INSPECTION OF SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENTS: A PILOT STUDY

Tereza G. Kirner
Janaina C. Abib

Federal University of São Carlos - PPG/CC
Rodovia Washington Luiz, Km 235 - 13.565-905, São Carlos, SP, Brazil.
<tereza, janaina>@dc.ufscar.br

Abstract
Software Requirements Specification is one of the first phases of system development. This phase results in the Software Requirements Specification (SRS) document, which must contain a complete, concise, high-quality description of the system being considered. The quality assurance of SRSs depends strongly on the use of appropriate techniques. This paper focuses on the use of inspection techniques in the quality evaluation of SRSs, presenting a pilot study on the subject.

Keywords
Software Requirements Specification document, Inspection techniques, Empirical study.

1. Introduction
System documentation includes all of the documents describing the system development, from the requirements specification to the final acceptance plan test. Among these, the Software Requirements Specification (SRS) is a key document, which must contain a complete, concise, high-quality description of the external behavior of the system being considered.

All the functional and nonfunctional requirements for the system have to be written in its SRS ([4], [7]).

To ensure that the contents of an SRS are adequate and compatible with the system's mission, it is necessary to evaluate the documentation generated, by means of appropriate techniques. In this sense, inspection techniques are very effective to guarantee the quality of these documentation ([7], [8]).

This article has several objectives, including: (1) to stress the importance of conducting inspections of SRSs, aiming to obtain high-quality system specification and documentation; (2) to describe how to use the Goal-Question-Metric (GQM) approach in the development and implementation of evaluation programs applied to SRSs; (3) to present a pilot study about inspection, which was performed with the purpose of detecting defects in SRSs.

Section 2 of this article presents an overview of the contents of a typical SRS. Section 3 summarizes the GQM approach. Section 4 describes a pilot study performed on measurement of SRSs, which was proposed and implemented with the support of GQM. Finally, section 5 gives the conclusions.

2. Software Requirements Specification Documents
Software requirements specification is one of the first phases of system development. It involves preparing a complete description of the system's external behavior, represented in terms of functional and nonfunctional requirements. It is a crucial phase of system development, because specification defects will become more and more difficult to repair, as the system is advancing to the next phases of its life cycle [4].

The concrete result of requirements specification is the SRS (Software Requirement Specification) document. There are several standards proposed for

Basic Overall Outline	
Section	**Contents of Section**
1. Introduction	Definition of: Purpose of SRS; Scope of product; Definitions, acronyms, and abbreviations; References; Overview of rest of SRS.
2. General Description	Definition of: Product perspective; Product functions; User characteristics; General constraints; Assumptions and dependencies.
3. Specific Requirements	Definition of all specific requirements for the application.
Specific Requirements	
Section	**Contents of Section**
3.1 Functional Requirements	Definition of each specific requirements in terms of: Introduction; Inputs; Processing; Outputs.
3.2 External Interface Requirements	Definition of: User interfaces; Hardware interfaces; Software interfaces; Communication interfaces.
3.3 Performance Requirements	Definition of all performance requirements.
3.4. Design Constraints	Definition of: Standard compliance; Hardware limitations.
3.5 Attributes	Definition of nonfunctional requirements, such as: availability; security; maintainability; safety; transferability/adaptation; etc.
3.6 Other Requirements	Definition of: Data base requirements; Operations; Site adaptation; etc.

Figure 1. Contents of an SRS

organizing the contents of an SRS. This work follows the standard approved by the Institute of Electrical and Electronics Engineers-IEEE [2]. This standard defines not only the basic overall outline that an SRS should contain but also the specific topics that should be tailored to the application at hand. Figure 1 presents the sections of an SRS.

Correct and complete SRSs are essential for the development of high quality systems. Additionally, better system documentation can be obtained by means of inspecting the considered document.

Inspection is a very promising technique to guarantee high quality at all phases of the system development in a cost-effective manner. It comprises software evaluation techniques, by which requirements, design or coding can be examined by a person or team, aiming to detect defects ([5], [9]). A defect is any condition or situation that could lead the system to behave in an undesirable manner with risks for the accomplishment of the system's mission ([7], [13]).

