UNIVERSITY OF MARYLAND
University College

Learning Resource

Requirements

What Are Requirements?

For purposes of this class, we will focus on what the end user needs or expects the system to do. These needs and expectations are documented as **requirements** for the system. They fall into two general categories: user requirements (sometimes referred to as *functional requirements*) and system performance requirements (sometimes referred to *non-functional requirements*).

1. User Requirements describe the tasks the user needs the system to perform, such as:

- What data the system is expected to collect.
- What the system is expected to do with the data that is input.
- What the system is expected to provide as output (reports, results, etc.).

Some example user requirements for an online shopping site might be:

- The system must calculate the total of all items in the online or website shopping cart.
- The system must display to the user similar items that the online shopper may be interested in.
- The system must require the user to provide a shipping address.
- The system must automatically fill in the State portion of the shipping address based on the zip code entered by the user.
- The system must provide the user with a report of all purchases made via the website.

2. System Performance Requirements are sometimes referred to as *system quality attributes*, since they define how the system is designed, how it will perform when used, and what the user experience will be (Microsoft, 2009).

They describe how the system will perform, or its quality, in areas such as:

- **Usability**—The ability for new users to quickly adapt to the software, including how easy the system is to use and how help is provided for the users
- **Scalability**—The ability of the system to accommodate additional users and/or additional records/transactions
- **Availability**—The amount or periods of time the system is to be operational and useable
- **Reliability**—The ability of the system to create and maintain the data correctly
- **Maintainability**—The ability of the system to be easily maintained, corrected and updated
- **Performance**—The ability of the system to meet time or volume requirements (respond to user inquiry, update a database, or handle the workload)
- **Portability**—The ability of the system to run/operate on a variety of end-user devices or with multiple operating systems
- **Interoperability**—The ability of the system to interact with other existing or legacy systems

System performance requirements also describe security requirements for the system and data, such as:

- Protection of the system from malicious or accidental actions
- Protection of data as it is transmitted and when it is stored
- User authentication; prevention of unauthorized access
- Authorization of users to perform specific functions; prevention of unauthorized changes to data
- Data backup and recovery

Some examples of system performance requirements are:

- The system must encrypt the user's payment information when it is transmitted.
- The system must require a retinal scan for login purposes.
- The system must be capable of handling 5,000,000 transactions per hour.
- The system must operate using Motorola handheld scanners.
- The system must be able to accept financial data directly from the company’s financial system.

To differentiate between user and system performance requirements, the business analyst determines whether each requirement describes a task that the system must perform (user requirement) or describes system quality or security (system performance requirement).

How Are the Requirements Used?

Requirements can be used to develop a system from scratch, in which case many detailed requirements for every step of every process need to be clearly laid out. For example, if an accounting system is to be developed, the developers will need to incorporate all the financial and legal aspects of the process. They will need to know exactly how each accounting function is to be performed in order to program the system to carry out the function.

However, if the intent is to acquire a commercial off-the-shelf (COTS) accounting system or to use a software-as-a-service (SaaS) system, then the requirements may be stated at a much higher level, such as: "the system must implement the Generally Accepted Accounting Principles (GAAP)" or "the system must produce a monthly expense statement." In these cases, the end user is not so concerned about each step in performing those functions, as long as the system provides them.

Once the requirements are listed, they can be used to:

- Develop a system and test it to be sure it meets the requirements
- Identify one or more COTS or SaaS systems that appear to meet the requirements
- Test the COTS or SaaS systems to determine which one meets the most requirements and select one for use
- Identify requirements that are not met that may need be added to the system or may require a separate or additional system(s) or processes to be implemented

According to Mitre (2018) requirements "can be tested, verified, and/or validated, and are unique, complete, unambiguous, consistent, and obtainable, and [can be traced] to original business and mission needs."

Documented requirements can be traced through an entire system development and implementation process. For example:

- They form the need for a system and define its scope (all the functions that are to be included).
- They form the basis for estimating the time and cost of developing or acquiring the system.
- They are used to develop the system.
- They are used to negotiate any requirements changes that are proposed by helping to determine how significant the change is.
- They are used to develop test cases to test the system to see if it functions as needed.

- They are used when modifications or enhancements are proposed to ensure that the new change does not unintentionally replace previous functionality, and that the new requirement fits within the scope of the system's overall functionality.
- They are used to test a modified system to ensure all previous functions, as well as the new functions, perform as needed.

References

Microsoft. (2009). Microsoft application architecture guide, 2009. Retrieved from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658094(v=pandp.10)

Mitre. (2018). Systems Engineering Guide—Analyzing and Defining Requirements. Retrieved from https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/requirements-engineering/analyzing-and-defining-requirements

All links to external sites were verified at the time of publication. UMUC is not responsible for the validity or integrity of information located at external sites.