FUNCTIONAL REQUIREMENTS DOCUMENT

Overview

The functional requirements document (FRD) is a formal statement of an application's functional requirements. It serves the same purpose as a contract. The developers agree to provide the capabilities specified. The client agrees to find the product satisfactory if it provides the capabilities specified in the FRD.

Quality is meeting requirements. For that reason, the FRD is the central document in system development. It is used for the following:

- Designing and developing tile application system.
- Evaluating the product in all subsequent phases of the life cycle.
- Determining the success of the project.

The FRD has the following characteristics:

- It demonstrates that the application provides value to the State in terms of the business objectives and business processes in the 5-year plan.
- It contains a complete set of requirements for the application. It leaves no room for anyone to assume anything not stated in the FRD.
- It is solution independent. The ERD is a statement of what the application is to do—not of how it works. The FRD does not commit the developers to a design. For that reason, any reference to the use of a specific technology is entirely inappropriate in an FRD.

A requirement is a condition that the application must meet for the customer to find the application satisfactory. A requirement has the following characteristics:

- It provides a benefit to the organization.
- It describes the capabilities the application must provide in business terms.
- It does not describe how the application provides that capability.
- It does not describe such design considerations as computer hardware, operating system, and database design.
- It is stated in unambiguous words. Its meaning is clear and understandable.
- It is verifiable.

1 GENERAL

1.1 Project Description

Provide a brief overview of the project.

1.1.1 Background

Summarize the conditions that created the need for the application.

1.1.2 Purpose

Describe the business objectives and business processes from the CONOPS document and the CBA that this application supports.

1.1.3 Assumptions and Constraints

Assumptions are future situations, beyond the control of the project, whose outcomes influence the success of a project. The following are examples of assumptions:

- Availability of a hardware/software platform
- Pending legislation
- Court decisions that have not been rendered
- Developments in technology

Constraints are conditions outside the control of the project that limit the design alternatives. The following are examples of constraints:

- Government regulations
- Standards imposed on the solution
- Strategic decisions

Be careful to distinguish constraints from preferences. Constraints exist because of real business conditions. Preferences are arbitrary. For example, a delivery date is a constraint only if there are real business consequences that can happen as a result of not meeting the date. For example, if failing to have the subject application operational by the specified date places the State in legal default, the date is a constraint. A date chosen arbitrarily is a preference. Preferences, if included in the FRD, should be noted as such.

1.1.4 Interfaces to External Systems

Name the applications with which the subject application must interface, State the following for each such application:

- Owner of application (if external to the Agency)
- Details of interface (only if determined by the other application)

1.2 Points of Contact

List the names, titles, and roles of the major participants in the project. At a minimum, list the following:

- Project Manager
- Development project leader
- User contacts
- Agency employee whose signature constitutes acceptance of the FRD

1.3 Document References

Name the documents that were sources of this version of the FRD. Include meeting summaries, white paper analyses, and other System Development Life Cycle deliverables, as well as any other documents that contributed to the FRD. Include the Configuration Management identifier and date published for each document listed.

2 FUNCTIONAL REQUIREMENTS

The functional requirements describe the core functionality of the application. This section includes the data and functional process requirements.

2.1 Data Requirements

Describe the data requirements by producing a logical data model, which consists of entity relationship diagrams, entity definitions, and attribute definitions. This is called the application data model. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database.

2.2 Functional Process Requirements

Process requirements describe what the application must do. Process requirements relate the entities and attributes from the data requirements to the users' needs.

State the functional process requirements in a manner that enables the reader to see broad concepts decomposed into layers of increasing detail.

Process requirements may be expressed using data flow diagrams, text, or any technique that provides the following information about the processes performed by the application:

- Context
- Detailed view of the processes
- Data (attributes) input to and output from processes
- Logic used inside the processes to manipulate data
- Accesses to stored data
- Processes decomposed into finer levels of detail

3 OPERATIONAL REQUIREMENTS

Operational requirements describe the non-business characteristics of an application.

State the requirements in this section. Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be?” Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs. Preferences are not. If, for example, the user expresses a desire for sub-second response but does not have a business-related reason for needing it, that desire is a preference.

