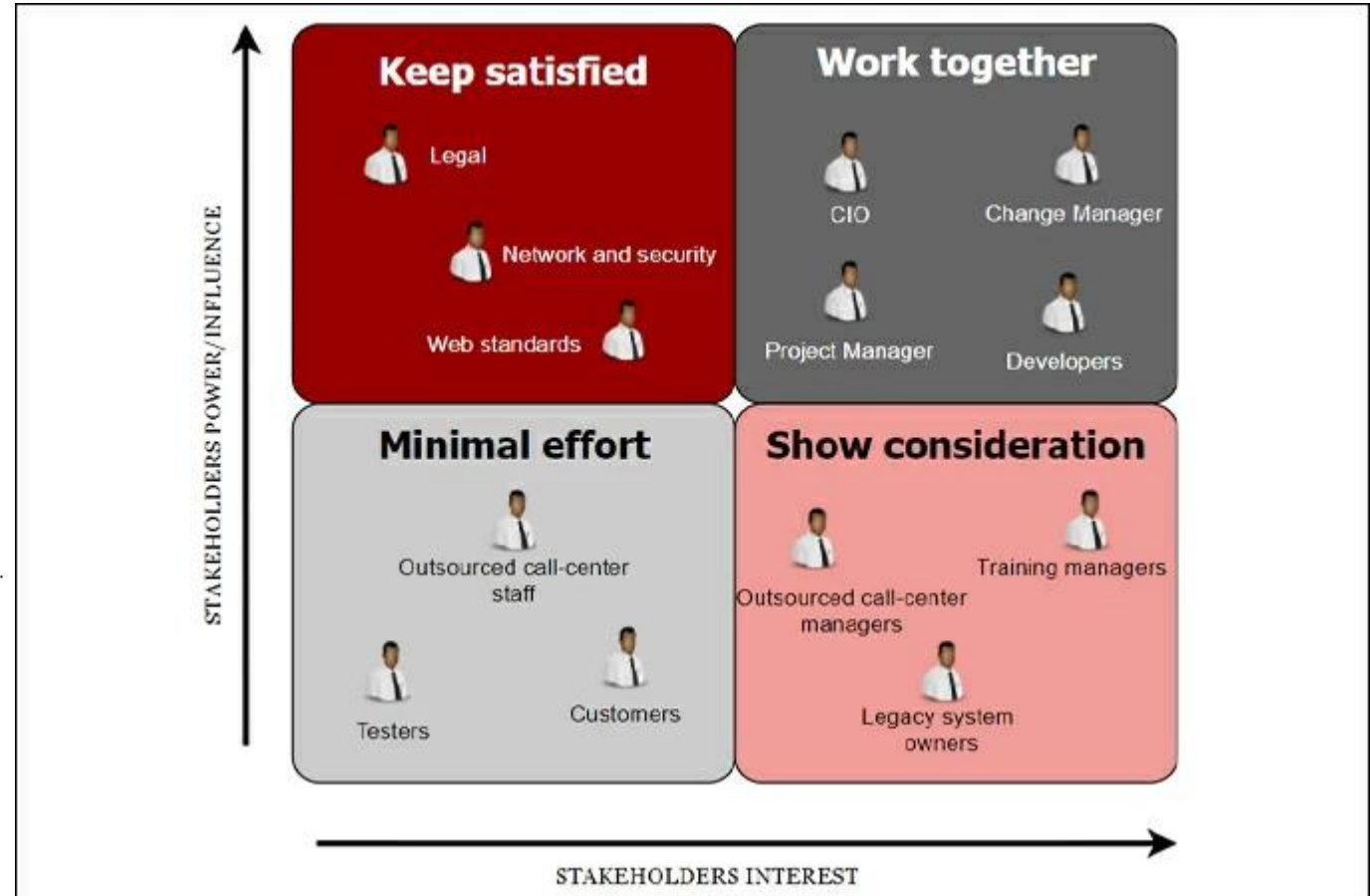
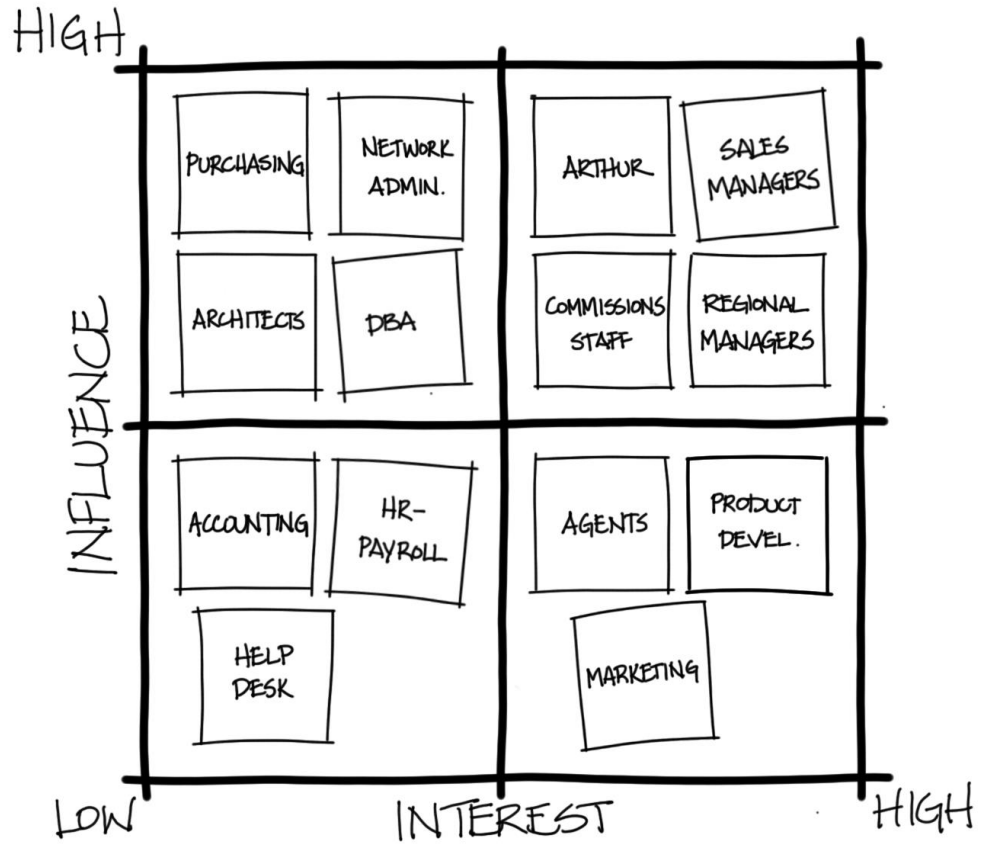
## 2. Categorizing Stakeholders:

- **High Influence / High Interest:** Key stakeholders who need to be managed closely. (e.g., Regulatory Agencies, Manager, Developer)
- **High Influence / Low Interest:** Stakeholders who should be kept satisfied. (e.g., Senior Management, Major Investors)
- **Low Influence / High Interest:** Stakeholders who need to be kept informed. (e.g., End-Users, Local Community Groups)
- **Low Influence / Low Interest:** Stakeholders who require minimal monitoring. (e.g., General Public, Minor Suppliers)

## 3. Understanding Stakeholder Influence and Interest:

- Assessing how much influence each stakeholder has on the project and their level of interest in its outcomes.

# Key Stakeholders Example



<https://insideproduct.co/stakeholder-map/>





<https://gitmind.com/stakeholder-analysis.html>

# Prioritizing Stakeholder Requirements

## MoSCoW Technique

Categorize requirements into Must Have, Should Have, Could Have, and Won't Have to prioritize effectively.

### 1. Categorization of Requirements

-  **Must Have:** Essential features critical for project success; without these, the project cannot function.
-  **Should Have:** Important but not critical requirements that add significant value to the project.
-  **Could Have:** Desirable features that enhance user experience but are not essential; can be deferred if necessary.
-  **Won't Have:** Features that are agreed upon as out of scope for the current iteration; these can be revisited later.

