

Improving Software Quality By Inspecting Functional Specifications

by

John Michael Roberts

B.Sc., University of Victoria, 1982


A thesis submitted in Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE
in the Department of Computer Science

We accept this thesis as conforming to the required standard


Dr. D. M. Hoffman, Supervisor (Department of Computer Science)


Dr. M. H. van Emden, Departmental Member (Department of Computer Science)


Dr. D. McCutcheon, Outside Member (School of Business)


Dr. P. Walsh, External Examiner (Department of Computer Science)

© John Michael Roberts, 1998

University of Victoria

All rights reserved. Thesis may not be reproduced in whole or in part, by photocopy or
other means, without the permission of the author.

QA 76.76
D47 R63

Supervisor: Dr. D. M. Hoffman

Abstract

The quality of a software system, as perceived by its users, is determined as much by the effectiveness of its functionality in supporting their business activities, as by its freedom from technical defects. Therefore, efforts to improve software quality should consider the quality of the Functional Specification, as it represents user expectations of the eventual functionality.

The technique of formal inspection has been used effectively for code, and has been extended to the technical specification of the code. I show how to apply inspection to the Functional Specification as well, resulting in higher quality in the finished product when evaluated by users. The process of inspection, in addition to detecting and correcting defects in the Functional Specification, increases user understanding of the system that will be built and the manner in which it will be used. Increased understanding may result in more accurate expectations and fewer failure reports that are the result of user error rather than system defects.

Although many researchers have reported the use of inspections and design reviews for Functional Specifications, none has documented an inspection process designed for the unique objectives and characteristics of Functional Specifications. My research, which addresses this omission, includes the experimental application of a new inspection approach.

For my experiment, I designed an inspection process based upon previously documented techniques and results, but modified to take into account experience from the design, development and acceptance testing of a system built for a commercial client by PricewaterhouseCoopers. I reviewed this design with the participants from the project,

revised it, and then applied it to another PricewaterhouseCoopers project, of similar size and scope. The experiment included training of the inspection team, management of the inspection processes, and evaluation of the results, for a Functional Specification of over 150 functions.

The objectives of the inspection design were achieved, in improved quality of the Functional Specifications and increased understanding of it among users and developers. The results were achieved both through the inspection activities and increased care in producing the specification. The experiment also produced useful information about implementation issues that will assist in future applications of the technique.

The inspection process will be applied to future projects of PricewaterhouseCoopers, as it promises to reduce acceptance test failures, with a corresponding increase in real and perceived quality of the delivered systems.

Examiners:



Dr. D. M. Hoffman, Supervisor (Department of Computer Science)



Dr. M. H. van Emden, Departmental Member (Department of Computer Science)



Dr. D. McCutcheon, Outside Member (School of Business)



Dr. P. Walsh, External Examiner (Department of Computer Science)

Contents

Abstract	ii
Contents	iv
List of Tables.....	vi
List of Figures	vii
Acknowledgements	viii
Dedication	ix
1 Introduction	1
1.1 Thesis Organization	3
2 Related Work.....	5
2.1 Terminology.....	5
2.2 Quality Definitions and Problems.....	6
2.3 Inspection Concepts and Design.....	7
2.3.1 Approach and Objectives	7
2.3.2 Timing.....	7
2.3.3 Inspection Criteria.....	8
2.3.4 Inspection Teams	8
2.3.5 Training.....	8
2.3.6 Project Characteristics.....	9
2.3.7 Mechanics and Tools.....	9
2.4 Inspection Application and Results	9
2.5 Inspecting the Functional Specification.....	10
3 Problems of Software Quality	12
3.1 Users' Perceptions of Software Quality	12
3.2 Defect Types	13
3.3 Defect Class Relationships to the Functional Specification	16
3.4 Defect Frequency	19
3.5 Sources of Defects in Functional Specifications	20
4 Quality in Functional Specifications	22

4.1	Inspection.....	22
4.2	Inspection Design Considerations	24
4.3	Inspection Design	31
4.3.1	Inspection Process.....	32
4.3.2	Inspection Elements	35
4.3.3	Inspection Evaluation.....	36
5	Applying the Inspection Approach.....	37
5.1	Research Plan.....	37
6	Results	42
6.1	Comments on Draft Inspection Design.....	42
6.2	Inspection Process.....	43
6.3	Inspection Results.....	46
6.3.1	Inspection Reports.....	47
6.3.2	Inspection Meetings	48
6.4	Inspection Evaluations.....	50
6.5	Results Summary	54
7	Conclusions	59
7.1	Value of Inspections for Functional Specifications	59
7.2	Practical Issues of Implementation	61
7.3	Areas for Further Research	62
	Bibliography.....	64
	Appendix A: Inspection Report	68
	Appendix B: Inspection Evaluation	72
	Appendix C: Summit D [®] System Delivery Specification	75
	Appendix D: Inspection Plan	83
	Appendix E: Sample Inspection Reports	84
	Appendix F: Sample Inspection Meeting Report.....	89
	Appendix G: Sample Inspection Evaluations	91
	Appendix H – Project Profiles	98

List of Tables

Table 1: Software Development Life Cycle Definitions.....	5
Table 2: Defect Taxonomy.....	15
Table 3: Defect Class Definitions	17
Table 4: Defect Class Relationship to Functional Specification.....	18
Table 5: Defect Class Frequency	19
Table 6: Inspection Process Details	34
Table 7: Inspection Process Implementation Considerations	41

List of Figures

Figure 1: Inspection Process 32

Acknowledgements

The following people and organizations have contributed to the creation of this thesis:

- Mr. Allan Hart and Mr. David Williams, partners in the firm of PricewaterhouseCoopers, whose commitment to the development of their staff is evident in their professional and financial support for my studies
- The management consultants of PricewaterhouseCoopers, who provided their knowledge and time to test and improve my ideas
- The British Columbia Buildings Corporation, who permitted me to use testing data from the Information Systems Alignment Project, and the Ministry of Finance and Corporate Relations of British Columbia, who enthusiastically participated in the application of my inspection design
- Dr. Dan Hoffman, my graduate supervisor, whose guidance always balanced the requirements of academia and the realities of business
- And most importantly, my family, whose endless patience and confidence gave me the strength to finish.

Dedication

To my wife, Tami Lynn Roberts, whose loving support has been essential to my success in completing this thesis, and to everything else I have attempted since I met her.

And to my son, Jakob Maxwell Roberts - may your learning never end.

1 Introduction

Quality is a characteristic desired in software systems. It is both objective and measurable, in the absence of failures caused by code defects, and subjective, in the perception of end users of the degree to which the system meets their needs and expectations.

Many software development organizations work to improve the quality of their products through improvements to the development process. These include activities that manage end user expectations, such as joint application design and prototypes, and internal activities such as testing and code inspection.

In the process there is a critical work product. The Functional Specification, which describes the externally observable behaviour of the system, represents both the users' expectations of functionality and the specification that designers and developers will implement. In the Information Systems Alignment Project (ISAP), a large systems development project that I managed, a significant proportion of the failures identified by users during acceptance testing could be attributed to the Functional Specification - either to defects in the document, or inadequate understanding of it by the users. The defect frequency and characteristic data is representative of commercial development projects, as ISAP encompassed both broad scope (all corporate systems for a large organization) and scale (approximately 12,000 person-days effort). A profile of ISAP is included in Appendix H.

Based on my analysis of these defect characteristics, I believed that the actual and perceived quality of a software system could be increased by improving the process of creating and understanding the Functional Specifications. Yet many projects do not include a rigorous approach to quality control for the Functional Specifications, and other researchers have not identified an approach that recognizes both the need to detect

and correct defects and the need to achieve understanding in users. The published literature describes techniques and experiments that inspect only code or technical specifications, and focus solely on defect detection and correction.

To address the broader objectives of inspecting Functional Specifications, I defined a modified inspection process. I used the principles employed by other researchers for inspection of code and technical specifications, and adjusted them to take into account the special characteristics of a Functional Specification, such as its placement in the development process and its multiple audiences. The modifications included extracting inspection criteria from the standards for Functional Specifications in *Summit D*, the system development methodology created and used by PricewaterhouseCoopers, the professional services firm of which I am a member.

I reviewed my design with a number of colleagues - analysts, designers and end users - who were members of my project teams. They contributed comments and suggestions that I used to improve the design. I then applied the inspection approach to another project of similar size and type, the Family Bonus Renewal (FBR) project. It too involved a broad range of functionality and significant size (approximately 4,000 person-days of effort); its profile is included in Appendix H. FBR was undertaken by PricewaterhouseCoopers on behalf of the Ministry of Finance and Corporate Relations of British Columbia. In approving the use of my inspection process for FBR, both organizations expected to achieve immediate and long-term benefits in the form of improved quality and reduced cost and risk. My evaluation of the benefits was subjective, relying on a survey of the project participants to determine the value of the inspections, because no control was available to allow quantitative analysis. This is consistent with the approach used by PricewaterhouseCoopers for evaluating new techniques and tools.

The experiment demonstrated that the inspection process achieved all of its objectives: according to the participants and my own observations, the quality of the Functional

Specification was higher than in other projects, and the level of understanding of its content and the system that would be produced was greater. During the experiment, a number of adjustments were made to the inspection process, which generated data regarding the scheduling of inspection activities, grouping of inspection subjects, and the requirements for training of the inspection teams. This data is summarized in my conclusions to point out important implementation considerations.

The objectives of the inspection process were achieved as much by the changes in project team behaviour as by the inspection activities themselves. Introducing formal peer reviews resulted in more thorough user walkthroughs prior to the inspection, and a consistent level of quality across all sections of the specification.

I conclude that the benefits of the inspection process make it a valuable technique for use in future PricewaterhouseCoopers projects. The effort expended in performing inspections resulted in a high quality Functional Specification, and that effort is offset by reductions in the communication between analysts, users and developers that would normally occur later in the project. Inspections produced an immediate, visible benefit – the rapid sign-off of the FBR Functional Specification by the client – and are expected to reduce acceptance test failures.

The technique will be adopted by PricewaterhouseCoopers for its system development projects and will become part of its training curriculum for project managers.

1.1 Thesis Organization

In Chapter 2, I review the work published by other researchers about software quality as it relates to inspection and functional specifications.

The problem I address is the subject of Chapter 3. It includes an analysis of the failures identified by users during Acceptance Testing of a large software system, and the relationship of the failure types to the quality of the Functional Specification.

Chapter 4 provides the design of an inspection process for Functional Specifications, based on the unique characteristics of that work product and its placement in the software development life cycle.

The inspection elements, the experiment that applied them, and the evaluation process are described in Chapter 5.

Chapter 6 reports the results of the experiment, and the conclusions I draw regarding the value of the inspection process are shown in Chapter 7.

2 Related Work

2.1 Terminology

Many models of the software development life cycle exist. For the purposes of this document, the following simplified definitions apply:

Step	Deliverable	Characteristics
Requirements Analysis	Requirements Definition	High-level description of the business functions and information required in the automated system, equated to business processes that the system will support.
Initial Design	Functional Specification	Description of the modules and data that the system will include, to define the entire externally-observable behaviour of the system, but none of its internal implementation characteristics. The Functional Specification translates the business functions in the Requirements Definition into a set of system modules.
Detailed Design	Technical Specification	Fully detailed specification of the system, including all internal implementation decisions, to define all information needed to proceed with program development.
Coding	Completed System	Finished program modules, data structures, test data and operating environment, and testing for compliance with the Technical Specification.
Acceptance Testing	Accepted System	Testing of the Completed System for compliance with the Functional Specification and Requirements Definition.

