



G.F.S. Computing, Inc.

Functional Requirements Specification **“The Misunderstood Document”**

Introduction

After years of collective experience and many, many control projects, we believe the success of a project lies in the definition of the requirements for the control system. This definition is usually contained within several documents. The typical project starts with the Piping and Instrument Drawings (P&ID's), the Input/Output (I/O) List and the Process Description. These documents are usually consolidated into the Functional Requirements Specification (FRS), which is then sent to system vendors and integrators for bid. At this point these four documents become the contractual scope of the project and the base line from which change orders will flow.

Successful projects typically have functional requirements that are complete, consistent and accurate. Those projects that are not so successful will have an FRS that lacks one or more of these characteristics. To completely understand, we must think about the purpose, the key characteristics and the audience of the FRS.

Purpose

The key to a well-written FRS is its focus on the purpose of the document. The FRS document serves three purposes within the scope of a control project:

- Functional Description
- Communication
- Scope Definition

The first purpose of the FRS is to define the functionality required in the delivered control system. FRS's should be limited to describing what the system will do. They should not describe how the system will be designed or implemented. A litmus test for this is to look for system-specific language in the FRS. A well-written FRS can be implemented on any one of the many control systems available on the market today.

The second purpose of the FRS is to communicate the Process Engineer's intent concerning operation of the process to Plant Operations, Project Control Engineers and, in a regulated industry, to Quality/Validation personnel and government regulators. A common mistake made in development of an FRS is to write the document to one audience at the exclusion of the others. Before the FRS is written, the different segments of the audience should be ranked by priority. This priority ranking is extremely valuable



G.F.S. Computing, Inc.

in answering questions about format, wording and content which arise during FRS development.

The third purpose of the FRS is to define the scope of the project. Control application budgets are typically determined by counts of process control objects and the perceived complexity of the overall project. These estimates are only as good as the documents that drive the counts and the perception of complexity. A well-written FRS will minimize the number of change orders generated during a project, as well as the pain that accompanies negotiation and justification of these changes. As a definition of the scope of the project, the FRS should also be the basis of Factory Acceptance Testing. A well-written FRS should leave no doubt as to required functionality. Thus, at FAT time, the number of surprises should be minimized.

Key Characteristics

Assuming that the FRS was written with the above-mentioned three purposes in mind, how does one determine a well written FRS from those that are not well written? There are four characteristics that separate a quality FRS from the rest:

- Functionality Focus
- Completeness
- Accuracy
- Testability

The most important characteristic is focus on functionality. If the document focuses on what the control system needs to do, it possesses this characteristic. A good test for this is to review the document looking for design-type language such as:

- “... Set the following flag”
- “Wait for cycle timer to timeout”
- “Download cycle timer setpoint”

It is important to remember that the FRS is not a design document. It is a functional document. A well-written FRS can be used to develop an application on any system and would not have to be modified to adjust for nuances in the system of choice. See Figures I and II for examples from actual FRS's.

The next characteristic that all well-written FRS's must have is completeness. Since this document is used as the scope of the project, making it as complete as possible is beneficial both to the end user and to the firm writing the application. A complete FRS describes not only the normal operation of the plant but also the exceptions to this normal operation. The following points should be kept in mind when writing an FRS:

- Include detailed description of interlocks.



G.F.S. Computing, Inc.

- Avoid vague descriptions of actions. For example when waiting for a level to reach target level, the FRS should specify whether the wait should be for the level to drop below the target or rise above the target and also be specific as to the exact level that is being monitored.
- Include a description of field values to be monitored during sequential operations.
- Include a description of the States and Transition required for sequential logic.
- Include a description of logic that is required when traveling from one state to the next for each transition described above.

All input documents to the process control development must be accurate. Figure III illustrates this point. In the figure, each ring represents the content of each of the input documents. The intersection of the three circles represents the accuracy of the documents. In a well-written set of input documents, the intersected area is maximized. The three documents should be continually reviewed for accuracy in order to avoid confusing the audience.

The final characteristic is testability. All functionality described in the FRS must be testable, and it should be testable by people other than those involved in the development. An FAT test protocol can easily be written from a well-written FRS.