### 2. Focus on Must Have Features

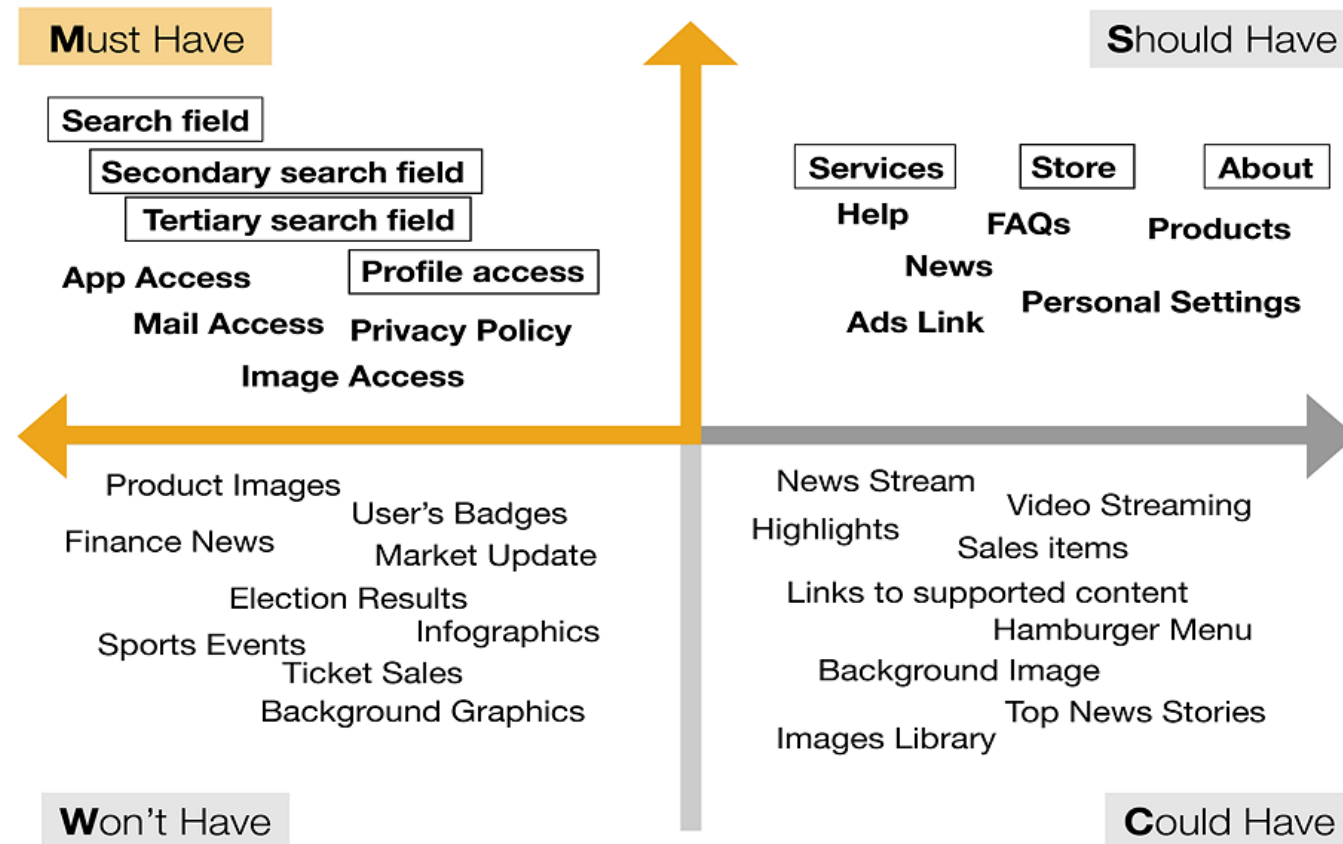
- Must Have requirements are non-negotiable and must be completed before considering the project ready for release.
- Ensuring these items are prioritized helps mitigate risks associated with project failure.

### 3. Managing Stakeholder Expectations

- The MoSCoW method enhances communication among stakeholders by clearly defining priorities and expectations.
- It helps avoid scope creep by deferring less critical features, allowing teams to focus on delivering a minimum viable product (MVP) earlier.
- Establishing a shared understanding of what constitutes each category fosters collaboration and reduces conflicts during project execution..