Table 1: Software Development Life Cycle Definitions

The terminology of the inspection design is derived from the standard methodology used by PricewaterhouseCoopers for its system delivery projects, entitled *Summit D* [PwC 97]. This was the methodology used for both ISAP, the project analyzed for defect frequency, and FBR, the project that was used for experimental application of the inspection design. It provides a common basis for understanding among project participants, and the analysts and designers that applied the inspection process were familiar with *Summit D*. *Summit D* uses the title "System Delivery Specification", or SDS, to denote the functional specification. An excerpt from the methodology, showing the description of the SDS, is included in Appendix C.

2.2 Quality Definitions and Problems

The IEEE defines software engineering quality as: "the totality of features and characteristics of a product or service that bears on its ability to satisfy given needs", and is summarized by [Fox 97] as both conformance to a standard or specification and fitness for use. There is a great deal of research that analyzes the failures of software to achieve quality, and the manner in which defects are created during the software development process

For example, observations by [Mashiko 97] have determined that there is a high rate of defect injection in the requirements and specification steps, and that rework costs are significantly higher for requirements-injected defects. [Blackburn 96] points out that studies of development efforts throughout the world have shown that the fastest developers spend more time in requirements and design than average, and are still twice as productive overall.

But defects that cause software failures are not the only measure of quality. The phrase "fitness for use" recognizes the quality of a system as a measure of the perceptions and experiences of its users. [Kitchenham 96] identifies work on the underlying problem - how do we define fitness?

Quality as a measure of customer perception is embodied in frameworks such as the ISO 9000 series, and quality management applies at all steps of a product creation process - input, output, and intermediate work products [Daily 92].

2.3 Inspection Concepts and Design

2.3.1 Approach and Objectives

There are two groups of approaches to inspection [Porter 97b]: the Fagan approach [Fagan 76], which involves an initial overview and an inspection based on paraphrasing, and the Active Design Review (ADR) approach of Parnas and Weiss [Parnas 87] in which the reviewers answer questions about the work product. In the former, preparation by the participants is for the purpose of increasing their understanding, not to identify defects. In ADR, the preparation is to find defects, and it has been found that a meeting is of more value in raising understanding than in detecting further defects.

A comprehensive analysis of practices in software inspections is shown in [Gilb 93], [Schulmeyer 87], and [von Mayrhauser 90]

2.3.2 Timing

Many researchers (e.g. [Ackerman 84]) have identified the timing of the inspections, both in relation to the project plan and of the inspection steps themselves, as critical to the success of the process. [Porter 97a] points out that the inspection process can have a negative effect on the schedule, with corresponding effects on carrying and opportunity costs.

In [Glass 97], there is a compelling argument for speed in the system development process and against excessive demand for perfection.

[Yourdon 86] argues that the inspection subject must be sufficiently small to allow the inspection meeting to be completed in ninety minutes, and [Raz 97] suggests that inspections are most valuable when they are frequent (based on smaller work products) and well prepared-for, rather than carefully tuned for efficiency.

2.3.3 Inspection Criteria

Each of the comprehensive references provides detailed inspection criteria, and [Fagan 86] links this to the need for unambiguous *exit criteria* for each step of the development process.

[Siddiqi 96] describes the essential difficulty in establishing requirements and the corresponding uncertainty in attempts to specify systems to meet them.

2.3.4 Inspection Teams

[Jeffrey 96] suggests that a project maintain an external perspective in the creation of the system, and many of the inspection designs include end users as observers or participants. The expectation of the researchers is that the variety of viewpoints may help smooth inconsistencies between teams in the thoroughness of the inspection and their definition of correctness or completeness, which is noted by [Shirey 92] as a contribution to the failure of inspections.

2.3.5 Training

All inspection processes include a provision for training of the team, but few details of the training approach are given. [Ackerman 82] suggests that training for inspections is best performed on a project-specific basis.

2.3.6 Project Characteristics

[Jones 85] suggests that other characteristics of the project be taken into account in preparation for the inspections, including methodologies, status tracking techniques and design languages.

2.3.7 Mechanics and Tools

Few references suggest specific tools for inspection. [Verner 97] discusses the value of prototyping, while [Basili 97] discusses the technique of reading, concluding that it is most productive when employed in a structured manner and the subject matter is important to the reader.

[Chernak 96] discusses improvements to the inspection process, and suggests that the inspection checklist should be closely tied to the structure of the Functional Specification itself.

2.4 Inspection Application and Results

There is a large body of work that presents case studies of inspections and their results, notably [Blakely 91], [Doolan 92], [Fagan 86], [Gilb 93], [Porter 97a], [Porter 97b], and [Weller 93]. [Kelly 92] and [Porter 97b], contain quantitative analyses.

In every case, the researchers concluded that inspections produced a favourable or neutral result, depending on the inspection design and the manner of its application. Code inspection has been most thoroughly analyzed, although some other work products have been considered, such as module specifications, as in [Hoffman 89] and [Jackson 94].

However, [Yourdon 89] states an opinion that inspections are not much more valuable than testing (which applies to code inspections only), if they are poorly managed or

expected to create quality themselves. [Weller 93] points out that inspections cannot overcome serious problems with the development process, but only help to identify them.

There have been conflicting research results regarding incremental effectiveness (additional defects detected) of an inspection meeting for Fagan-style inspections. In [Porter 97a] a 30% "gain rate" was reported at inspection meetings, but other work [Votta 93] has shown rates as low as 5% for design review meetings.

2.5 Inspecting the Functional Specification

While the case studies often refer to the results of design inspections, none of the published work provides an inspection process for Functional Specifications. For example, [Daily 92] talks about design reviews as part of a quality management program, but does not define a rigorous process. In [Schulmeyer 87], Chapter 9 (by J. Dobbins), there is a discussion of inspection types, including high-level design inspection, but still limits the process to confirming that the functional design is correct with respect to the requirements definition. [Kohli 75] defines inspections for high-level designs in a similar manner.

But even using an untailed process, the results have been encouraging. [Doolan 92] found a thirty-fold payback of effort resulting from inspections of System Requirements Specifications. A study of inspections of various work products, conducted at the Jet Propulsion Laboratory and described in [Kelly 92], concluded that "significantly more defects were found per page at the earlier phases of the software life cycle". Other beneficial side-effects noted by [Doolan 92] include improvements in the overall quality practices skills of the system development team, and [Blakely 91] noted that a lack of design reviews made code inspections more difficult for software engineers who had not been involved in the creation of a work product.

The elements of a Functional Specification - its standard structure, content, and quality measurement - can be found in [CLC97], [IEEE 93] and [IEEE 95]. In addition, [Meyer 85] characterizes common defects in specifications, and [Thayer 97] consolidates a variety of work related to the process and quality of software requirements specification.

In addition to the eventual end users of the system, other important customers of the Functional Specification are discussed in [Xenos 97], such as the users who will write documentation and provide training, and the designers who will turn the specification into an automated system.

3 Problems of Software Quality

3.1 Users' Perceptions of Software Quality

One of the primary measures of the quality of a software system is the number and severity of defects detected during acceptance testing. Since acceptance testing is performed by users, their perception of quality is shaped by their first impressions, and even if the system is ultimately accepted by the users' organization, there are often long-standing and widespread attitudes toward the system and its developers arising from the acceptance testing. This is particularly true if, as in many projects, the system is put into production use before all of the defects have been found and fixed. Those early users also shape the opinions of their colleagues and management, as they are frequently the most proficient users and are consulted for assistance and opinions.

The implementation of a new system causes change to the procedures and working environment of its users; they seldom welcome the change. Failure of the system to meet their business requirements, or lack of knowledge about its use, exacerbates the user impact. Users' unwillingness to learn and adopt a new system, driven by a perception that the system is flawed or difficult to use, increases the time during which productivity is negatively affected. This results in a slower realization of benefits than could be achieved if the users greeted the system with enthusiasm. Improving the actual and perceived quality of a system, with a corresponding decrease in reported acceptance test failures, can be expected to reduce the costs of the development project and achieve its anticipated benefits more quickly and fully.

Much research and industrial activity has been applied to improving software quality. Many approaches have been considered or implemented, to all steps in the software development life cycle. The analysis below shows that an intermediate work product in

the life cycle - the Functional Specification - is of significance in the actual and perceived quality of the completed system.

3.2 Defect Types

It is common for projects to split Acceptance Testing into two parts: a test performed by the development team for compliance with the Functional Specification (often called System Testing), followed by a distinct test by end users (called Acceptance Testing) again with regard to the Functional Specification. Testing against the Requirements Definition is often implicit, in that the acceptance criteria are expected by the project team to be based only on the Functional Specification.

This section includes an analysis of defect types, and a discussion of the frequency and effect of each type.

In the course of Acceptance Testing, many different defects are identified. A possible taxonomy, based on actual examples encountered during ISAP, is shown in the following table. The Defect Class is discussed below.

Defect Source	Typical User Comment	Class
Error in Production of Functional Specification Requirements Mismatch to User Needs Incorrect Requirements Definition Incomplete Requirements Definition Misunderstanding of Requirements Definition Specification Mismatch to Requirements Incorrect Specifications Incomplete Specifications User Misunderstanding of Specification Unclear Requirements Changing Requirements Inconsistent Requirements Hostile User	"That's not what I need it to do" "It doesn't do what I need it to do" "That's not what I told you I needed" "That's not what I expected" "It doesn't do what I expected" "That's not how I expected it to work" "Now I can see that I want ... instead" "I wanted that before, but now ..." "I want it different from the others" "I just don't like it"	CR CR USER / CR SPEC SPEC USER / SPEC CR / USER CR CR USER
Errors in Implementation of Specification Code Mismatch to Specification Incorrect Function Incorrect Logic Incomplete Implementation Inadequate Performance Incorrect Presentation Technology Failure to Provide Functionality Incorrect Capability Definition Incorrect Preparation / Configuration Inadequate Capacity Data Mismatch to Specification / Assumptions Data Values Outside Specified Domain Data Values Missing When Expected Data Value Combination Invalid	"It doesn't work" "It doesn't do what it's supposed to" "It's not fast enough" "It doesn't look right" "It doesn't do what they said it would" "It doesn't work right" "There's not enough ..." "I can't enter ..." "I have to enter ..." "It doesn't accept ..."	SOFT SOFT ENVI / SOFT SOFT / ENVI ENVI ENVI ENVI DATA / SOFT DATA / SOFT DATA / SOFT
Errors in Validation of Implementation Test Defect Test Script Does Not Match Specification Test Script Shows Wrong Expected Result Test Script Demands Incorrect Operation Test Script Incomplete vs. Specification	"It looks right, but it's supposed to ..." "I can't run the test" "I never tested ..."	TEST TEST TEST

Acceptance Test Incomplete vs. Requirement	"I don't know if it will do ... for me"	TEST/USER
Testing Error		
Tester Deviates from Test Plan		
Tester Makes Error in Test Plan Execution	no comment made until failure	
Tester Willfully Alters Test Plan	"I wanted to try ..."	TEST
Tester Fails to Recognize Failure		
Tester Fails to Notice Failure	no comment made or failure logged	
Tester Finds Failure to be Valid Behaviour	no comment made or failure logged	
Tester Fails to Recognize Valid Behaviour	any of above comments	TEST / USER
Other Classifications		
Repeated Failure		DUPL
Additional Failure Arising from Single Defect		any
Additional Defect Arising from Single Error		any
User Preference	"I'd like it to look like ... instead"	PREF
User Unable to Perform Test	"I don't know how to ...:"	USER

Table 2: Defect Taxonomy

During ISAP Acceptance Testing, all defect reports were logged in an online Incident Report form that allowed complete analysis and classification. Testers were not asked to supply a Defect Class - these were added by user management or technical personnel later. In some instances (roughly 10%), the Defect Class was difficult to determine either because of the complexity of the defect, or because it exhibited aspects of multiple classes. All Defect Class assignments were confirmed by the author, who acted as Project Manager for ISAP.