3.1 Security

The security Section describes the need to control access to the data. This includes controlling who may view and alter application data.

State the consequences of the following breaches of security in the subject application:

- Erasure of contamination of application data
- Disclosure of Government secrets
- Disclosure of privileged information about individuals

State the type(s) of security required. Include the need for the following as appropriate:

State if there is a need to control access to the facility housing the application.

- State the need to control access by class of users. For example, “No user may access any part of this application who does not have at least a (specified) clearance.”
- State the need to control access by data attribute. State, for example, if one group of users may view an attribute but may not update it while another type of user may update or view it.
- State the need to control access based on system function. State for example, if there is a need to grant one type of user access to certain system functions but not to others. For example, “This function is available only to the system administrator.”
- State if there is a need for accreditation of the security measures adopted for this application. For example, C2 protection must be certified by an independent authorized organization.

3.2 Audit Trail

List the activities that will be recorded in the application’s audit trail. For each activity, list the data to be recorded.

3.3 Data Currency

Data currency is a measure of how recent data are. This section answers the question, “When the application responds to a request for data how current must those data be?” Answer that question for each type of data request.

3.4 Reliability

Reliability is the probability that the system will be able to process work correctly and completely without being aborted.

State the following in this section:

- What damage can result from this system's failure?
  - Loss of human life
  - Complete or partial loss of the ability to perform a mission-critical function
  - Loss of revenue
  - Loss of employee productivity

- What is the minimum acceptable level of reliability?

State required reliability in any of the following ways:

- Mean Time Between Failure is the number of time units the system is operable before the first failure occurs.
- Mean Time To Failure is computed as the number of time units before the system is operable divided by the number of failures during the time period.
- Mean Time To Repair is computed as the number of time units required to perform system repair divided by the number of repairs during the time period.

3.5 Recoverability

Recoverability is the ability to restore function and data in the event of a failure. Answer the following questions in this section:

- In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?
- In the event the database is corrupted, to what level of currency must it be restored? For example "The database must be capable of being restored to its condition on no more than one hour before the corruption occurred."
- If the process site (hardware, data, and onsite backup) is destroyed how soon must the application be able to be restored?

3.6 System Availability

System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed. In this section state the hours (including time zone) during which the application is to be available to users. For example, "The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST." If the application must be available to users in more than one time zone state the earliest start time and the latest stop time.

Include the times when usage is expected to be at its peak. These are times when system unavailability is least acceptable.

3.7 Fault Tolerance

Fault tolerance is the ability to remain partially operational during a failure. Describe the following in this section:

- Which functions need not be available at all times?
- If a component fails what (if any) functions must the application continue to provide? What level of performance degradation is acceptable?

For most applications, there are no fault tolerance requirements. When a portion of the application is unavailable there is no need to be able to use the remainder for the application.

3.8 Performance

Describe the requirements for the following:

- Response time for queries and updates
- Throughput
- Expected volume of data
- Expected volume of user activity (for example, number of transactions per hour, day, or month)

3.9 Capacity

List the required capacities and expected volumes of data in business terms. For example, state the number of cases about which the application will have to store data. For example, “The project volume is 600 applications for naturalization per month.” State capacities in terms of the business. Do not state capacities in terms of system memory requirements or disk space.

3.10 Data Retention

Describe the length of time the data must be retained. For example, “information about an application for naturalization must be retained in immediately accessible from for three years after receipt of the application.”

4 REQUIREMENTS TRACE ABILITY MATRIX

The requirements trace ability matrix (RTM) provides a method for tracking the functional requirements and their implementation through the development process. Each requirement is included in the matrix along with its associated section number. As the project progresses, the RIM is updated to reflect each requirement’s status. When the product is ready for system testing, the matrix lists each requirement, what product component addresses it, and what test verify that it is correctly implemented.

Include columns for each of the following in the RTM:

- Requirement description
- Requirement reference in FRD
- Verification Method

- Requirement reference in Test Plan

5 GLOSSARY

Include business terms peculiar to the application. Do not include any technology-related terms.