# MoSCoW Example

<https://www.edrawmind.com/templates/moscow-matrix-template.html>






# Prioritizing Stakeholder Requirements

## Kano Model

Classify features into Basic Needs, Performance Needs, and Excitement Needs to enhance customer satisfaction

### 1. Understanding Customer Satisfaction and impact

-  **Basic Needs:** Fundamental requirements that customers expect; their absence leads to dissatisfaction.
-  **Performance Needs:** Features that increase satisfaction when fulfilled but don't cause dissatisfaction when absent; these directly affect user experience.
-  **Excitement Needs:** Unexpected features that can delight customers and significantly enhance satisfaction; these can differentiate a product in the market.

### 2. Importance of Basic Needs

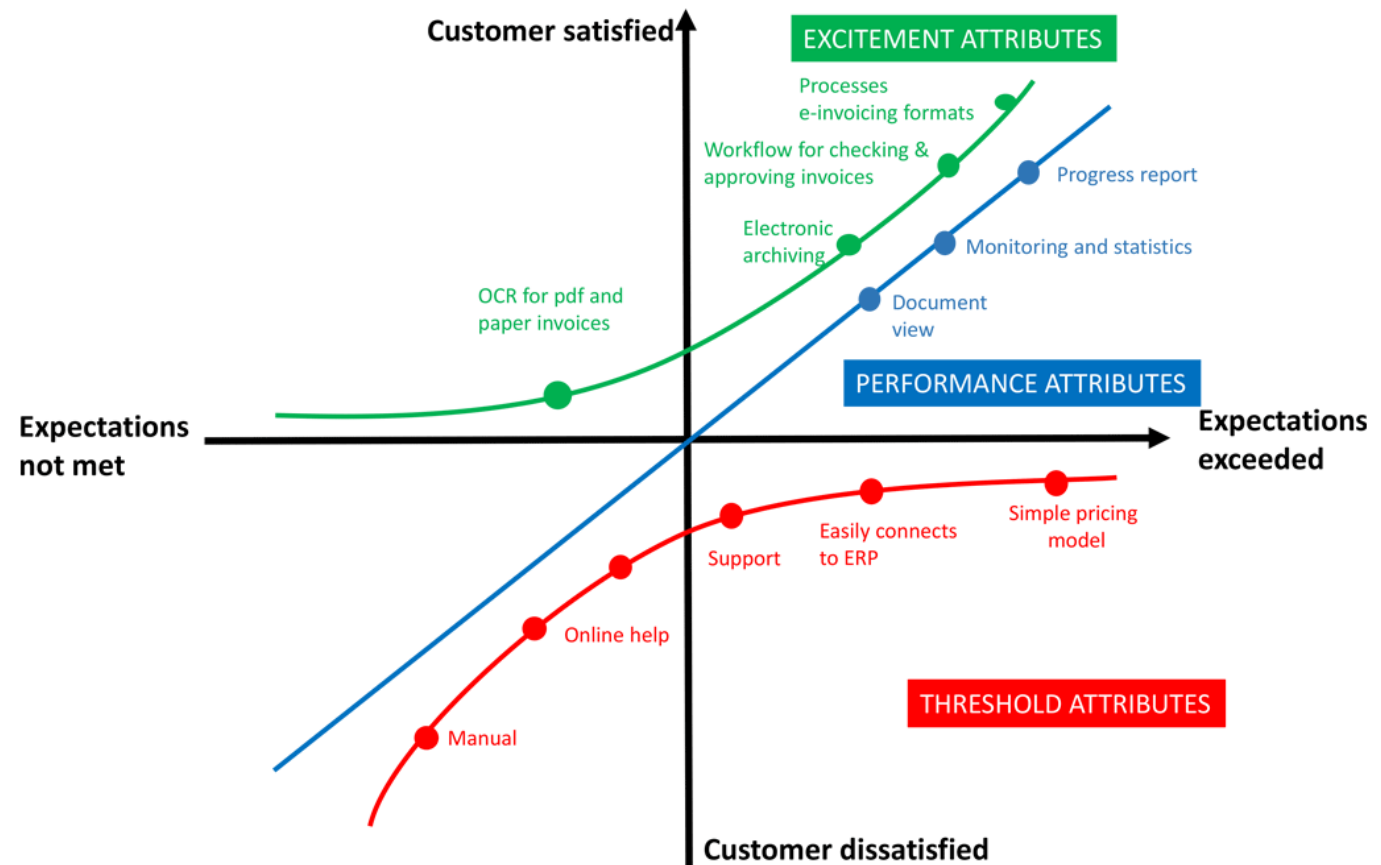