3.3 Defect Class Relationships to the Functional Specification

The definitions used for assignment of the Defect Classes were a summary of the defect types shown above, and are listed in the following table.

Code	Defect Class Name	Definition
CR	Change Request	Additional functionality requested by the tester, that was not included in the requirements specification or the functional specification.
PREF	Preference	Alternate functionality or presentation requested by the tester, different from the form included in the functional specification.
SPEC	Specification Defect	A defect introduced into the functional specification through an error in transforming the requirements definition, and propagated into the completed system. This could take the form of: <ul style="list-style-type: none"> • incomplete support for a requirement • inconsistency in the functional specification • ambiguity in the functional specification.
USER	User Error	An error in the operation of the system made by the tester (and acknowledged not to be a failure), due to imperfect understanding of the functional specification and the operating procedures derived therefrom.
SOFT	Software Defect	A defect introduced into the completed system through an error in transforming the functional specification.
ENVI	Environmental Defect	A failure in the operation of a technology component supporting the system, usually caused by a defect in

		its configuration.
DATA	Data Defect	A defect in the data supplied for testing, introduced during data conversion or direct creation, in that the data does not adhere to domain restrictions or integrity rules expressed in the functional specification.
TEST	Test Error	An error made by the tester and incorrectly recorded as a system failure.
DUPL	Duplicate	Multiple reports of the same failure or different failures arising from the same defect.

Table 3: Defect Class Definitions

Each of these Defect Classes has a distinct relationship to the Functional Specification, as described below.

Defect Class	Relationship to Functional Specification
CR	Change requests are by definition outside the scope of the Functional Specification, but experience suggests that requirements that were overlooked by users may be more often identified if the Functional Specification is easy to understand, and if the users' review process is effective.
PREF	The Functional Specification usually defines user interfaces, but if users take part in reviews of the document or supporting materials (diagrams, prototypes etc.) they may be more likely to identify different preferences.
SPEC	All SPEC failures are defined to be a result of defects in the Functional Specification.

USER	User manuals and training are usually derived from information in the Functional Specification document. User errors may therefore be reduced through improvements in the source document or the process by which the information in the document is imparted to users.
SOFT	Weak or incomplete understanding of the Functional Specification may lead to coding errors. Distinguishing these from other causes of coding errors is difficult. As well, examples of technical infeasibility of the Functional Specification are not always eliminated during coding and pre-acceptance testing, and manifest themselves as inadequate performance (type ENVI or SOFT), user confusion (type USER), or a request for an different implementation (PREF).
ENVI	Environmental defects are usually not related to the Functional Specification, except where explicit user interaction with the technology is defined.
DATA	As for SOFT, misunderstanding of the Functional Specifications may lead to DATA defects.

Table 4: Defect Class Relationship to Functional Specification

From this analysis, it is apparent that the Defect Classes that will be most directly affected by improvements to the quality and understanding of the Functional Specifications are SPEC and USER. While many of the other classes are also affected (such as SOFT), these are addressed through many other control techniques, and are therefore not the primary focus of this thesis.

3.4 Defect Frequency

The following table shows the actual defect frequency found during the first four days of Acceptance Testing of the product of ISAP (the ICON System).

Defect Class	% of all Defects
CR - Change Request	9.3%
PREF - User Interface Preference	9.9%
SPEC - Specification Defect	11.0%
USER - User Understanding Error	22.5%
SOFT - Software Defect	17.6%
ENVI - Environmental Defect	11.0%
DATA - Data Defect	2.7%
TEST - Tester Error	5.5%
DUPL - Duplicate Report	10.4%

Table 5: Defect Class Frequency

When taken together, SPEC and USER make up over one-third of the total.

It is clear that in this example, a typical software development project, improvements in the quality and level of understanding of the Functional Specification could lead to a significant reduction in the number of failures reported during Acceptance Testing.

Functional Specification defects cannot generally be detected through code inspections or technical testing (unit and integration). In some cases, technical staff may uncover inconsistencies prior to development, or may trigger the correction of ambiguities when they request clarification, but many defects appear only when users perform acceptance testing.

3.5 Sources of Defects in Functional Specifications

Functional Specifications are normally created by analysts, skilled systems professionals who are able to gather information from users and structure it a manner that can be used by system developers to create an automated system. Their work is essential to bridging the gap between users and systems developers: the users cannot understand the program code, the program specifications, or the programmers; at best, they can understand a carefully formatted and written requirements definition, or a Functional Specification. The system developers usually do not understand the business objectives, requirements and processes that will depend on the system.

A defect in the Functional Specification must first be introduced through an error during the gathering and recording of information in the specification, and then be overlooked during the process of reviewing the specification. The defect taxonomy shown above gives examples of some of the errors; these fall into a few categories:

Introduction Problems

- User participants do not fully represent the requirement domain.
- Users have different requirement domains.
- Users do not fully understand their requirement domain.
- Users are unable to express their requirements accurately or completely.

- Analysts misunderstand user statements.
- Analysts mis-record user statements.
- Analysts commit errors during creation of documents or models.

Review Problems

- Users are unable to detect defects.

- Users, Analysts or Designers/Developers do not adequately review the Functional Specification.
- No person reviews parts of the Functional Specification containing inconsistent information.

Most projects, and certainly any that are successful, attempt to

- include the participation of a representative set of users,
- establish a process for efficient transfer of information from the users to the analysts,
- express the information in the form of a document containing narrative and models, and
- remove the most egregious defects from the document through solitary or group reviews.

It is the last point that is the subject of this thesis. There have been many concepts, techniques and tools defined to support the transfer of information from users to analysts, and they are regularly and effectively applied in industrial practice. However, the process by which a Functional Specification is reviewed is still often unstructured or irregular. In most cases, the project team members who were involved in its creation are asked to read some sections and note defects, which are then passed to the author (usually the analyst) for correction. The degree to which defects are actually found and corrected depends entirely on the skill and effort of the individuals that happen to review a given section. To achieve consistently high quality in the Functional Specifications, projects must employ a process that can be expected to detect and correct the most defects possible given resource and time constraints.

There is a limit to the improvement to perceived quality possible through the reduction of defects in Functional Specifications. For example, users are often unhappy with decisions made by management to limit the scope, budget, technology or timing of a project. Project managers are unable, in the application of inspection techniques, to fully address user hostility; at best they can reduce it.

4 Quality in Functional Specifications

Many approaches have been used to improve the quality of functional specifications.

These include:

- application prototyping
- formal specification
- automated design documentation tools
- joint application design, and other interviewing and consensus-building techniques

In most industrial system development projects, there is also some manner of deliverable review or walkthrough. These can be considered informal inspections, as they approximate the activities and sequence of code inspections (preparation, review, and correction) but are usually without a fixed format, agenda, set of participants or outcome.

4.1 Inspection

Inspections have been used for some time to verify that code follows conventional usage, and has no syntactic or semantic defects that could be detected in a nearly context-free review. More recently, inspections have been extended to verify that code is correct in a rigorously defined context - using module interface specifications, and informal proof methods, the code can be inspected for the correctness of its implementation of the specification. Further, the module interface specifications can be inspected themselves, for both correctness and completeness, with resulting improvement in their quality and the quality of the code they specify.

Module interface specifications define the functionality of code "parts", the building blocks or objects in a larger system. They do not necessarily describe functionality that

will be apparent to a user. Functional Specifications, by definition, do, and they too may benefit from inspection.

By inspecting work products earlier in the system development process, I expect to be able to reduce acceptance test failures. The ISAP acceptance test failure reports included a significant proportion that were related to defects in the Functional Specification, but an even larger proportion were caused by user misunderstandings - the testers did not know what to expect and regarded unexpected behaviour of the system as a failure.

The Functional Specification is an intermediate work product that is used by two distinct groups - it tells the developers what to build and the users what to expect (and to include in their documentation). Both sets of team members spend time reading and questioning the specification during any project, usually during design and development phases. Their review activity can occur earlier in the project, as during an inspection process for the Functional Specification. The readers can be expected to achieve understanding, and identify defects, at an earlier point in the project.

This leads to the following objectives for an inspection of a Functional Specification:

1. Improve the internal quality of the document, by ensuring completeness (in the coverage of requirements), consistency, and clarity. This will reduce acceptance test failures of type SPEC.
2. Improve the degree to which users understand the document and can envision the system that will result from development. This will reduce failures of type USER, and possibly CR and PREF.
3. Improve the degree to which developers understand the document and are able to build the desired system. This will reduce failures of type SOFT, and possibly DATA and ENVI. I expect that involving designers and developers in functional

specification inspections will improve their productivity in ongoing quality management activities.

The inspection cannot be expected to replace other interactions between project team members and users. Furthermore, since the process by which functional specifications are created is dependent on "people" skills - interviewing, listening, reading and discussing, there will be variations in the inspection results arising from the different skill levels between specification teams.

4.2 Inspection Design Considerations

The design of the inspection process includes the following elements:

- Schedule** Most researchers agree that the cost-effectiveness of inspections lies in the reduced effort and cost required to correct defects early in the life cycle, rather than after the system has been put into production. However, the inspections must be scheduled in such a manner as to avoid impeding follow-on work. This means that:
- inspections should be scheduled in advance
 - allowance must be made for schedule changes if inspection products are not ready, and for notification of participants
 - attendance at inspection meetings must take priority over most ad hoc requirements
 - revision must take place immediately after the inspection meeting
- Scope** In addition to timeliness (holding the inspection as soon as the work product is ready), the inspection subject must be sufficiently small to allow the inspection meeting to be completed in ninety minutes. This goal is well suited to code inspections, particularly when modular design techniques have prevented code units from becoming

excessively large. It may also have been easily applied to earlier specification and design approaches that partitioned applications into components with clear and limited interfaces. The question of scope becomes more complex with modern integrated systems, in which functions share data structures and are therefore subject to implicit interactions.

Since in the Functional Specification the purpose is to describe the externally observable behaviour of the system, in a manner that can be understood and signed off by users, an appropriate limitation of inspection scope would be one that is recognizable to users - a single business transaction. This may include multiple system components. For example, a transaction entitled "Reserve Space" may include both a component that updates the *Space* data records and one that searches the records for appropriate inventory. Some common analysis techniques, such as Scenario Modelling or Use Cases, will normally define functional scope boundaries.

The inspection scope should be limited to those components that are required to complete the business transaction. For a component that is shared by many transactions (such as the search for inventory), the project manager may choose to remove it from the scope of an inspection if it has been covered sufficiently elsewhere.

