Functional specification

A **functional specification** (also, *functional spec, specs, functional specifications document (FSD), functional requirements specification*) in systems engineering and software development is a document that specifies the functions that a system or component must perform (often part of a requirements specification) (ISO/IEC/IEEE 24765-2010).[1]

The documentation typically describes what is needed by the system user as well as requested properties of inputs and outputs (e.g. of the software system). A functional specification is the more technical response to a matching requirements document, e.g. the Product Requirements Document "PRD". Thus it picks up the results of the requirements analysis stage. On more complex systems multiple levels of functional specifications will typically nest to each other, e.g. on the system level, on the module level and on the level of technical details.
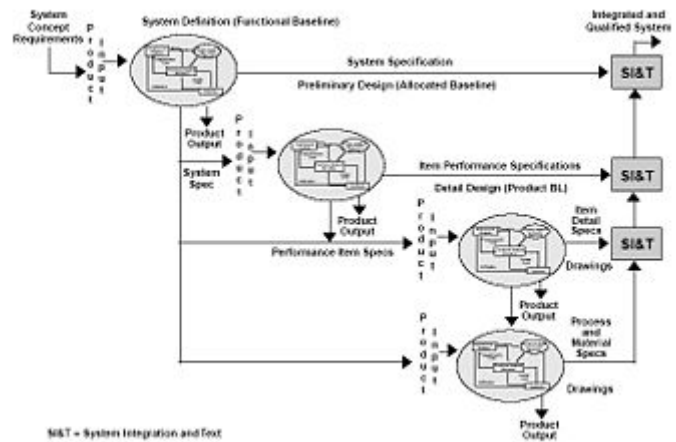
Systems engineering model of Specification and Levels of Development. During system development a series of specifications are generated to describe the system at different levels of detail. These program unique specifications form the core of the configuration baselines. As shown here, in addition to referring to different levels within the system hierarchy, these baselines are defined at different phases of the design process.Note: There is one minor (and ironic) typo in the image above. SI&T is "System Integration and Test" not "System Integration and Text".

Contents

Overview

A <u>functional</u> specification does not define the inner workings of the proposed system; it does not include the specification of how the system function will be implemented. Instead, it focuses on what various outside agents (people using the program, computer peripherals, or other computers, for example) might "observe" when interacting with the system.

A functional requirement in a functional specification might state as follows:

> *When the user clicks the OK button, the dialog is closed and the focus is returned to the main window in the state it was in before this dialog was displayed.*

Such a requirement describes an interaction between an external agent (the <u>user</u>) and the software system. When the user provides input to the system by clicking the OK button, the program responds (or should respond) by closing the dialog window containing the OK button.

Functional specification topics

Purpose

There are many purposes for functional specifications. One of the primary purposes on team projects is to achieve some form of team consensus on what the program is to achieve before making the more time-consuming effort of writing <u>source code</u> and <u>test cases</u>, followed by a period of <u>debugging</u>. Typically, such consensus is reached after one or more reviews by the <u>stakeholders</u> on the project at hand after having negotiated a cost-effective way to achieve the requirements the software needs to fulfill.

1. To let the <u>developers</u> know what to build.
2. To let the <u>testers</u> know what tests to run.
3. To let <u>stakeholders</u> know what they are getting.

Process

In the ordered industrial software engineering life-cycle (<u>waterfall model</u>), functional specification describes *what* has to be implemented. The next, <u>Systems architecture</u> document describes *how* the functions will be realized using a chosen software environment. In non industrial, prototypical systems development, functional specifications are typically written after or as part of <u>requirements analysis</u>.

When the team agrees that functional specification consensus is reached, the functional spec is typically declared "complete" or "signed off". After this, typically the software development and testing team write source code and test cases using the functional specification as the reference. While testing is performed, the behavior of the program is compared against the expected behavior as defined in the functional specification.

Methods

One popular method of writing a functional specification document involves drawing or rendering either simple wire frames or accurate, graphically designed UI screenshots. After this has been completed, and the screen examples are approved by all stakeholders, graphical elements can be numbered and written instructions can be added for each number on the screen example. For example, a login screen can have the

username field labeled '1' and password field labeled '2,' and then each number can be declared in writing, for use by software engineers and later for beta testing purposes to ensure that functionality is as intended. The benefit of this method is that countless additional details can be attached to the screen examples.

Examples of functional specifications

- Advanced Microcontroller Bus Architecture
- Extensible Firmware Interface
- Multiboot Specification
- Real-time specification for Java
- Single UNIX Specification

Types of software development specifications

- Bit specification (disambiguation)
- Design specification
- Diagnostic design specification
- Product design specification
- Software Requirements Specification

See also

- Benchmarking
- Software development process
- Specification (technical standard)
- Verification and Validation (software)

References

1. ISO/IEC/IEEE 24765:2010 Systems and software engineering — Vocabulary (http://www.iso.org/iso/catalogue_detail.htm?csnumber=50518)

External links

- Painless Functional Specifications, 4-part series by Joel Spolsky (http://www.joelonsoftware.com/articles/fog0000000036.html)