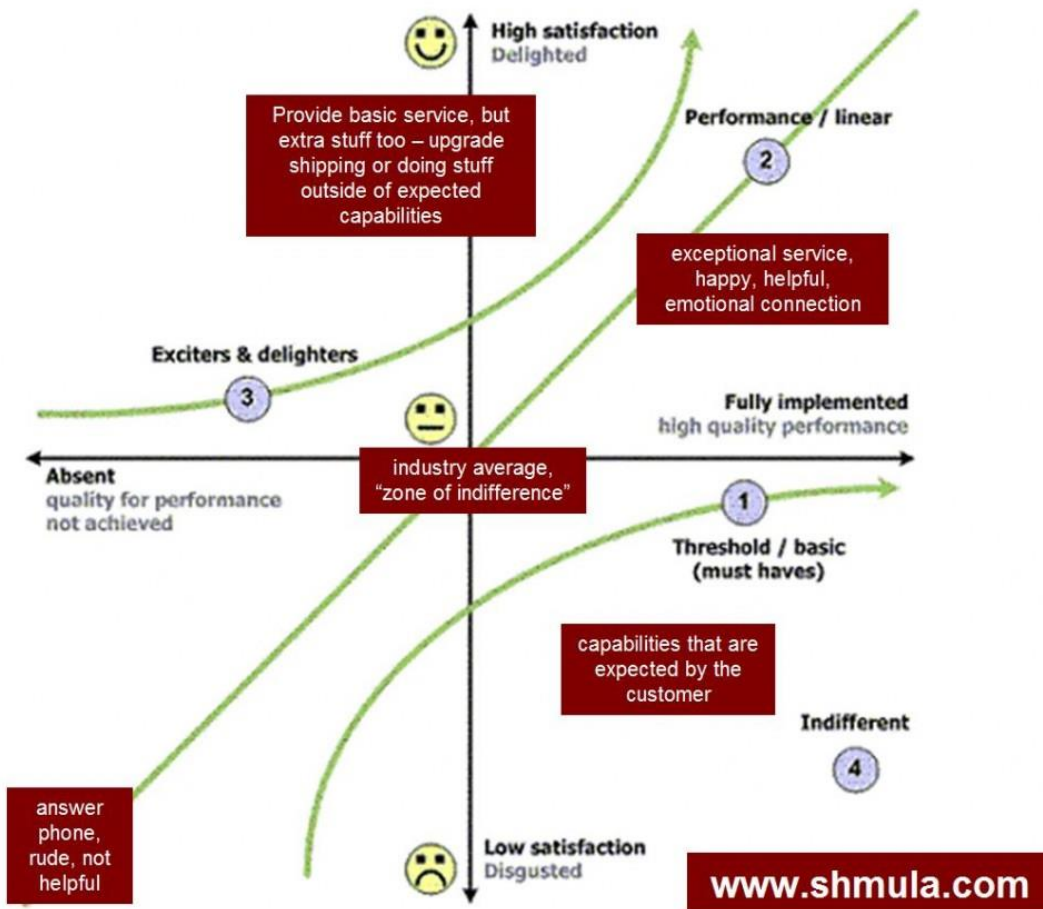
- Meeting Basic Needs is crucial for maintaining customer loyalty and satisfaction; failing to deliver on these can lead to negative perceptions of the product or service.
- Identifying these needs early in the project helps ensure that foundational elements are prioritized and delivered effectively.

### 3. Leveraging Excitement Needs for Competitive Advantage

- Focusing on Excitement Needs allows teams to create memorable experiences that can set their product apart from competitors.
- By identifying and incorporating these delightful features, teams can foster customer loyalty and enhance overall satisfaction.
- The Kano Model encourages teams to continuously seek feedback from users to identify potential excitement features that could be added in future iterations.