Meetings

Because our objectives also include knowledge transfer and understanding (for which the reviewers will need to see and possibly discuss the resolutions to their defect reports), and because we do not have the homogeneity of the work product and participants normally found in a code inspection, we will:

- have participants prepare for the inspection meeting by reading the

document to increase their understanding; while they will not be expected to identify defects, they will be given the opportunity to do so or to raise questions about parts of the specification they do not understand

- require that the preparation reports, particularly the questions, be returned to the author prior to the inspection meeting to allow responses to be prepared
- include in the inspection meeting an overview and paraphrasing of each element of the specification as in the descriptive text in the Functional Specification; user participants will focus on the degree to which the specification meets the requirements definition, the analysts will examine the interactions with their own areas of responsibility, and the designers will inspect closely the logic and business rules that they must implement
- following the inspection meeting, direct the author to make changes to the document (with revision marking turned on) and distribute the updated version to all participants; each person must explicitly communicate their approval

Team

The Functional Specification is an intermediate work product, with ongoing discussion required between its authors and users, so concerns about reviewer overlap (and possible loss of inspection efficiency) will be less important than the need for broadly shared knowledge of the specification and the inspection results. Our initial inspection procedure will include:

- the analyst author
- the users that participated in the production of the specification
- analyst(s) and user(s) representing other functional areas that directly integrate or interface with the functions specified in the

inspection document

- the lead designer(s) who will be responsible for the detailed design and construction of the system components that fulfill the specification

Many inspection designs include a moderator and a scribe, who are responsible for managing the inspection meeting and recording defects, respectively. There may be value in these roles in an inspection of the Functional Specification, as there will be a tendency for discussions to extend into historical issues and related functions. The Project Manager or Leader, who is responsible for planning and organizing the inspection process, will act as moderator and scribe whenever possible. In a realistic setting, it may not be possible to have a distinct person for this role. The success of the inspection meetings in staying within the specification's scope, and within the time allotted, will be evaluated.

During the course of the inspection, the inspection team members will subjectively evaluate the relative contributions of, and value to, each group of participants; subsequent experiments may separate the analyst/user inspection from the designer inspection. Conversely, the variety of viewpoints may help smooth inconsistencies between teams regarding the thoroughness of the inspection and their definition of correctness or completeness.

Known Issues During the process of gathering information, modelling and documentation that results in the Functional Specification, business issues are often uncovered. In many cases, the resolution of these issues is not complete at the time that the document is ready, so that the potential for significant change to the specification remains. Most projects establish a mechanism for identifying and managing issues,

which allows the project team to make assumptions (based on default issue decisions) and embody them in the specifications. Therefore, every inspection must include a review of the outstanding assumptions that are subject to change - the Functional Specification document must explicitly identify all such assumptions.

Iterations

A Functional Specification is a model of the needs of a group of users for a system to support their business. It is common for that business to change, or for the users' perceptions of their needs to be changed by the system project itself, such that the original specification is no longer perfect (if it ever was) by the end of the project. This is manifested in the defect reports of type CR and PREF as discussed above, and is the basis for ongoing enhancement and change to the finished system.

Business improvement practice in recent years has emphasised speed - time to market, turnaround time, customer service response time - in recognition of the value of time. A strong systems project sponsor will often declare that "we are after the 80% solution" in recognition of the limited resources, particularly time, that are available to the project.

By including a broad audience in the inspection process, and by including both preparation and follow up activities, I expect to be able to avoid multiple iterations of inspection while achieving an acceptable degree of defect detection and correction. However, once the process is instituted, a successful inspection will be of fundamental importance in obtaining user management approval of the functional specification.

Therefore, the process described above will be repeated only as necessary to obtain approval.

Training and Mechanics All of the participants will need to be trained in, at a minimum, the process to be followed and the forms to be used. The training session will be performed just before the initiation of the inspection activity, and will employ materials adapted to the project.

At this stage of system development, our only inspection techniques are:

- checking the reports produced by the design automation (CASE) tool
- reading

Consistency checking and defect reporting from the CASE tool is very useful to the authors of the specification, and will allow the elimination of such activity from the inspection procedures. However, there are other outputs from CASE tools that may be valuable to the inspection - descriptive reports and prototypes. The descriptive reports restate some of the information contained in the metadata repository in a tabular or narrative form, and can reduce the amount of effort required by the author or reader to create understanding. Many CASE tools are able to generate prototype screens or reports without significant effort, and these may also help illustrate the functionality of the system as specified.

Prototyping is valued for its contribution to communication with users. It brings the danger that, if used in an inspection meeting, participants will focus on presentation details rather than the underlying specification.

Because each project employs different technologies for design, it

makes sense for the project manager and analysts to decide upon the degree to which CASE will contribute to inspections.

Relevance is addressed by involving as inspectors those individuals who are directly affected by the quality of the Functional Specification, but to ensure that participants contribute as much as possible, the format of the specification and its corresponding inspection report must be well structured.

Forms

The inspection process, to be as efficient as possible, should employ technology for recording and managing defects. The inspection checklist should also be closely tied to the structure of the Functional Specification itself (and when possible, the Functional Specification should be structured to facilitate inspection). To guide the participants, the checklist should contain a list of the specification elements, with:

- a description of the element
- the purpose it serves
- any standards defined for the element
- the measures of completeness and correctness for the element

At each point in the checklist, and at the end for summary purposes, the inspector should be given the opportunity to insert questions about the element and reports of any defects detected.

4.3 Inspection Design

The inspection design comprises three aspects:

- The Inspection Process defines the activities required to prepare for and undertake inspections
- The Inspection Elements define the characteristics of the work product to be inspected, in terms of both form and content
- The Inspection Evaluation defines the feedback that will be used to improve the Inspection Process and Inspection Elements.

4.3.1 Inspection Process

The inspection design considerations are consolidated into the following process:

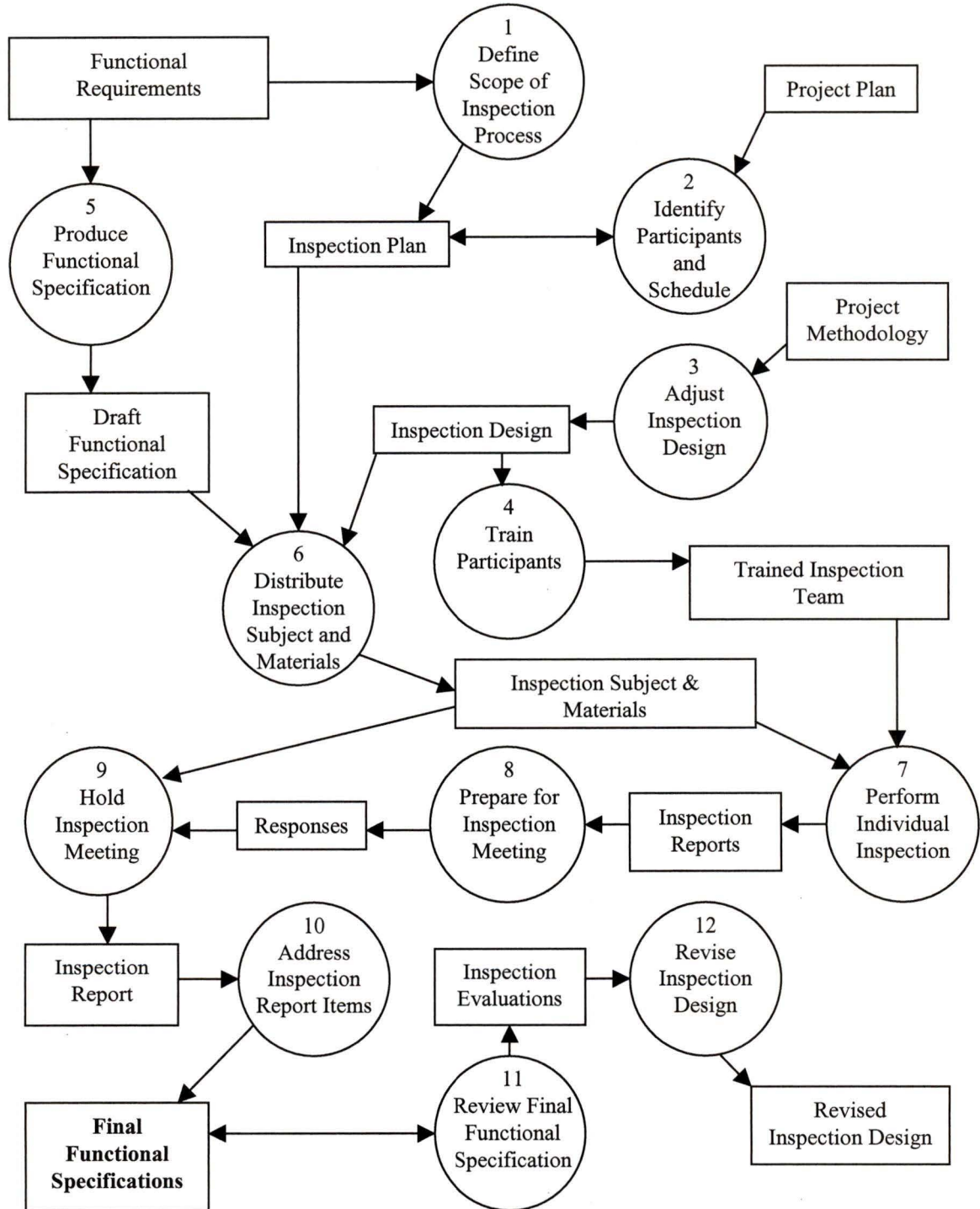


Figure 1: Inspection Process

Here are additional details of the process:

Activity		Participants	Comments
1	Define scope of inspection process	Project Manager	Identify which elements of the Functional Specification will be inspected, and group according to relevance and size. Create an initial Inspection Plan showing the Inspection Groups and their components.
2	Identify participants and schedule for inspections	Project Manager	Use the project plan to assign inspection team members and schedule inspection meetings. Add to the Inspection Plan.
3	Adjust inspection design to fit project methodology and tools	Project Manager, Analysis/Design Tool Specialist	Review the project methodology, the outputs of any analysis and design tools, and previous inspection designs. Produce a revised Inspection Design (process, elements, and evaluation).
4	Train participants in inspection process and techniques	Project Manager, all inspection participants	Review Inspection Plan and Inspection Design with all inspection team members.
5	Produce Functional Specifications	Lead Analyst	Use standard format from project methodology.
6	Distribute inspection subject and materials (inspection forms)	Lead Analyst	Send to all inspection team members. Include background documentation (Requirements Definitions, Data Models) that would assist in understanding.

7	Perform individual inspection	Other Analysts, Users, Designers	Read and complete Inspection Report.
8	Prepare for inspection meeting	Lead Analyst	Review Inspection Reports - prepare corrections to defects and responses to questions.
9	Hold inspection meeting	Lead Analyst, Other Analysts, Users, Designers	Moderator / Scribe will control discussion. Defects identified during meeting will be addressed following meeting unless trivial.
10	Address inspection meeting items	Lead Analyst	Review Inspection Report from meeting and revise Functional Specification. When complete, distribute to inspection team.
11	Review Final Functional Specification	Other Analysts, Users, Designers / Developers	Confirm that all defects and questions are appropriately addressed in Final Specification. Complete and submit Inspection Evaluation.
12	Review inspection results	Lead Analyst, Project Manager	Review Inspection Evaluations and revise Inspection Design as necessary.