Although inspection techniques were originally proposed for design and code review [5], they were proved to be successful for the evaluation of many different software processes and products ([10], [11]). Studies on SRS inspection have reinforced the importance of detecting defects as early as possible in the software system, in order to avoid the propagation of such defects and, consequently, minimize the costs in future repair ([4], [7]).

3. The GQM Approach

The GQM is based on the paradigms of goal-oriented measurement and quality improvement, applied to software processes and products ([3], [6]). In this work, GQM served as a basis for the implementation of an evaluation program (a pilot study) on inspection of SRSs.

Figure 2 shows an overview of the GQM. Its stages and phases are summarized as follows.

The objective of the *Development of the GQM Plan* stage is to detail the intended measurement plan. This is achieved by the following phases: (1) *Prestudy*, that involves the characterization of the general context for the considered problem; (2) *Elaboration of the GQM Plan*, in which the goals, questions and metrics for the intended measurement study are defined; and (3) *Elaboration of the Measurement Plan*, involving the detailing of strategies and techniques for the measurement implementation.

The objective of the *Execution of the Measurement Plan* stage is to implement the intended

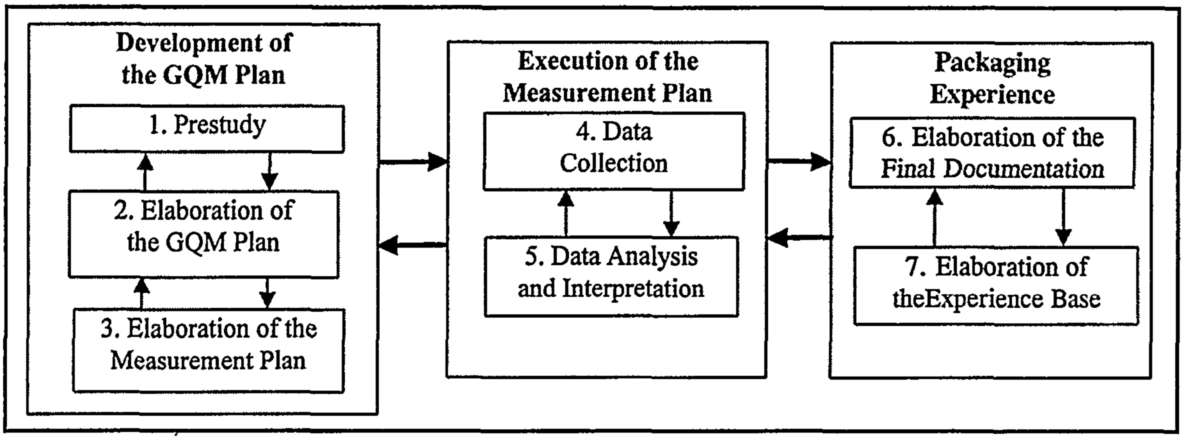

Figure 2. Stages and Phases of the GQM Approach

measurement plan. Its main phases include the mecessary *data collection* and the *analysis and interpretation* of these data. Some specific techniques, such as statistical techniques [12], are normally used here.

The objective of the *Packaging Experience* stage is to prepare the results obtained by the measurement, in a way that enables their efficient utilization.

The process involved in the GQM approach will be detailed and exemplified in the next section, through the description of a pilot study on SRS quality evaluation.

4. The Pilot Study

The application of GQM in the development of software quality measurement programs was tested through a pilot study on inspection of SRSs, described in the next subsections.

4. 1 Development of the GQM Plan

4. 1.1 Prestudy

This phase aims to collect the necessary information to perform the intended measurement program. For the pilot study, it included the analysis of possible alternatives, which resulted in the following decisions:

- *Performing the study in an academic environment.* This was based on the opportunity of the authors in working with these environment. Besides, some similar projects in academic environments have demonstrated useful research practices ([9], [10], [11]).

- *Choosing the participants.* The study involved a group of Computer Science graduate students from the Federal University of São Carlos. The group was considered adequate for the study, once the number of participants (24 students) was sufficient and all of them already had basic knowledge on the focused themes (software engineering concepts, and software requirements specification). The participants were divided in eight teams of three students each.