# Kano Example

<https://blog.seeburger.com/what-is-the-kano-model/>





# Scenario- Based Requirements

Scenario-based requirements capture user needs in context, providing a clearer understanding of user interactions

## 1. Definition of Scenario-Based Requirements:



- Scenario-based requirements involve capturing user needs through real-life scenarios that illustrate how users interact with a system.

## 2. Types of Scenarios:



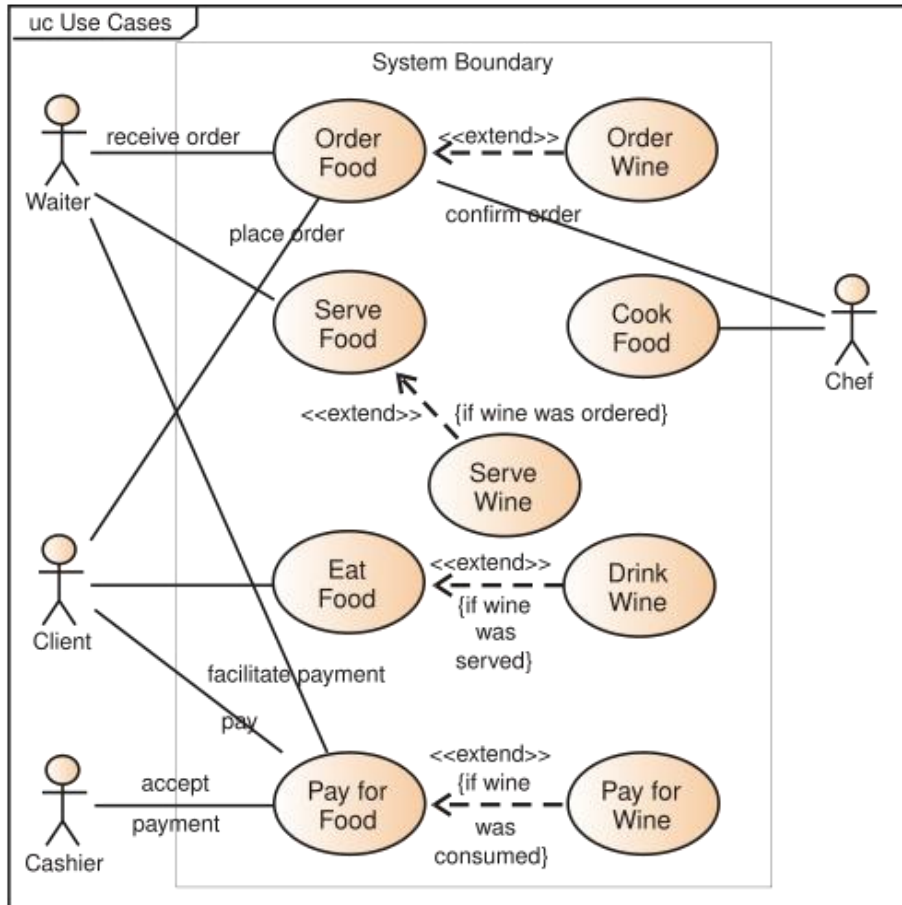
- **Use Cases:** Describe interactions between users (actors) and the system to achieve specific goals.
- **User Stories:** Short, simple descriptions of a feature from the perspective of the user.
- **Job Stories:** Focus on the situation and motivation behind a user's action rather than just the feature itself.

## 3. Benefits of Using Scenarios to Capture Requirements:



- Provide context for user interactions, making it easier to understand user needs.
- Facilitate communication among stakeholders by using relatable narratives.
- Help identify edge cases and exceptions that may not be captured in traditional requirements.

# Scenario-Based Requirements Examples



[https://en.wikipedia.org/wiki/Use\\_case\\_diagram](https://en.wikipedia.org/wiki/Use_case_diagram)

As a <user role>  
I want <goal>  
so that <benefit>.