Table 6: Inspection Process Details

There is clearly an expectation that the Project Manager is able to effectively adapt the inspection process to the project, both in identifying appropriate scope and participants and in defining the effects of the project methodology and tools on the process. In the latter task, the Project Manager may need assistance from a tool specialist or others who can contribute to the structure of the Functional Specification. The best help may be from examples of earlier projects, and these will also prove valuable for training the inspection team. The Inspection List and Plan do not have a particular format - they are

simply lists of sections of the Functional Specification and the individuals who will take part in inspections of each one.

4.3.2 Inspection Elements

The Functional Specification is a complete description of the observable behaviour of a system. In any such specification there will be sections that describe common elements (for example, standard user interface features) and others that relate to specific functionality. While all sections would benefit from inspection, the greatest proportion of effort will be required to inspect the specific functionality sections. The Inspection Elements are designed for this task. Some features, such as Security, would usually be specified in a global sense but would also allow for optional override of the general approach, so the Inspection Elements must also address these features. Other non-specific areas, such as Operations Procedures and Acceptance Criteria, are not inspected.

The Inspection Elements, in the form of the Inspection Report, are shown in Appendix A.

Both the Inspection Process and Inspection Elements may depend on the design approach. While the Inspection Elements used for this paper imply the separation of process and data models, an object-oriented approach would encapsulate them during design. The Inspection Elements are worded in a manner that attempts to eliminate any dependency on a particular documentation technique or design approach.

Data Conversion specifications can be inspected in the same manner as the production system specifications, but some criteria (such as User Interface) will not apply or must be modified, and the inspection team may be smaller.

4.3.3 Inspection Evaluation

There is value in establishing, from the outset, evaluations by the participants as part of the Inspection Process. These will serve several purposes:

- Capturing the impressions and suggestions of the variety of participants included in this type of inspection, rather than only the people managing the process
- Providing explicit feedback of results to allow adjustment of the Inspection Design
- Institutionalizing the continuous improvement of the process, and the recognition that it must be adapted to each project.

An Inspection Evaluation form is included in Appendix B. It would normally be distributed at the same time as the Revised Functional Specification, so that the inspector's comments are fresh. In a large-scale implementation of the inspection process, the evaluations may be limited to a subset of the scope.

5 Applying the Inspection Approach

5.1 Research Plan

The objectives of this research were to decrease defects in the Functional Specification, and increase the level of understanding among users and designers. Determining whether the experiment achieved the goal is difficult, since there can be no control with which to compare results. While the project contributes to a body of results that can be analyzed statistically to show improvements, the greatest value may be in the anecdotal results. Some quantitative comparisons may be possible if this technique is applied by organizations that undertake projects with similar characteristics.

The experiment included the following tasks:

Define Experimental Project and Obtain Approvals	Permission was granted by British Columbia Buildings Corporation to use results from the Information Systems Alignment Project (ISAP) and to include its staff in the review of the draft Inspection Design. Permission was granted by the Ministry of Finance and Corporate Relations to apply the inspection process to the Family Bonus Renewal project and to report the results.
Create Draft Inspection Design	A draft version of the Inspection Design: Process, Elements and Evaluation, was created based on previous research and the author's experience.

Distribute Draft Inspection Design	<p>The draft Inspection Design was distributed to selected analysts, users and designers, many of whom were active participants in one or both of the experimental projects.</p> <p>Each person was asked to review the draft Inspection Design to identify whether they felt they could use it effectively, and to indicate suggested improvements that would make the design more efficient or more likely to improve quality.</p>
Update Inspection Design	<p>A number of changes and clarifications were suggested by the reviewers and were incorporated into the Inspection Design and its documentation. These revisions are described in Chapter 6 - Results.</p>
Implement Inspections	<p>The inspection process was applied to the Family Bonus Renewal (FBR) project in Phase IV, the creation of the System Delivery Specification. In all, fifteen project team members were trained and participated in the inspection.</p>

The inspection implementation followed the process shown in the Inspection Design. Here are some considerations that were specific to the FBR project.

Activity		Comments
1	Define scope of inspection process	The scope included all online, batch and report processes. They were collected into groups for inspection. The groupings and schedule are shown in Appendix D.
2	Identify participants and schedule for inspections	<p>Taking into account the size and physical distribution of the project team, the inspection schedule was arranged as follows:</p> <p>Day 0: lead analyst submits inspection subject and supporting documentation to project office</p> <p>Day 1: project office distributes subject, supporting documentation and inspection report forms to inspection team</p> <p>Day 4: inspection team forwards completed inspection report forms to project office and lead analyst</p> <p>Day 6: lead analyst prepares responses to defect reports and questions</p> <p>Day 7: inspection meeting held</p> <p>Day 9: lead analyst completes final changes to inspection subject and forwards to project office</p> <p>Day 10: project office distributes inspection subject and inspection evaluation forms to inspection team</p> <p>Day 11: inspection team forwards confirmation that expected changes have been made, and a completed inspection evaluation form, to the project office</p>
3	Adjust inspection design to fit project methodology and tools	Some minor wording changes were made to the inspection report form to correspond to the Ministry's own terminology.

4	Train participants in inspection process and techniques	Training would normally be provided by the Project Manager or an analyst who is proficient in the process. Because the FBR project is the first use of this inspection design, I provided training to all participants, including the Project Manager. The objectives, process and forms were presented and questions were entertained. All of the systems professionals expressed a belief that the process would be beneficial; the users accepted the process as necessary on the basis of the Project Manager's direction.
5	Produce Functional Specifications	A standard template was defined for System Delivery Specifications, with variations for online, batch and report processes.
6	Distribute inspection subject and materials (inspection forms)	As described in the schedule above, this was done by the Project Office on behalf of the lead analyst. All materials were distributed electronically.
7	Perform individual inspection	Individuals returned their completed inspection forms, usually in hardcopy format. See the Results chapter for a review of the inspection quality and timing.
8	Prepare for inspection meeting	Analysts used the inspection reports during their meeting preparation, generally scanning for substantive defects or questions.
9	Hold inspection meeting	Inspection meetings were held with the timing and team as planned. During the course of the process, the inspection subject underwent some change. Details are provided in the Results chapter.
10	Address inspection meeting items	Inspection meeting items (remaining defects or other problems with the SDS) were addressed immediately by the analyst, and an updated version was created.

11	Review Final Functional Specification	The review of the final version occurred in two distinct ways, as decided by the team. For specifications with significant defects, a second inspection meeting was held for the relevant subset of the specification. Specifications that had no significant defects were not explicitly re-reviewed; the final version was made available in a shared file for any inspection team members that wished to confirm that the changes they expected had been made.
12	Review inspection results	The inspection reports and evaluations were reviewed by the Project Manager and myself. Our observations are included in the Results and Conclusions chapters.

Table 7: Inspection Process Implementation Considerations

Overall, the inspection process conformed closely to the original design. The most important changes were in the areas of:

- scope, as the inspection groups changed over time,
- timing of the inspection steps, and
- review of the final version of the specifications.

6 Results

6.1 Comments on Draft Inspection Design

The first draft of the Inspection Design was reviewed by eight people who had worked on ISAP or similar projects. They included four analysts, three designer/developers and one end user.

All of the reviewers commented that the idea of applying inspections to Functional Specifications had promise. Some felt that many projects would not be able to afford the time to conduct formal inspections, while others believed that the ISAP project would have been more successful if we had used an inspection approach.

The most significant change arising from the review, which was suggested by three reviewers, was to include a moderator/scribe role in the inspection meetings. They felt that the inspection meetings would fail to complete successfully without someone to moderate the discussion, because of the tendency of the team to explore solutions, rather than simply identify defects.

Two reviewers suggested that there be multiple iterations, as they believed that the mutual understanding gained by analysts would result in changes to their own specifications, and would thus require another round of inspection. I chose not to adopt this approach, but emphasized during training that the inspections are intended to be a final check of the deliverable, and are not a substitute for communication between analysts.

Some of the other comments, and my responses, were:

- Several reviewers requested a more precise definition of the Functional Specification, its contents and quality criteria. To respond, I revised the wording of several sections, and modified the Inspection Report form.
- There was some confusion regarding the processes for creation of the deliverable and its inspection. I created the flow chart shown in Chapter 4 to help illustrate the process, and to clarify the inputs and outputs of each step.
- One inspector pointed out inconsistencies in terminology related to Inspection Elements and role titles. These have been corrected.
- The original inspection design did not adequately discuss the role of the Project Manager. I added comments about the inspection planning, customization and training activities of the Project Manager, but decided not to specify that the Project Manager act as moderator (as had been suggested) in recognition that this may not be appropriate for some projects.
- Two reviewers made conflicting suggestions regarding the scope of inspection, particularly with respect to Data Conversion. One person felt that the inspection approach, because it involves a large number of people, would not be appropriate for areas like Data Conversion because of the limited audience that would find it useful. The other suggested that as much as possible of the Functional Specification be inspected, believing that the inspection infrastructure and process flow would lead to efficiencies that should be fully exploited. I have chosen not to strictly specify the scope of inspections, again because each project will have different characteristics. I expect, however, that inspecting Data Conversion specifications would often be cost-effective, because of the difficulty in identifying defects in this area.

6.2 Inspection Process

The comments that follow include both implementation considerations and the observations obtained during the inspection activities.

Scheduling inspections proved to be difficult because:

- the inspection teams were relatively large (6-10 people),
- while the Functional Specification was being created there were other tasks underway (business process change implementation, test planning, data cleanup) that imposed conflicting time demands on team members, and
- the time of some participants was not entirely available to (and controlled by) the Project Manager.

Therefore, early scheduling of the inspection processes was needed to ensure attendance. This also had the advantage of providing the analysts with fixed targets and a heightened sense of urgency.

The disadvantage of a fixed date was observed in the reluctance of the analyst and Project Manager to reschedule the first inspection, and they proceeded with a specification that was not ready. As a result, the first inspection meeting suffered from the following shortcomings:

- Inspection Reports were not consistently filled out (lack of time, and too many defects).
- Because of the number of defects and the lack of earlier reviews with users, the inspection meeting resulted in a review of each sentence of the specification and did not cover the scope that was intended in the time available.
- There was a great deal of discussion of defect corrections, rather than identification of defects alone.

The meeting was therefore recognized to be a walkthrough, rather than an inspection. Although it was not planned, the exercise proved to be valuable in illustrating the distinction to the team members, and residual questions such as the difference between a defect and an outstanding issue were addressed. The results prompted the analysts and users to reconsider the process they planned to use to complete other specifications, in many cases adding explicit walkthrough sessions prior to inspection. A second iteration for the first inspection subject was scheduled.

Another improvement to the process was the agreement by the analysts that they would, whenever time permitted, bring corrected or improved versions of their Functional Specifications to each inspection meeting, incorporating the responses to defects and questions in the Inspection Report. This allowed the inspection team to observe immediately whether the correction or clarification they expected had been included.

The Inspection Reports and comments at the inspection meeting included a surprisingly high proportion related to manual processes. This probably occurred because of emphasis made in training that one purpose of the Functional Specification was to support the users in the development of user procedures and documentation. They carefully checked each interaction with the automated system, whether explicit or implied in the Functional Specification. To assist them in this work, the teams agreed to be vigilant in keeping any technical implementation details in a separate section of each specification document. This was part of the standard approach to Functional Specification (in which any implementation considerations are isolated), but it had not been consistently followed in previous projects. As well, each manual process that preceded or succeeded an automated process was identified and named. This will likely ease the effort required to produce the user procedures and manuals.