- *Deciding which inspections techniques to use.* The techniques adopted were *Ad Hoc, Checklist* and *Scenario,* indicated for detecting defects in SRSs ([9], [10]). *Ad Hoc* is characterized by a higher degree of informality and a consequent necessity of the participants' knowledge, during the inspection sections. *Checklist* is based on a series of questions that guide the participants in the detection of defects. *Scenario* is characterized by the use of different perspectives, previously defined, from which the participants will detect existing defects in the software document. These three techniques are presented in Appendix A.

- *Preparing the software requirements documents.* Three SRS for hypothetical systems, involving complexity levels compatible with the intended study, were used: a Video Rental System, a Parking Garage Control System, and an Automated Teller Machine Network System. The SRSs for these systems were prepared previously, according to the standards proposed by the IEEE [2]. In addition, a series of defects were injected in the documents, in order to enable the inspection.

Object *What will be analysed?*	Analyse	inspections techniques of SRS (Ad Hoc, Checklist, Scenario)
Purpose *Why will the object be analysed?*	for the purpose of	better understanding of the analyzed techniques
Quality Focus *What property of the object will be analysed?*	with respect to	efficiency of the techniques, according to the number and types of defects found
Viewpoint *Who will use the data collected?*	from the viewpoint of	researcher
Environment *In which environment does the analysis take place?*	in the following context	academic environment

Figure 3. Goal of the Pilot Study

4. 1. 2 Elaboration of the GQM Plan

a) Identification of the Measurement Goals

In this step, the goals of the measurement plan are identified. The different goals are formulated in terms of *object, purpose, quality focus, viewpoint* and *environment*. For the pilot study, the main goal was to evaluate the three inspection techniques, aiming to determine the efficiency of these techniques in the detection of defects in SRS.

Figure 3 summarizes the main goal defined for the pilot study.

b) Production of the GQM Plan

This step comprises the definition of a series of questions with respective metrics, which will guide the measurement plan. The questions must be formulated specifically for the considered problem, according to the respective goal. The metrics must be associated with the questions, trying to answer them in a quantitative way.

The GQM plan is produced through the following activities: preparation of abstraction sheets, definition of questions, and derivation of metrics.

b1) Abstraction Sheet

This is a technique for the acquisition and structuring of knowledge for the generation, refinement and review of the GQM plan [6]. It is an one-page document with a header and four quadrants. The goal is stated in the header and the following information is defined in the quadrants:

- *Quality focus*. It is defined at the upper left quadrant of the abstraction sheet and expresses the main content of the measurement goal.
- *Baseline hypothesis*. They are at the lower left quadrant in the abstraction sheet and represent the estimated results for each quality aspect related to the quality focus for the measurement program.
- *Variation factors*. They are defined at the upper right quadrant of the abstraction sheet and capture relationships between environmental factors and the quality focus, representing aspects that could interfere in the results of the measurement program.
- *Impact on baseline hypothesis*. They are described at the lower right quadrant in the abstraction sheet and state the (assumed) impact of each variation factor on the quality focus

Figure 4 defines the contents of a generic abstraction sheet. A simplified abstraction sheet taken from the main goal of the pilot study is shown the Figure 5.

b2) Questions

According to the GQM, the questions must be compatible with the objective previously defined for the measurement. It is suggested to formulate the questions concerning to the quality focus first, and then the questions concerning to the variation factors.

GOAL	Object	Purpose	Quality Focus	Viewpoint	Environment

Quality Focus *Describe the quality focus*	Variation Factor *Which factors have an impact on the quality focus?*
Baseline Hypothesis *What do you estimate is the current state according to the quality focus?*	Impact on Baseline Hypothesis *How do variation factors influence the quality focus?*

Figure 4. GQM Abstraction Sheet

Object	Purpose	Quality Focus	Viewpoint	Environment
inspection techniques for SRS quality evaluation	better understanding of the analyzed techniques	efficiency of techniques, according to the number and types of defects found	researcher	academic

