

# ISM101 – Information Systems Management

## *Knowledge Check #4*

**Name:** Aldenir D Flauzino

**Student No.:** CT1010383

1. Explain the importance of gathering requirements in defining project scope and ensuring alignment with stakeholders' expectations. Provide an example to illustrate your point.

**A:** Gathering requirements is important to understanding the needs and constraints of a system or project. It helps ensure that the final product meets the stakeholders' expectations and aligns with the project's objectives. Without a clear understanding of these requirements, it's easy for projects to go off track, resulting in missed objectives, wasted resources, or dissatisfaction among stakeholders. Imagine you are developing a system to sell cell phones. The client simply says: "I want to create an app where I can sell smartphones". However, you know that vague requirements like that can lead to misunderstanding. By gathering these detailed requirements, you now have a well-defined scope for the project.

2. What are some common pitfalls when eliciting user needs, and how can project teams avoid these pitfalls to ensure user satisfaction?

**A:** Sometimes, users may not articulate their needs accurately, so it's important to dig deeper and understand the underlying problems they want to solve. Because of that it is important to dig deeper and understand the underlying problems they want to solve. To solve this problem is important: Use clear, specific questions, and engage users in detailed conversations or workshops to clarify needs. Prototyping or using mockups can also help users visualize and refine their requirements.

3. Describe the role of requirements traceability and management in ensuring that project requirements are properly implemented. Provide a real-world example to illustrate this concept.

**A:** Managing and tracking requirements throughout the project's lifecycle to ensure they are properly implemented and any changes are well-documented. Imagine you're working on a project to develop an

e-commerce application, to sell smartphones. The key stakeholders provided a list of high-level requirements, such as:

1. User must be logged in their account to pay;
2. Users should be able to view your orders;
3. Users should be able to pay with multiple credit cards
4. The system must adhere to stringent security standards (e.g., two-factor authentication, encryption).

**Traceability:** Each requirement is assigned a unique ID and is traced back to specific business objectives

**Managing Changes:** If in the middle of the development the client requests an additional feature, the change is reviewed, assessed for feasibility, and prioritized according to its impact on the project's timeline and budget.

**Testing and Validation:** During the testing phase, each requirement is mapped to specific test cases to ensure it has been properly implemented.

4. In the context of analyzing stakeholder expectations, how can project managers effectively prioritize stakeholders and manage conflicting expectations to ensure project success?

**A:** Prioritizing stakeholders based on their influence, impact, or level of interest in the project. To manage conflicting expectations, there are a number of techniques such as:

1. **Negotiate and balance** competing interests by finding common ground or compromises.
2. **Communicate transparently** with stakeholders about limitations, trade-offs, and project constraints (time, budget, resources).
3. **Set clear priorities and focus** on delivering the most critical needs that align with project goals.

By understanding the needs of stakeholders, managing conflicts, and continuously communicating, project managers can ensure that the most important expectations are met

5. Using an analogy or real-life example, explain why decomposing high-level requirements into functional specifications is essential in software or system development. What benefits does it offer to the project team?

A: Decomposing high-level requirements into functional specifications is like breaking down a recipe into individual steps for cooking.

Imagine you're planning to make a complex dish, like a gourmet lasagna. The high-level requirement is simple: "Make a lasagna." But to successfully prepare it, you need to break it down into functional specifications like:

Prepare the pasta sheets: Boil water, cook the pasta.

Prepare the meat sauce: Brown the meat, sauté onions, add tomatoes.

Assemble layers: Layer pasta, sauce, cheese in a baking dish.

Without breaking it down, you'd be left with unclear directions, and things might get missed or go wrong.

In software or system development, decomposing high-level requirements into detailed specifications ensures everyone on the team knows exactly what to do and when, leading to a smoother, more successful project.