A beneficial effect of the broad membership of the inspection teams was the frequent identification of scope creep - areas where users expected automation that had not been planned during Requirements Definition. These were found and corrected because the Moderator was frequently the Project Manager, and others were involved who were aware of the approved scope and responsible for managing it.

Analysts responded to the formal scrutiny of their products by improving the process by which they created the specifications, and by increasing the degree of detail they included. This initially exceeded the standard for the Functional Specification, in that some internal designs were defined, and user interactions with the system were described

to an unnecessary level of detail for this stage of the project. To keep the analysts working toward an appropriate target, we had to review their draft specifications regularly in the first few weeks of work.

As the revised inspection dates approached, the user community requested an extension to the time allotted for creation of the Functional Specification. Both they and the analysts felt that to perform effective inspections, a greater degree of detail was required in the specifications than we had set as a target.

Since the additional work would mostly be required later in the project (during Detailed Design or in the development of user procedures), the Project Management Committee approved the extension.

This change had beneficial effects:

- The analysts spent more time in walkthroughs and reviews of the draft specifications prior to inspection, so that inspection reports identified fewer defects and questions.
- The users had sufficient time to understand the specifications prior to the inspection meeting, and thus were content to review identified defects and questions rather than walk through the entire specification in the meeting.

These effects contributed to the achievement of the objectives of the inspection process (quality of the Functional Specification and better user understanding). The application of an inspection process may therefore have been as valuable in its effect on team behaviour prior to inspections than in the inspection activities themselves.

6.3 Inspection Results

The inspection results observations are derived from review of the inspection reports created by the inspection team members and observation of the inspection meetings.

6.3.1 Inspection Reports

Inspection reports were completed diligently by the inspection team members. In most cases they preferred to work with paper versions of both the Functional Specification and the inspection report form. This caused some instances of missing pages and distribution delays.

The analysts and system developers generally followed the inspection report format closely. They reviewed the inspection criteria and used the separate defect and question listing pages to record details. Representative samples of each type of inspection report are included in Appendix E.

The example shown of an analyst's inspection report is interesting in its use of the generic "Other Comments" section to classify most defects and questions. Some of these comments could readily have been listed under more specific criteria.

In discussions with the inspection teams, I found that the failure to classify defects and questions was caused by three factors:

- Users preferred to record comments in the order read in the specification, rather than scanning for defects related to each criterion.
- The inspectors (both users and technical staff) did not perceive a need to classify defects, and the effort required for careful classification was not always trivial.
- The inspectors often felt unsure of their comments and did not wish to be seen as unjustly criticizing the specification.

The last point is significant. Unlike code inspections, where defects are most often unarguable and inspectors need not have an understanding of the problem solved by the deliverable, our inspection of Functional Specifications revealed many cases where the

apparent defect was actually correct. The inspectors were often unable to determine with certainty whether the problems they found in the specification should be defects or questions, so they avoided the problem by making unclassified comments.

Nevertheless, the range of comments covered all of the inspection criteria, and they were usually very precise. In this respect, they strongly resembled code inspections. Many of the users' comments pointed out incompleteness - areas of the Requirements Definition (Business Requirements Definition or BRD in the project's terminology) that had not been addressed in the Functional Specification. The analysts' comments were more frequently related to logical inconsistency or ambiguous wording.

6.3.2 Inspection Meetings

Inspection meetings were run as planned, with a moderator ensuring that the discussion was limited to identifying and clarifying defects, rather than resolving them. The moderator also acted as scribe, and produced inspection reports that listed any outstanding or new defects. An example is shown in Appendix F.

For one meeting, no moderator was available. As predicted by early reviewers or the inspection process, the discussion was not as effective, with the result that the inspection scope was not completed. In all other meetings, the scope was completed, often in less time than allocated.

The team had requested that the inspection groups include a larger scope of functionality than originally anticipated, in order that cross-inspection of related specifications was easier. This resulted in inspection meetings that were longer than the planned ninety minutes. The shortest meetings required thirty minutes, but the longest required three hours.

Unlike code inspection meetings, cosmetic and typographic defects were not discussed, at the request of the inspection teams. Before meetings, inspection reports were scanned for design defects and questions, and the author of the specification was responsible for correction of the remaining items, all of a trivial nature, after the meeting.

The most frequent comment about inspection meetings was the failure of the project to meet its inspection process schedule for distribution and collection of materials. As a result, some inspection reports were brought to meetings rather than given to the analyst in advance, and in some cases the inspectors did not feel that they had sufficient time to do a thorough job. The schedule problems resulted from:

- the need to fix inspection meeting dates,
- specifications becoming ready at much the same time, and
- the short period (two weeks) over which inspections occurred.

This meant that inspectors (who in most cases were involved in creation of the specifications as well) had to perform a large number of inspections in a short period, rather than throughout the project phase as is normally the case with code inspections.

The integrated nature of modern systems will often lead to the same schedule issues. It is not practical to plan a simple sequence of inspections when the quality of the first specification may depend on the content of the last specification. Projects that use inspections for Functional Specifications will need to plan for this situation.

The aggregation of scope into larger inspection subjects, to cover an entire range of business functionality, will exacerbate the problem, both in terms of the schedule and in the makeup of the inspection team. Because of the broader scope, the inspections included more people than planned. However, the increased size of the inspection team was valuable, as the important objective of increasing understanding was more widely achieved. As most of the discussion in the inspection meeting was related to questions, misunderstandings or unclear specifications, rather than defects, a large group of team members was present to hear the explanation.

The team members also actively helped each other during the meetings. On several occasions, answers were provided not by the author of the specification but by the users who provided input to it. In other cases, a poorly formed question by a user was restated by one of the analyst or system developer inspectors, with the result that an important defect was identified. This was particularly true when discussing logical processing cases, where the users did not press for complete coverage of the input domain, likely because they did not recognize the need to do so.

The level of understanding was also increased between members of a group. Many of the questions raised by analyst or system developer inspectors related to interfaces between the inspection subject functions and their own area. Users, too, exchanged important information, such as one situation where a user had decided to drop some automated functionality (preferring for cost and complexity reasons to use a manual process) but had not told other users who would be affected by the decision.

Relatively few new defects were detected at the inspection meetings. However, the fact that a formal inspection meeting was to occur, after which the specification would be considered final, led the inspectors to be much more diligent than in offline reviews conducted in other projects. Had the meetings not been included in the process, each team member's effort or lack thereof would not have been apparent.

6.4 Inspection Evaluations

All of the participants in the inspection process were asked to complete evaluations. The format of the questionnaire is shown in Appendix B, and some representative samples of completed evaluations in Appendix G.

Although there was general agreement that the objectives of the inspection process had been achieved, the comments made by each participant varied considerably. This appears to be correlated to differences between the type of processes involved in each design and inspection group.

For example, those individuals who took part in design of reports generally valued the inspection process more highly than those who were involved in design of batch processes. This is probably because, for this particular system, the batch processes are very complex and include the application of detailed business rules, while the reports are less complex and have relatively few business rules that govern them.

All of the participants cited a lack of time as a shortcoming of the inspection process, and in some cases suggested that the quality of the Functional Specification was adversely affected by insufficient time for inspection. Other inspectors felt that the compressed schedule was offset by additional effort on the part of the inspection team members. The problem was apparently worst for inspections that occurred near the end of the schedule, as slippage in earlier work had reduced the time remaining, yet corrections arising from earlier inspections had an impact on the outstanding specifications.

Users noted that not all defects had been resolved by the end of the inspection meeting. This was as planned in the process (except for the decision to deal with cosmetic changes offline), and the same inspectors allowed that the required changes appeared in the final version of the specification.

Users also consistently agreed that their level of understanding was improved. They particularly commented that their knowledge of the overall system development methodology was greater, as was their understanding of the concerns of other stakeholders (the client's Information Systems department and the PricewaterhouseCoopers design team). Several analysts felt that the greatest

improvement in user understanding was in their knowledge of their own business, especially where users' job functions did not entirely overlap, and the inspection process was their first opportunity to review each other's business procedures.

Analysts split in their opinion of whether user or designer understanding was increased. Some felt that the designers did not necessarily understand the user requirements any better, but that the degree of interaction and level of detail in the Functional Specification would result in a faster and better Detailed Design.

Other analysts wanted more time with the stakeholders of each specification. One felt that the same people should be involved in both design activity and inspection, but suggested that the inspection teams should be smaller. This approach, which could be considered a walkthrough, is useful in the creation of the specification but is not an inspection.

One analyst expected a decrease in acceptance test failures arising from user misunderstanding, because of the extensive explanations required before the inspection process was complete. Another expected an increase in acceptance test failures arising from defects, not because of a reduction in quality of the system, but because they anticipated that users would be more thorough in their testing as a result of their heightened awareness of quality processes and their degree of involvement.

An interesting comment was that "more effort gives the perception of more quality". This could be interpreted cynically to suggest that poor quality is hidden by the bustle of activity in the inspection process, or more optimistically to indicate that by having a rigorous and open process, users are more likely to understand the quality characteristics of a system and be more comfortable in accepting it. An inspector commented that the process forced people to speak up, as it provided an official opportunity and impetus to identify defects, rather than remain quiet or never be consulted. Another inspector felt that the inspection process itself did not contribute to defect detection or user

understanding; she believed that the high degree of user involvement, compared to other projects, was the contributing factor. In discussion, she agreed that the inspection process helped to formalize and ensure user involvement and was more successful in obtaining consistent effort from the user community than less formal processes used in other projects.

The degree of interdependency between inspection groups had schedule implications, as noted earlier. One reviewer suggested that more time be spent on examining the interfaces and connections between specifications in distinct inspection groups. This was noted as particularly important because of a peculiarity in this project, in the client's insistence in producing an early physical data model (despite the methodology's direction that only a logical model be produced), resulting in confusion during discussion in which both logical and physical data model elements were used.

After the last inspection and the production of a complete System Delivery Specification, the entire project team took part in a workshop that reviewed every component of the finished document to confirm that a complete set of specifications had been produced, and that the correct versions were in place. The team also reviewed each issue and assumption that appeared in the SDS to confirm whether the assumption was valid. During this workshop, a total of fifteen modules were noted (out of a total of 157) that had incomplete or missing specifications. Most of these incomplete modules were last-minute additions, such as the recognition during inspection that a sub-process would be required to extract data for the report being inspected, or that a utility process was needed to allow submission of the batch process under inspection. Other modules were apparently identified earlier in the process, but the specifications had not been completed because there were no resources available to do so.

Late identification of processes frequently occurs in system development projects, including ISAP. In many cases, they are identified only after further work has been completed (sometimes as late as testing), rather than at the end of SDS. By noting the

omissions at this point in the life cycle, we were able to make reasonable approximations of the effort and cost associated with them and include those costs in our quote for the remaining work. Had we not identified them until later, the cost of the scope change would have represented a serious issue for the client.

6.5 Results Summary

The results of applying the inspection process must be reviewed against the original design considerations and its objectives. Here are the design consideration categories and the results observed for each.

Schedule The scheduling of inspection activities, overlapping the end of the Functional Specification, was not entirely successful. Although the inspections occurred as planned, this was due to the project team's willingness to complete their inspections in shortened time frames. More explicit planning of inspection process steps and balancing of load on team members would have produced better results.