Quality Focus • Analysis of the efficiency of the techniques, according to the number and the types of the defects found in the inspected SRSs.	**Variation Factors** • Type of inspection technique used. • Experience of participants. • Motivation to participate in the pilot study. • Performed training. • Time to perform the inspection.
Baseline Hypothesis • Total number of defects found, related to the inspection technique used: - *Ad Hoc* = 40% - *Checklist* = 50% - *Scenario* = 60% • Total number of defects found respected with the type of the defects: 1) Checklist - *Omission* = 40% - *Commission* = 60% 2) Ad Hoc - *Omission* = 40% - *Commission* = 60% 3) Scenario - *Data Type Consistency* = 50% - *Incorrect Functionality* = 60% - *Ambiguity and Missing Functionality* = 40%	**Impact on Baseline Hypothesis** • *Type of technique*: the percentage of defects found by the Ad Hoc technique is smaller than the percentage of defects found by Checklist, that is smaller than the percentage of defects found by Scenario. • *Type of defects*: the percentage of omission defects found is smaller than the percentage of commission defects found. • *Motivation to participate in the pilot study*: the higher the motivation is, the higher the amount of defects is found. • *Training to participate in the pilot study*: the better the training is, the higher the amount of defects is found. • *Time to perform the inspection*: the longer the inspection time is, the higher amount of defects is found.

Figure 5. Abstraction Sheet for the Pilot Study

Questions and Metrics Related to the Quality Focus
Q1 - What is the percentage of defects found, by team, in relation to the total amount of existing defects? M1.1 - Total amount of existing defects in the software requirements documents. M1.2 - Number of defects found by team. Q2 - What is the number of the defects found, according to the defect classification, in the total amount of existing defects? M2.1 - Total amount of existing defects in each defect class. M2.2 - Total amount of defects found in each defect class.
Questions and Metrics Related to the Variation Factors
Q3 - What is the inspection technique used by the participants? M3.1 - Inspection techniques considered. Q4 - What is the document inspected by the team? M4.1 - Software Requirements Specification documents.

Figure 6. Questions and Metrics for the Pilot Study

b3) Metrics

The metrics aim to quantify the formulated questions. Each metric must be related with, at least, one question, and each question must be, at least, associated with one metric.

Figure 6 contains a sample of questions and their respective metrics defined for the pilot study.

4. 1. 3 Elaboration of the Measurement Plan

In this phase, the GQM plan and the measurement goal are analysed to determine, for example, who will collect the measurement data, how and when these data will be collected, and how the activities will be conducted.

In the pilot study, all data were collected through the following forms:

- *Form 1.* It was utilized by the participants to annotate the defects found in the requirements document during the individual inspection sessions.
- *Form 2.* It was utilized to annotate the defects found in the requirements documents during the team inspection sessions.
- *Form 3.* It was utilized to collect complementary information such as motivation of the participants, difficulties in finding defects, etc.

Moreover, the three SRSs and the instructions for using each technique during the inspection sessions, were prepared.

The pilot study consisted of an initial training, followed by the measurement application.

- ***Training.*** Initially, the basic concepts, the IEEE SRS standards, and the three inspection techniques were presented. After these presentations, which lasted 30 minutes, the second part of the training was carried out. In this part, the participants inspected the SRS document of the Video Rental System. The document was read individually (in 30 minutes), and then, the participants were organized into teams that performed the inspection. The defects found were registered in the form 2. The Video System was only used for training.
- ***Measurement application.*** The measurement included two trials, performed on two different days, using the SRSs previously prepared. During the first trial, four of the eight teams inspected the SRS of a Parking Garage Control System and the other ones inspected the SRS of an Automated Teller Machine System. The inspection techniques used, by team, are indicated in Figure 7. The detection of defects, made individually, was limited to one hour and a half, and the defects found were annotated in form 1. After the individual inspection session, the team inspection section were carried out and the defects found by team were written in form 2. The second trial was similar to the first, except that the inspected SRSs and the technique allocated to the participants changed.

	Trial 1		Trial 2	
Techniques	Automated Teller Machine System	Parking Garage Control Systems	Automated Teller Machine System	Parking Garage Control System
Ad Hoc	Team 1	Team 5	Team 5	Team 1
Checklist	Team 2, Team 3	Team 6	Team 7	Team 2, Team 8
Scenario	Team 4	Team 7, Team 8	Team 3, Team 6	Team 4

Figure 7. Allocation of SRS Documents and Inspection Techniques to the Teams

After the two trials, the participants filled in the form 3.

4. 2 Execution of the Measurement Plan

4. 2. 1 Data Collection

In this phase, the data are collected and validated. For this pilot study, the data were collected in the forms and validated according to the defined goal.