<https://www.scrumwithstyle.com/effective-user-stories/>

<b>BLFS-228</b> As a customer I want to view my account balance so that I know my available funds.	<b>32</b>	<b>BLFS-40</b> As a fraud officer I want to see a report of failed logins.	<b>21</b>
<b>Customer Records</b>		<b>Fraud Detection</b>	
<b>BLFS-46</b> As a customer services representative I want to access historical account	<b>48</b>	<b>BLFS-95</b> As a bank I want to receive a correctly formatted transaction file	<b>81</b>

<https://www.equinox.co.nz/blog/jira-trello-user-stories-how-customise-print>

# Creating Use Cases for Requirement Elicitation

## Use cases help capture detailed user interactions and validate requirements

### 1. Components of a Use Case:

- **Actor:** Represents the user or system interacting with the application. This could be a single user, a team, or another system.
- **Goal:** The objective that the actor wants to achieve through their interaction with the system.
- **Steps:** A sequence of actions taken by the actor to accomplish their goal. Each step should clearly indicate who is performing it.
- **Outcome:** The expected result once all steps are completed successfully.

### 2. Writing Effective Use Cases

- Start with a clear title and description that outlines the purpose of the use case.
- Define each component (actor, goal, steps, outcome) in simple language.
- Include alternate flows to cover potential errors or variations in user actions. This ensures robustness in capturing all possible scenarios.

### 3. Using Use Cases to Validate Stakeholder Requirements

- Use cases serve as a validation tool by ensuring that stakeholder goals are accurately represented.
- Engage stakeholders in discussions about use cases to confirm their understanding and expectations.
- By mapping out interactions through use cases, teams can identify gaps in requirements and refine them accordingly.

# User Stories and Job Stories for Elicitation

## User stories and job stories are simple yet powerful tools for capturing user needs

### 1. Structure of a User Story:

- Format: "As a [user], I want [action], so that [goal]." This structure highlights:
  - **Who** is the user (e.g., customer, admin).
  - **What** action they want to perform.
  - **Why** they want to achieve this goal (the benefit).
- Example: "As an online shopper, I want to filter products by price so that I can find items within my budget."

### 2. Structure of a Job Story

- Format: "When [situation], I want to [motivation], so I can [outcome]."
- This format emphasizes:
  - The **context** or situation in which the action occurs.
  - The **motivation** driving the user's action.
  - The desired outcome or result of that action.
- Example: "When I am searching for shoes, I want to see customer reviews so that I can make an informed purchase decision."

### 3. Benefits of Using User Stories and Job Stories:

- Simplifies communication about user needs.
- Encourages collaboration among stakeholders.
- Provides clarity on user goals and motivations, leading to better design decisions.

# Validating Scenario- Based Requirements

Validating requirements ensures they accurately capture all user needs and exceptions

## 1. Review and Validation Techniques:

- **Stakeholder Reviews:** Involve stakeholders in reviewing scenarios to ensure they reflect their needs and expectations.
- **Walkthroughs:** Conduct step-by-step walkthroughs of scenarios with stakeholders to identify gaps or misunderstandings.
- **Simulations:** Use simulations to demonstrate how scenarios play out in real-time, allowing stakeholders to visualize interactions.

## 2. Refining Scenarios Based on Feedback:

- Gather feedback from stakeholders after reviews and walkthroughs.
- Update scenarios to address any identified gaps or issues.

## 3. Ensuring Scenarios Cover All Critical Paths and Exceptions:

- Validate that scenarios include all possible user interactions, including edge cases and exceptions.
- Ensure comprehensive coverage to prevent oversight of critical requirements..

# Summary and Q&A

## Recap of key concepts and an open floor for questions

### **1. Importance of Understanding Stakeholder Requirements:**

- Grasping stakeholder needs is crucial for the success of any project. It ensures that the final product meets user expectations and business objectives.

### **2. Scenario-Based Techniques for Capturing Requirements:**

- Utilizing techniques like use cases, user stories, and job stories helps to clarify user needs in context. These methods provide a structured way to capture interactions and motivations.

### **3. Validating Requirements Through Scenarios:**

- Validating scenario-based requirements through stakeholder reviews, walkthroughs, and simulations ensures that all critical paths and exceptions are covered, leading to more robust requirements.