

Discussion-2 key challenges of requirements analysis, and how can these challenges be overcome

Discussion Topic:

Please choose one of the following questions to discuss in your initial post:

What are the different types of software development process models, and what are the advantages and disadvantages of each?

What are the key challenges of requirements analysis, and how can these challenges be overcome?

What are the essential elements of a well-written software design document?

My Post:

Hello Class,

For this discussion, I chose the following topic:

What are the key challenges of requirements analysis, and how can these challenges be overcome?

The key challenges of requirements analysis are to identify and define requirements. These challenges are derived from communication issues with stakeholders and the technical aspects of the software development itself.

Communication Issues with Stakeholders

Gathering information/requirements from people is one of the first issues software developers face. The challenges of gathering requirements from people include the following issues:

- Stakeholders often do not know what they really want.
- Stakeholders may express requirements in their own terms, which can be unclear for or misinterpreted by developers. For instance, terms like “fast” or “easy to use” can be interpreted differently by stakeholders and developers. For example, a “fast” and “easy to use” feature (functionality) expressed without an exact metric can mean, for stakeholders, that the feature should be simple (not complex), requiring only a few steps to perform and easy for users to learn. On the other hand, the same statement can be interpreted by developers as the feature should have “fast” response times, not necessarily having few steps, and the UI/components of the feature, although complex, are intuitive to use, not necessarily easy to learn.
- Different sets of stakeholders can have conflicting requirements or opposite requirements.
- The requirements can change over time, that is, during the process of gathering requirements and/or during the process of analyzing them, especially as new stakeholders emerge.

(CSU Global, 2025)

Technical Aspects of Software Development Issues

Requirements analysis focuses not only on the domain requirements (what the software must do), but also on nonfunctional requirements, that is, the components or aspects of the software that are independent of the problem domain (of the direct functionality of the software). As pointed out by Richards and Ford (2020), within our course textbook, while these requirements are often called

“nonfunctional requirements,” that term can be unhelpful because it has a negative connotation that diminishes its importance. A more descriptive term is “architectural characteristics.” The architecture characteristics of an application are often implicit and not often described in a requirements document. Nonetheless, characteristics like availability, reliability, and security are necessary for the success of almost all applications. Additionally, these requirements are rarely mentioned by stakeholders, making them, in certain circumstances, challenging to identify and analyze.

How These Challenges Can Be Overcome

First, it is important to create a requirements document stating “what” the application will do, NOT ‘how’ it will do it (CSU Global, 2025). The requirement should be clearly defined and detailed; feasible to change, correct, consistent, precise, understandable, and testable.

Secondly, use iterative processes, that is, using methodologies such as Agile. This is important as it establishes a robust line of communication between stakeholders and developers. Additionally, it allows for flexibility in the development process, as requirements can change often and quickly, and stakeholders are often unsure initially about what they really want. (CSU Global, 2025). Rigid development models like the waterfall model are not suitable for this task.

Finally, managing architecture trade-offs is essential. It is important to recognize that an application cannot fully support every possible architecture characteristic, such as performance, scalability, security, and maintainability. The goal should never be to shoot for the “best” architecture, but rather the “least worst architecture” that balances architecture trade-offs and architecture manageability with the system requirements (Richards & Ford, 2020).

References:

CSU Global. (2025). Module 2: Project Requirements and Specifications [Interactive Lecture]. Canvas. https://csuglobal.instructure.com/courses/110425/pages/module-2-overview?module_item_id=5733175

Richards, M., & Ford, N. (2020). Chapter 1: Introduction. Fundamentals of software architecture: An engineering approach (Softcover). O'Reilly Media. ISBN-13: 978-1-492-04345-4