4. 2. 2 Data Analysis and Interpretation

The objective of this phase is to analyse and interpret the collected data. The analysis is normally supported by Statistics techniques [12].

In the pilot study, the data were prepared and analyzed, and some graphics, tables and statistical measures were produced, for example:

- Amount of defects found in each SRS document, by participant and by team, using each technique.
- Amount of participants and teams that found each existing defect, in each SRS document.
- Average of defects found, by technique, in each SRS document.
- Amount of defects found, considering the different types of defects defined in the adopted taxonomy.

4. 3 Packaging Experience

4.3.1 Elaboration of the Final Documentation

The objective of this phase is to prepare the final document of the measurement program. This document has to include all the materials produced during the evaluation process.

The most important results of the pilot study were those that answer the question related to the stated goals, that is, *"How efficient are the inspection techniques Ad Hoc, Checklist and Scenario, according to the amount and the types of defects that were detected by participants and teams?"*

It is important to stress that, after their preparation and injection of defects, the SRSs were in the following situation: (1) the Automated Teller Machine System had 41 defects, including 12 omission defects and 29 commission defects; (2) the Parking Garage Control System had 31 defects, including 6 omission defects and 25 commission defects.

The main results of the pilot study are summarized as follows.

- Independently of inspection techniques adopted and types of defects found, the inspection results identified by individuals and by teams were:
  - In the Automated Teller Machine System, 26 (63,4%) of the 41 existing defects were detected by individual inspections, and 32 (78%) defects were detected by team inspections.
  - In the Parking Garage Control System, 22 (70,9%) of the 31 existing defects were detected through individual inspections, and 25 (80%) defects were detected by team inspections.

- Considering the team inspection and the followed techniques, independently of types of defects, the results were:
  - In the Automated Teller Machine System, 13 of the 41 existing defects (31,7%) were detected by the Ad Hoc technique, 17 defects (41,4%) were detected by the Checklist technique, and 26 defects (63,4%) were detected by the Scenario technique.

  - In the Parking Garage Control System, 18 of the 31 existing defects (58%) were detected by the Ad Hoc technique, 18 defects (58%) were detected by the Checklist technique, and 20 defects (64,5%) were detected by the Scenario technique.

- Considering the team inspection and the types of defects found, independently of the followed technique, the results were:
  - In the Automated Teller Machine System, 32 defects, including 5 (16%) omission defects and 27 (84%) commission defects, were detected.
  - In the Parking Garage Control System, 25 defects, including 2 (8%) omission defects and 23 (92%) commission defects, were detected.

The main results indicated that the team inspection were more efficient than individual inspections, since they detected a higher amount of defects in the two inspected documents. It is important to point out that the team inspection were carried out immediately after the individual inspections, in both trials.

Considering the inspection techniques, *Ad Hoc* was less efficient or similar than *Checklist*, which was less efficient than *Scenario*.

With respect to the types of defects, omission defects were less frequently found, in percentage, than commission defects.

The interpretation above is restricted to the simplified data analysis developed, which was based on number and types of defects existing and detected. In addition, they are limited to the scope of the pilot study.

A complete description of the pilot study and its results is given in [1].

4. 3. 2 Elaboration of the Experience Base

The objective of this phase is to package the experiences gained in the measurement program, to reuse the knowledge in future software projects and evaluation programs.

In this sense, the lessons learned by the pilot study will be useful for the next studies on system documentation. The results can be reused in new programs on quality evaluation of SRSs and in the proposal of strategies to improve the quality of these documents.

5. Conclusions

This article focused on the evaluation of Software Requirements Specification documents, performed by inspection techniques. Moreover, it proposes the use of the Goal-Question-Metric (GQM) approach, as a guide for the development and implementation of evaluation programs related to inspections of SRSs The ideas were presented and exemplified through a pilot study on the evaluation of inspection techniques applied to the detection of defects in SRSs.

The GQM made easy the preparation and execution of the measurement program, serving as an effective guide during the process.

The main results obtained in the pilot study indicated the difficulties of detecting defects in SRSs. These results are compatible with other results, obtained in similar studies ([10], [11]). However, new experiments are necessary to validate and extend these results.

Finally, it is important to point out the need of empirical studies for the measurement and improvement of SRSs and other system documentation.