The plan for new applications of the inspection process should include a distinct set of tasks, scheduled immediately prior to the end of the Functional Specification phase, and not overlapping other project activity to any significant degree.

Scope The inspection teams felt that large-scope inspection groups were most valuable, and this was embodied in the inspection plan. This was needed to allow inspectors to view the interrelationships between specifications in a business area. The inspection groups were in fact defined in terms of the groupings identified in the Requirements Definition, and this approach should be used in other projects.

- Meetings** The inspection meetings produced the expected results. Few new defects were identified in the meetings, but the many questions were addressed and the meeting served as a fixed milestone that focussed the attention and effort of the design and inspection team.
- Inspection meetings, with a distinct Moderator / Scribe, should be included in all new inspection processes. Whenever possible, the Moderator / Scribe should be one of the project leadership, and should not have any design responsibilities.
- Team** The inspection teams were large, as anticipated, and good representation was available from all project sub-teams (users, analysts and system developers). The team interacted appropriately and the amount and quality of communication was better than expected, because of the team's recognition of the need to achieve consensus prior to the inspection meeting. All inspection processes in new projects should have teams constituted as designed here.
- Known Issues** Issues were properly identified, and in most cases reasonable assumptions were made. The workshop at the conclusion of the phase reviewed the issues, and found that none required the attention of the Project Steering Committee. This may have been largely due to the nature of the project, but was also the result of the level of user involvement in the design and inspection work. This approach to issue identification should be continued.
- Iterations** One iteration was sufficient for most inspection groups. Where a second inspection meeting was required, it was usually a result of failing to complete the original meeting in the time available, rather

than because of excessive defects in the inspection subject. The first attempted inspection meeting was an example of the latter, and illustrated a need for changes to the training program (see below). The inspection plan should reserve time for additional inspection meetings for a small number (less than 50%) of the inspection groups.

Training and Mechanics

Training was delivered as planned, but it was clear that many of the participants did not understand the process at the conclusion of the training. The most valuable training activity was the first inspection meeting, which was unsuccessful as an inspection. This indicates that a "practice" inspection would be a useful approach to training the inspection teams, especially if examples of Functional Specifications and Inspection Reports were presented to the teams during the session. The training must also recognize differences in the backgrounds of the team members, and take explicit steps to address inconsistencies in expectations (for example, of the level of detail required in the Functional Specifications).

New projects should implement the original training process augmented with a practice inspection.

Forms

The inspection forms were used by all participants, but many of them (especially the users) did not classify defects and questions. The original intent in doing so was to ensure that all of the quality elements were given appropriate attention.

This is also a training issue, in that good examples of inspection reports, with illustrations of representative defect classifications, would have helped clarify usage of the forms.

The problems with distribution would have been alleviated had the electronic forms been employed. New projects should test the electronic distribution and insist that all specifications and forms be handled electronically.

The results of the inspection process in achieving its objectives were:

1. *Improve the internal quality of the document, by ensuring completeness (in the coverage of requirements), consistency, and clarity.*

Most of the inspectors agreed that the process achieved all aspects of this objective. Completeness was achieved at the final workshop, when missing modules were identified. Consistency and clarity were greater than in any other Functional Specification in my experience.

2. *Improve the degree to which users understand the document and can envision the system that will result from development.*

Inspectors' opinions varied, but most believed that this objective had been achieved. The measurement of user understanding is affected by the object of the understanding (their business, the document, or the system that will result) and the definition of "understanding" (direct contribution to the result, complete comprehension, or awareness).

By one measure, the effort and time required to obtain approval of the SDS, this project was extremely successful. The ISAP project required nearly a month, while this project (FBR) needed only a day. Even after accounting for differences in the structure and history of the projects, it is clear that the level of user understanding was such that they

felt comfortable in committing their organization to contracting for the system embodied in the SDS.

3. *Improve the degree to which developers understand the document and are able to build the desired system.*

While some inspectors (notably the system developers) did not feel this objective was achieved, I believe that their comments are based on comparison of levels of understanding at different phases of the project. In most projects, including ISAP, the SDS concluded without significant involvement from the system developers. A great deal of work was required by them during subsequent phases to understand the document and obtain clarifications. Some of the clarification did not take place until system testing.

In this project, the same level of understanding has been achieved before the end of the Initial Design (SDS) phase. While the system developers are still faced with technical challenges in implementation, they are already aware of the challenges and understand the goals that the users have set for the system.

7 Conclusions

The summary in the previous section showed that the original objectives of the experiment were achieved, either through the inspection process itself or through changes in the behaviour of the project team that were triggered by the process. More defects were detected and corrected, and both users and developers better understand the Functional Specification, than at this point in other projects. These results were achieved on a business project of sufficient scale and scope that we can expect similar results on future projects of PricewaterhouseCoopers.

To establish whether inspection of Functional Specifications should be used again, the results must be evaluated against the cost, to determine whether there is value to the project. Additionally, those factors that are critical to the successful application of the technique must be identified if the value is to be realized.

7.1 Value of Inspections for Functional Specifications

The organizations that supported and participated in this experimental application of inspections did so not for the purpose of obtaining interesting results for a thesis, but in the anticipation that the technique would lead to immediate improvement in the quality of their projects and would prove to be a valuable addition to their development methodologies. In the business of systems integration consulting, each project differs in significant ways from all others, so meaningful quantitative comparisons are typically not feasible. For PricewaterhouseCoopers, the measure of success in applying a new technique can only be the opinions of the professional consultants – project managers, analysts, and developers – who take part in the experiment and evaluate the result against their previous experience. By this measure, as shown in the comments received from the team, this experiment was a success.

An important consideration is any incremental cost that a project bears by using the inspection technique. If this cost outweighs the value of the quality improvements, it should not be used.

The incremental cost of inspecting Functional Specifications cannot be extrapolated from data obtained from code inspections. Developers need not perform inspections of others' code to be able to complete their own programming assignments. To be cost-effective, the incremental work they perform in completing inspections must be offset by reductions in effort elsewhere.

In contrast, inspection of Functional Specifications may result in no incremental cost to the project. The work undertaken by the inspection teams was, in every case, necessary to the completion of a quality system. This is because the inspection process was designed not only to detect defects, but also to increase understanding, and the inspection teams comprised the same individuals who will transform the Functional Specification into programs and user procedures.

Communication of the Functional Specifications to the project team members who use it to create other work products must occur in every project. We chose to include the communication effort in the phase that produced the Functional Specification. As a "side-effect" of the formal nature of this communication effort, defects were detected and corrected in the Functional Specification, with the result that it is more complete, consistent and clear than any such document I have seen.

This quality has a direct impact on the risk and cost of the project. Most significantly, the risk that unanticipated programming effort will be required is lower, and thus the contingency applied to the plans for project completion can be smaller. This translates directly into savings for the client, and a greater expectation of a financially successful project for PricewaterhouseCoopers. A clear indication of the effects on risk and cost is the unusually quick sign-off period achieved on the FBR project, illustrating the

confidence of the client in the process and result, and avoiding the costly delays that often occur at major project milestones.

As a result, formal inspection of Functional Specifications will become a standard quality management technique for medium- and large-scale custom development projects undertaken by the Custom Delivery practice of PricewaterhouseCoopers in British Columbia. Excerpts from this thesis will be added to the *Summit D* repository used by our project teams.

I will also be including a tutorial on the inspection process in a seminar on advanced project management for experienced managers that I will be leading at PricewaterhouseCoopers' Canadian training series ("Consulting University") in November 1998.

7.2 Practical Issues of Implementation

Inspection is one technique that contributes to the completion of the Functional Specification, as part of an overall system development methodology. It cannot be successful unless other activities and other techniques are organized to produce the inspection subject. A key element of this methodology is a definition of the structure, content and format of the Functional Specification that is understood by all authors and inspectors.

The threat of inspections is as valuable in meeting the quality objectives as the inspections themselves. When project team members are aware that they will be required to publicly present their work, and respond to comments and questions about it, they will take steps to ensure that the work product is ready. The project methodology (and Project Manager) must provide guidance to the analysts so that the preparatory work - interviewing, writing and walkthroughs - is efficient and consistent.

To be most effective, the inspection implementation must include:

- leadership by an individual who has previously taken part in such inspections, understands the objectives of the process, and is able to train the inspection teams,
- detailed planning of the inspection activities and schedules that recognize the impact that the inspection process will have on individuals' behaviour and the work load they will bear during the inspection period,
- a concerted training effort, using concepts, examples and practice inspections to ensure that all participants have a common understanding of the process and their role in it,
- coaching and assistance to the analysts and inspectors so that they view the process as an important part of the work needed to produce specifications, not as an evaluation of their own capability or performance, and
- efficient document control and distribution mechanisms that minimize overhead and maximize consistency in producing and inspecting specifications.

Finally, the value of inspections must be universally understood. In the project that first applied the inspection process, no resistance was presented by the Project Sponsor, Steering Committee, user management or technical management. This happy situation may not always occur, so the Project Manager may need to incorporate education sessions in the events leading to the approval of the Project Charter.

7.3 Areas for Further Research

This inspection process has been tested on a single project. More experience is needed, from different types and sizes of projects, to confirm the value of the approach and to optimize it.

Ongoing application of the process by PricewaterhouseCoopers will provide some of this information, and I expect to collect Inspection Evaluations from each project. In particular, I will be looking for variations from:

- very large projects, with Functional Specification teams of more than thirty individuals,
- small projects, with Functional Specification teams of less than five individuals,
- projects that employ alternative methodology sequences, such as those that involve early rapid prototyping as a design technique,
- projects with different design paradigms, such as object oriented design, and
- projects that result in hybrid solutions, incorporating packaged software or turnkey elements.

Eventually, it would be valuable to correlate inspection results with acceptance test results by application to a statistically significant number of projects. This may never be achievable because of the variation between projects in scope, technology and team. Some large development organizations may be able to achieve enough homogeneity in their projects to draw quantitative conclusions.

While recent advances in design and development tool technology have simplified the translation of user requirements into finished code, it seems unlikely that the Functional Specification will ever become unnecessary. Until it does, its quality will remain a primary factor in the success of system development projects.

