# 4.2 Functional Requirements

Return to Requirements

Functional Requirements define the basic system behavior. Essentially, they are requirements stating what the system does or must not do, and can be thought of in terms of how the system responds to inputs. Functional requirements usually define if/then behaviors and include calculations, data input, and business processes.

Functional Requirements are features that allow the system to function as it was intended. Put another way, if the functional requirements are not met, the system will not work. Functional requirements are product features and focus on user requirements. Functional Requirements can be used during all phases of a project Lifecycle independent of the development model (i.e., Waterfall Model or Agile Model). In the Waterfall method, these requirements are generally specified early on in the process. In the Agile method, they can be applied throughout each Sprint or applied during specific Sprints.

## Things to Consider When Designing Distributed systems

- **Note:** Because these are Functional Requirements, they need to be written specifically for the system being specified in the proposal or solicitation.

In DIDO, the following areas minimally cover Functional Requirements:

### Hardware Architecture

The goal of hardware functional requirements is to decide which hardware architectures are going to be supported and which, if any, will be left out.

- Which Platforms must be supported?
- Which Embedded Systems must be supported?
    - Which Standalone Subsystems must be supported? | |
    - Which Standalone Systems must be supported? | |
    - Which Network Systems must be supported? | |
3. Which servers must be supported?
    - Which software Servers must be supported?
    - Which hardware Servers must be supported?
4. Which desktop computers must be supported?
5. Which handheld computers must be supported?
6. Which supercomputers must be supported?
7. Which network computers must be supported?

### Operating System

Nodes on the distributed network must be designed specifically for the operating system that runs on the node. So the distributed system needs to support more than one operating system. But too many OSs can cause the support and maintenance of the system to be more difficult and costly.

- Which operating systems must be supported?

### Network

A distributed system is a collection of networked nodes, that run on networking equipment all connected using wireless or USB connections. Network functional requirements is to decide which platform, devices, and permissions are used throughout the system.

- Which network platforms must be supported?
- Which networking architectures must be supported?
- Which hnetworking access must be supported?

### RunTime Libraries

RunTime functional requirements is to establish a RunTime Library that can communicate to with different operating/hardware platforms across the system.

- Which runtime environments must be supported?

### Community/Development Tools

Tool functional requirements include making sure your system has all the necessary development, management, and operation tools for the project.

- Which support tools must be supported?

**Note**: The tool list provided is not exhaustive.

### Virtualized Nodes

Utilizing Virtual Nodes in a distributed system can provide flexibility because a virtual node can run on many hardware/operating system platforms. Virtual Node functional requirements entail determining which type of virtual node is necessary for the system.

- Which types of virtualization nodes must be supported?

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
**https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:1_func&rev=1616424912**

Last update: **2021/03/22 10:55**