References

1. Abib, J.C. **Evaluation of Inspection Techniques Applied to Software Requirements Specification Using the GQM Approach**. São Carlos, UFSCar, 1997. Master Thesis, in preparation (in Portuguese).

2. ANSI/IEEE Std 830-1994. **IEEE Guide to Software Requirements Specifications**. IEEE Computer Society Press, 1994.

3. Basili, V.R., Caldiera, G., and Rombach, H.D. *Goal Question Metric Paradigm*. **Encyclopedia of Software Engineering**, John Wiley & Sons, New York, Volume 1, 1995, pp. 527-532.

4. Davis, A.M. **Software Requirements** Prentice Hall, Englewood Cliffs, New Jersey, 1993.

5. Fagan, M.E. *Design and Code Inspection to Reduce Errors in Program Development* **IBM System Journal**, Volume 15, Number. 3, 1976, pp. 219-248.

6. Gresse, C., Rombach, H.D., and Ruhe, G. A **Practical Approach for Building GQM-Based Measurement Programs: Lessons Learned from Three Industrial Case Studies**. Tutorial Report, São Carlos, SP, Brazil, 1996, 50 p.

7. Kirner, T.G., and Davis, A.M. *Nonfunctional Requirements of Real-Time Systems.* **Advances in Computers**, Volume 42, Academic Press, New York, 1996, pp. 1-39.

8. Kirner, T.G., and Rombach, H.D. InTec-SRE: *Innovative Technologies for Software Requirements Engineering.* **Proc. of the 3rd. German-Brazilian Workshop on Information Technology** (St. Jahnichen, and Lucena, Eds.), Berlim, Germany, 1995, pp. 56-61.

9. Laiterberger, O. **Perspective-based Reading: Technique, Validation and Research in Future.** ISERN Technical Report ISERN-95-01, University of Kaiserslautern, Germany, 1995.

10. Porter, A.A., Votta, L.G., and Basili, V.R. *Comparing Detection Methods for Software Requirements Inspection: A Replicated Experiment.* **IEEE Transactions on Software Engineering,** Volume 21, Number 6, 1995, pp. 563-575.

11. Schneider, G.M., Martin, J., and Tsai, W.T. *An Experimental Study of Fault Detection in User Requirements Documents.* **ACM Transactions on Software Engineering and Methodology,** Volume 1, Number. 2, 1992, pp. 188-204.

12. Sininchi T. **Business Statistics by Examples.** Dellen Publ., San Francisco, CA, 1992.

13. Thayer, R., and Dorfman, M. **System and Software Requirements.** IEEE Computer Society, Los Alamitos, CA, 1990.

Acknowledgments

We would like to thank *Christiane Gresse* (IESE FhG, Germany), for her valuable help in the definition of the pilot study, mainly on the use of GQM. We would also like to stress the support of CNPq and GMD in the context of the *InTec-SRE Project*, sponsored by the *Brazil/Germany Cooperation Program on Information Technology.*

Appendix A

Software Inspection Techniques

This appendix summarizes the three inspection techniques for detecting defects, used in the present work. The techniques were adapted from [9] and [10], and the taxonomy of defects included in the techniques was obtained in [11].

A1. *Ad Hoc* Technique

This technique is characterized by its informality, since it does not define a specific process to perform the inspection. For this study, the technique was summarized to guide the participants during the inspection sessions, in the identification of the defects in the requirements documents, according to the following taxonomy:

- **Omission Defects**

– *Missing Functionality.* Information describing the desired internal operation behavior of the system has been omitted from the requirements document.

– *Missing Environment.* Information describing the required hardware, software, database, or personnel environment in which the system will run has been omitted from the requirements document.

– *Missing Performance.* Information describing the desired performance specification has either been omitted or described in an unacceptable way.

– *Missing Interface.* Information describing how the proposed system will interface and communicate with objects outside the scope of the system has been omitted from the requirements document.

- **Commission Defects**

– *Ambiguous Information.* An important term, phrase or sentence essential to the understanding of system behavior, has either been left undefined or defined in a way that can cause confusion and misunderstanding.

– *Inconsistent Information.* Two sentences, contained in the requirements document, directly contradict each other or express actions that cannot both be correct or cannot both be carried out.

- *Incorrect Fact.* Some sentence contained in the SRS asserts a fact that cannot be true under conditions specified in the requirements document.