Results

Because well-written FRS's fully define the scope of the project, large efficiencies in development of the control system can be realized. Listed below are just a few examples of these productivity increases:

- Fewer questions by the implementation engineers lead to more productivity.
- Consistent documentation allows engineers to float from area to area and project to project with ease.
- Less experienced engineers are able to do more on the project.

During a recent project using well written FRS's, we saw the following results on a validated pharmaceutical project:

- Reduction in Development Time – The initial estimate was that it would take four months to complete and test the software for a given area. With a well-written FRS we saw these improvements over this estimate:

	> 120 Days	< 120 Days
Number of Areas	11	16
%	40%	60%
Budget – Actual	-17 Hours	120 Hours
Average Days		90 Days



G.F.S. Computing, Inc.

- Reduction in Man Hours – By continually improving the work processes of the group, the number of required man hours were decreased. These improvements to the process were made possible by the well written FRS's.

	Unit #1	Unit #2	Unit #3
Days	94	88	46
Budgeted Hours	1534	1036	525
Actual Hours	1145	803	282
Control Modules	176	132	20
Phases	73	32	9
Units	1	1	1

- Changes were minimized – Because of the depth of information available to the control engineer, changes to the scope of the control system were easily identified. Because this was a pharmaceutical project, the FRS's were kept current during the development process. Also, because the FRS's were written consistently, engineers were able to re-adjust the layout of the sequence logic to deliver fewer Phases, thus saving the user money on implementation and, in the long run, on maintenance of the system.
 - FRS's were revised twice during the development process.
 - Twelve change orders were submitted on a 30,000 man hour application effort.
 - The net change to the final budget based on the change orders was -2% with additional functionality being added to the job.

Summary

The Functional Requirements Specification is the pivotal document for a successful control system application. If the FRS is written with functional focus, with the full audience in mind and with the proper level of detail to define the scope of the project, the risk associated with an automation project can be greatly reduced if not eliminated altogether.



G.F.S. Computing, Inc.

Poorly Written FRS

12. Circuits 101A, 101B, 101C and 101D repeat the following sequence until RtnFlowCheck is true:
- a) Download cycle timer setpoint from cyc_time1 and start timer.
 - b) Wait for cycle timer to time out.
 - c) Reset cycle timer.
 - d) Open MediaXfrBrnch3.
 - e) Wait for the valve to open.
 - f) Close the following valves:
 - 1) MediaXfrSIP3
 - 2) MediaXfrBrnch4
 - g) Download cycle timer setpoint from cyc_time2 and start timer.
 - h) Wait for cycle timer to time out.
 - i) Reset cycle timer.
 - j) Open MediaXfrBrnch2.
 - k) Wait for the valve to open.
 - l) Close the following valves:
 - 1) MediaXfrSIP2
 - 2) MediaXfrBrnch3
 - m) Download cycle timer setpoint from cyc_time3 and start timer.
 - n) Wait for cycle timer to time out.
 - o) Reset cycle timer.
 - p) Open MediaXfrBrnch1

Figure I



Same FRS Written Well

5.3.1.5.1 Phase Parameters

This table lists the parameters used in this phase:

Description	Variable Name
HI ALARM LIMIT FOR RINSE PRESSURE	<u>rise_pres_halm</u>
LOW ALARM LIMIT FOR RINSE PRESSURE	<u>rise_pres_lalm</u>
HI ALARM LIMIT FOR RINSE TEMPERATURE	<u>ret_temp_halm</u>
LOW ALARM LIMIT FOR RINSE TEMPERATURE	<u>ret_temp_lalm</u>
CIP SUPPLY FLOW RATE HIGH	<u>CIP_flow_rate_h</u>
CIP SUPPLY FLOW RATE LOW	<u>CIP_flow_rate_l</u>
CIP RETURN TIME	<u>CIP_ret_time</u>
TARGET CIP FLOW RATE 1	<u>CIP_flow_rate_1</u>
TARGET CIP FLOW RATE 2	<u>CIP_flow_rate_2</u>
TARGET CIP FLOW RATE 3	<u>CIP_flow_rate_3</u>
TARGET CIP FLOW RATE 4	<u>CIP_flow_rate_4</u>
TARGET CIP FLOW RATE 5	<u>CIP_flow_rate_5</u>
TARGET CIP FLOW RATE 6	<u>CIP_flow_rate_6</u>
TARGET CIP FLOW RATE 7	<u>CIP_flow_rate_7</u>
TARGET CIP FLOW RATE 8	<u>CIP_flow_rate_8</u>
TARGET CIP FLOW RATE 9	<u>CIP_flow_rate_9</u>
TARGET CIP FLOW RATE 10	<u>CIP_flow_rate_10</u>
TARGET CIP FLOW RATE 11	<u>CIP_flow_rate_11</u>
TARGET CIP FLOW RATE 12	<u>CIP_flow_rate_12</u>

5.3.1.5.2 Phase Logic

1. Change the CIP Flow Rate controller (FIC-950#18) setpoint to the CIP_flow_rate that corresponds to the CIP circuit number.
2. Open the following valves:

Description	Tag Template
CIP RINSE TANK OUTLET VALVE	HS-950#01
CIP SUPPLY BLOCK VALVE	HS-950#26
CIP PRIMARY RETURN PUMP DISCHARGE VALVE	HS-950#31
CIP DRAIN VALVE	HS-950#35

3. Start the CIP Supply Pump (VSC-950#18).
4. Start the CIP Return Pump (HS-950#45) as per Continuous Control section 3 of this document.
5. Wait until the CIP Return Flow limit switch (FI-950#32) indicates flow.
6. Wait for CIP_ret_time to establish return flow.
7. Set and enable the following alarms:

Description	Tag Template	Alarm Type	Value
CIP FLOW RATE CONTROLLER	FIC-950#18	High Alarm	<u>CIP_flow_rate_h</u>
CIP FLOW RATE CONTROLLER	FIC-950#18	Low Alarm	<u>CIP_flow_rate_l</u>
CIP LOOP PRESSURE	PI-950#37	High Alarm	<u>rise_pres_h</u>
CIP LOOP PRESSURE	PI-950#37	Low Alarm	<u>rise_pres_l</u>
CIP RETURN TEMPERATURE	TI-950#90B	High Alarm	<u>rise_temp_h</u>

Figure I (Continued)



Example #2

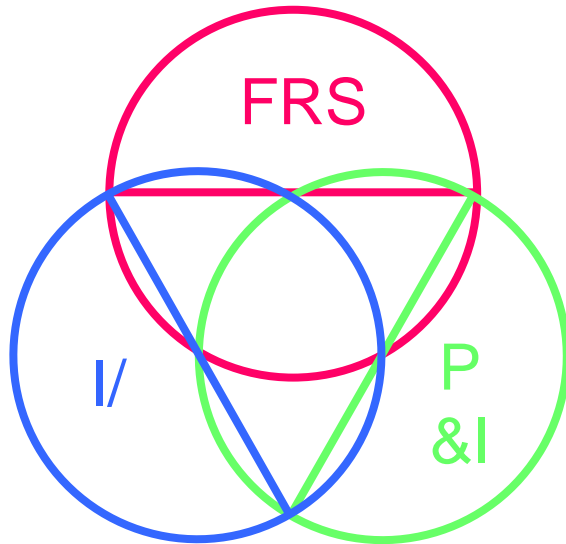
The SIP_FB_6 operation ~~script~~ will execute the following ~~sequence~~ phase.

1. ~~Generate Batch alarm "SIP_FB_6 Cool Started".~~ Notify the operator of the start of the phase.
2. ~~Set the cool_pause flag.~~ Pause the cooling process.
3. ~~Download a zero (0) to HoldAgitSpd.~~ Disable the Agitator Speed hold function (described in Section 3.2).

Figure II



Measure of Accuracy



Non-overlapping areas add
risk to project

Figure III

Written By: Denny Edwards, President
Property of G.F.S. Computing
Reprint by permission only.