- *Wrong Section.* Essential information is misplaced within the requirements document.

A2. *Checklist* Technique

This technique guide the participants of the inspection to identify defects through some questions. They are elaborated according to the used defects taxonomy. Examples of questions for the present study are:

- **Omission Defects**
- *Missing Functionality*
* Are the described functions sufficient to meet the system objectives?
* Are all inputs to a function sufficient to perform the required function?
* Are undesired events considered and their required response specified?
* Are the initial and special states considered (e.g., system initiation, abnormal termination)?

- *Missing Environment*
* Has the functionality of hardware or software interacting with the system been properly specified?

- *Missing Performance*
* Can the system be tested, demonstrated, analyzed or inspected to show that it satisfies the requirements?
* Have the data type, rate, units, accuracy, resolution, limits, range and critical values for all internal data items been specified?
* Have the accuracy, prediction, range, type, rate, units, frequency, and volume of inputs and outputs been specified for each function?

- *Missing Interface*
* Are the inputs and outputs for all interfaces sufficient?
* Are the interface requirements between hardware, software, personnel, and procedures included?

- **Commission Defects**
- *Ambiguous Information*
* Are the individual requirements stated so that they are discrete, unambiguous and testable?
* Are all mode transitions specified deterministically?

- *Inconsistent Information*
* Are the requirements mutually consistent?
* Are the functional requirements consistent with the overview?
* Are the functional requirements consistent with the actual operating environment?

- *Incorrect Fact*
* Are all the described functions necessary to meet the system's objectives?
* Are all the inputs to a function necessary to perform the required function?
* Are all the inputs and outputs for all interfaces necessary?
* Are all the outputs produced by a function used by another function or transferred across an external interface?

- *Wrong Section*
* Are all the requirements, interfaces, constraints, etc., listed in the appropriate sections?

A3. Scenario Technique

In this technique, the inspection must be done based on three different scenarios defined according to the used defects taxonomy. For this study, the scenarios used were:

- **Data Type Consistency Scenario**

1. Identify all data objects mentioned in the document overview (e.g., hardware component, application variable, abbreviated term or function):
a) Are all data objects mentioned in the overview listed in the external interface section?

2. For each data object, appearing in the external interface section, determine the following information:
* Object name.
* Class: (e.g., input port, output port, application variable, abbreviated term, function).
* Data type: (e.g., integer, time, Boolean, enumeration).
* Acceptable values: Are there any constraints, ranges, limits, etc., for the values of this object?
* Failure values: Does the object have a special failure value?
* Units or rate.

* Initial values.
a) Is the object's specification consistent with its description in the overview?
b) If the object represents a physical quantity, are its units properly specified?
c) If the object's value is computed, can that computation generate a non-acceptable value?

3. For each functional requirement, identify all data object references:
a) Do all data object references obey the formatting conventions?
b) Are all data objects included in the requirement listed in the input or output sections?
c) Can any data object use be inconsistent with the data object's types, acceptable values, failure value, etc.?
d) Can any data object definition be inconsistent with the data object types, acceptable values, failure value, etc.?

- **Incorrect Functionality Scenario**

1. For each functional requirement, identify all input and output data objects:
a) Are all values written to each output data object consistent with its intended function?
b) Identify at least one function that uses each output data object.

2. For each functional requirement, identify all specified system events:
a) Is the specification of the events consistent?
b) If the object represents a physical quantity, are its *units properly specified?*
c) If the object's value is computed, can that computation generate a non-acceptable value?

3. Develop an invariant for each system mode (i.e., under what conditions must the system exit or remain in a given mode?):
a) Can the system's initial conditions fail to satisfy the initial mode's invariant?
b) Identify a sequence of events that allows the system to enter a mode without satisfying the mode's invariant?
c) Identify a sequence of events that allows the system to enter a mode, but never leave it?

- **Ambiguities and Missing Functionality Scenario**

1. Identify the required precisions, response times, etc., for each functional requirement:
a) Are all required precisions indicated?

2. For each requirement, identify all monitored events:
a) Does a sequence of events exist for which multiple output values can be computed?
b) Does a sequence of events exist for which no output values will be computed?

3. For each system mode, identify all monitored events:
a) Does a sequence of events exist for which transitions into two or more system's modes is allowed?