Bibliography

- Ackerman 82 F. Ackerman, A. Ackerman, R. Ebenau, "A Software Inspections Training Program," Proc. COMPSAC '82: Computer Software and Applications Conference, IEEE CS Press, Los Alamitos, CA, 1982, pp. 443-444
- Ackerman 84 F. Ackerman, P. Fowler, R. Ebenau, "Software Inspections and the Industrial Production of Software," in Software Validation, H.L. Hausen, ed., Elsevier, Amsterdam, 1984, pp. 13-40.
- Basili 97 V. Basili, "Evolving and Packaging Reading Technologies", *Journal of Systems and Software*, 38 (1), July 1997, pp. 3-12
- Blackburn 96 J. Blackburn, G. Scudder, L. Van Wassenhove, "Improving Speed and Productivity of Software Development: A Global Survey of Software Developers," *IEEE Transactions on Software Engineering*, 22 (12), December 1996, pp. 875-885
- Blakey 91 F. Blakely, M. Boles, "A Case Study of Code Inspections," *Hewlett-Packard Journal*, 42 (4), October 1991, pp. 58-63
- Chernak 96 Y. Chernak, "A Statistical Approach to the Inspection Checklist Formal Synthesis and Improvement," *IEEE Transactions on Software Engineering*, 22 (2), December 1996, pp. 866-874
- Daily 92 K. Daily, "Quality Management for Software," NCC Blackwell, Oxford, England, 1992
- Doolan 92 E. Doolan, "Experience with Fagan's Inspection Method," *Software-Practice and Experience*, 22 (2), February 1992, pp. 173-182
- Fagan 76 M. Fagan, "Design and Code Inspections to Reduce Errors in Program Development," *IBM Systems Journal*, 15 (3), 1976, pp. 182-211
- Fagan 86 M. Fagan, "Advances In Software Inspections," *IEEE Transactions on Software Engineering*, 12 (7), July 1986, pp. 744-751
- Fox 97 C. Fox, W. Frakes, "The Quality Approach: is IT Delivering?", *editorial in Communications of the ACM*, 40 (6), June 1997, pp. 25-29

- Parnas 87 D. Parnas, D. Weiss, "Active Design Reviews: Principles and Practices," *Journal of Systems and Software*, No. 7, Elsevier, New York, 1987, pp. 259-265
- Porter 97a A. Porter, P. Johnson, "Assessing Software Review Meetings: Results of a Comparative Analysis of Two Experimental Studies," *IEEE Transactions on Software Engineering*, 23 (3), March 1997, pp. 129-145
- Porter 97b A. Porter, H. Siy, C. Toman, L. Votta, "An Experiment to Assess the Cost-Benefits of Code Inspections in Large Scale Software Development," *IEEE Transactions on Software Engineering*, 23 (6), June 1997, pp. 329-346
- PwC 97 PricewaterhouseCoopers, "Summit D Version 5," PricewaterhouseCoopers Summit D Centre, Princeton, NJ, 1997
- Raz 97 T. Raz, A. Yaung, "Factors Affecting Design Inspection Effectiveness in Software Development," *Information and Systems Technology*, 39 (4), March 1997, pp. 297-305
- Schulmeyer 87 G. Schulmeyer, J. McManus, eds. , "Handbook of Software Quality Assurance," Van Nostrand Reinhold, New York, 1987
- Shirey 92 G. Shirey, "How Inspections Fail," *Proceedings of the 9th International Conference on Testing Computer Software*, 1992, pp. 151-159
- Siddiqi 96 J. Siddiqi, M. Shekaran, "Requirements Engineering: The Emerging Wisdom," *IEEE Software*, 13 (2), March 1996, pp. 15-19
- Thayer 97 R. Thayer, M. Dorfman, eds., "Software Requirements Engineering," IEEE Computer Society Press, Los Alamitos, CA, 1997
- Verner 97 J. Verner, N. Cerpa, "Prototyping: Does Your View of its Advantages Depend on Your Job?," *Journal of Systems and Software*, 36 (1), January 1997, pp. 3-16
- von Mayrhauser 90 A. von Mayrhauser, "Software Engineering Methods and Management," Academic Press, San Diego, CA, 1990

- Votta 93 L. Votta, "Does Every Inspection Need a Meeting?," *ACM Software Engineering Notes*, 18 (5), December 1993, pp. 107-114
- Weller 93 E. Weller, "Lessons from Three Years of Inspection Data," *IEEE Software*, 10 (5), Sept. 1993, pp. 38-45
- Xenos 97 M. Xenos, D. Christodouakis, "Measuring Perceived Software Quality," *Information and Software Technology*, 39 (6), June 1997, pp. 417-424
- Yourdon 86 E. Yourdon, "Managing the Structured Techniques," Yourdon Press, Englewood Cliffs, NJ, 1986
- Yourdon 89 E. Yourdon, "Structured Walkthroughs," Yourdon Press, Englewood Cliffs, NJ, 1989

Appendix A: Inspection Report

This appendix contains a printed version of the SDS Inspection Report form that was completed by inspection team members for each Functional Specification.

SDS Inspection Report

SDS Section _____
Date Distributed _____ **Date Req'd** _____
Lead Analyst _____

Instructions for Completion

This is your opportunity to ensure that a section of the System Delivery Specification (SDS) is complete and correct. When it has been signed by the Project Steering Committee, the SDS will become the basis for final acceptance of the system. Therefore, it is important that you identify any defects in this document that would result in an unacceptable system.

First, review this inspection report form so that you understand all of the elements of the SDS and the criteria by which they are judged. Then carefully read the SDS section (and any attachments) you have been asked to inspect. As you do so, note any defects by number in the Inspection Element table, and write a description of the defect in the Defect table. If there are parts of the SDS section you do not understand, note a question by number in the Inspection Element table and write your question in the Question table. Use as much space as is necessary to fully describe the defect or question, but please keep your description concise.

Ask yourself whether the inspection element fully satisfies the inspection criteria.

You may wish to review your defects and questions with others, particularly if you are the representative to the project of a group of people. Before the Date Required, complete your defect and question list, save a copy, and return the entire Inspection Report to the Lead Analyst. The Lead Analyst will prepare responses to all defects and questions and will invite you to an inspection meeting at which you and the other inspectors will review the responses and check for any other defects.

Following the inspection meeting, the Lead Analyst will revise the SDS section and will circulate it to all inspectors. At that time, please confirm that all of your defects and questions have been adequately addressed.

SDS Inspection Report

SDS Group _____

Date _____

Inspector _____

Inspection Element	Inspection Criteria	Defects	Questions
Description	concisely and accurately describes the business transaction addressed in this section		
Sub Process	identifies all of the functional components (subsystems) that will be created to satisfy the business transaction, with a definition of the degree and manner of automation		
BRD XRef	correctly identifies the processes defined in the Business Requirements Definition that are wholly or partially addressed by these subsystems		
Process Dependency Diagram	correctly shows all interactions of these subsystems to external processes, whether automated or manual, and the linkages that will be provided between these subsystems		
Data Architecture	correctly shows all data entities and attributes used or manipulated by these subsystems, the relationships among the entities, and the domain, presentation format and default value (if any) for the attributes		
Process/Data Cross Reference (CRUD)	correctly shows the usage type of each data element by each subsystem		
Inputs and Outputs	correctly identifies all input sources and values (parameters and external files) and outputs (messages, reports and external files)		
Business Rules	correctly identifies the rules defined by the business that affect this process		
Special Edits	correctly identifies the rules governing the presentation, input and validation of data and the automated initiation of other processes; where a rule is defined, there is a set of cases that covers all possible data combinations		
User Interface	shows the format and functionality of the subsystem as it will be seen by the user, including derived or summary data, screen or page layout, and data grouping or sequencing		

Appendix B: Inspection Evaluation

This appendix contains a printed version of the SDS Inspection Evaluation form that was completed by inspection team members at the end of each inspection process, and the Inspection Results Observations that each participant completed once after all inspection processes had concluded.

SDS Inspection Evaluation

SDS Section _____

Inspector _____

Lead Analyst _____

You have been involved in the inspection of a section of the System Delivery Specification (SDS). To ensure that the inspection process is effective and efficient, we require your evaluation. Please complete this form and return it to the Lead Analyst.

What is your role on the project?	Analyst	User Representative	System Designer	Other
Were you involved in creating the SDS section that you inspected?	Yes	No		
Were you adequately prepared to perform the inspection?	Yes	No (please identify improvements)		
Were the SDS section and attachments complete?	Yes	No (please identify missing items)		
Were you able to adequately complete the Inspection Report?	Yes	No (please identify issues and suggestions)		
Did you find the inspection meeting useful?	Yes	No (please identify issues and suggestions)		
Were you able to contribute to the inspection meeting?	Yes	No (please identify reasons why)		
Were all of your defects and questions satisfactorily addressed at the inspection meeting?	Yes	No (please identify why and any suggestions)		
Were the right people at the inspection meeting?	Yes	No (please identify who should / should not be included)		
Did the revised SDS section address all of the inspection defects and questions?	Yes	No (please identify remaining issues)		
Was the timing of the process steps appropriate and did you have enough time to complete your work?	Yes	No (please identify issues and suggestions)		

Other Comments

Inspection Results Observations

Name _____

You have been involved in an application of inspection techniques the System Delivery Specification (SDS), and have already evaluated the process itself. Now, we need your observations on the value of the process to the project. Please complete this form and return it to John Roberts.

What was your role on the project? Analyst User System Other
 Representative Designer

Do you feel that the process will result in a SDS with fewer defects than in other projects? Yes No

Why do you believe this?

Do you feel that users have a better understanding of the system and how to use it than in other projects? Yes No

Why do you believe this?

Do you feel that users will be able to apply this understanding in producing better quality procedures, documentation and training than in other projects? Yes No

Why do you believe this?

Do you feel that users will identify fewer failures during acceptance testing, resulting from misunderstanding of the system, than in other projects? Yes No

Why do you believe this?

Do you feel that designers have a better understanding of the system and how to build it than in other projects? Yes No

Why do you believe this?

Do you feel that designers will be able to apply this understanding in producing better quality programs than in other projects? Yes No

Why do you believe this?

Do you feel that users will identify fewer failures during acceptance testing, resulting from system defects, than in other projects? Yes No

Why do you believe this?

Do you feel that overall perceptions of quality will be higher for this system than for other systems? Yes No

Why do you believe this?

Other Observations

Appendix C: Summit D® System Delivery Specification

"I authorize you to include the description of the System Delivery Specification, from the Summit D Methodology owned by PricewaterhouseCoopers, in your thesis. I am aware that you are granting an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis by any means and in any form or format to make it available to interested persons."



Mr. Allan Hart

Partner, PricewaterhouseCoopers

The following excerpt from the Summit D methodology of PricewaterhouseCoopers was the framework used to define the structure of the System Delivery Specification used in the projects referenced in this thesis.

SYSTEM DELIVERY SPECIFICATION

The System Delivery Specification is the documentary Major Client Deliverable developed during the Solution Definition phase. It includes information on functions, user interfaces, manual processes, automated processes, data structures, security, control, back-up and contingency requirements, as well as relevant error handling procedures.

This is a particularly crucial document because:

- it is likely to be the last specification document reviewed by the users
- sign-off of this document effectively represents approval for the remainder of the system delivery cycle
- the detailed plans and estimates for the remainder of the project will be based on its contents.

The system design will be developed based upon the information contained in the System Delivery Specification.

1. **Executive Summary:** The Executive Summary is a high level overview of the System Delivery Specification. It identifies and defines the key recommendations, assumptions, and proposed delivery strategies that will be used in the development and implementation of the system. It is appropriate in the Executive Summary to address significant user expectations and to discuss how they will be resolved.
2. **Summary of Functional Solution Approach:** This section provides the user with a generalized description of the processes, data, and inter-relations that comprise the functional solution. It should indicate, where appropriate, how the solution is broken down into subsystems. It may be appropriate to use a diagram or other graphic representation, such as a data flow diagram, to depict the inter-relations between processes. The graphic may be supplemented by narratives or reports

from automated tools. The type of graphic and level of detail is dependent upon project type and user expectations.

If an application product is to provide all or part of the solution, the product should be identified and the extent to which it provides the functional solution clearly stated (for instance, through a boundary clearly shown on the graphic representation). Functions not provided by the application product should also be clearly identified. An indication should be given of any product options likely to be selected and of any modifications to the product required. Product usage should be defined in detail in section 3.1.

3. System Specification: For each subsystem defined in the functional solution, define a model showing the inter-relations between automated and manual procedures, interfaces to other subsystems, and user interfaces. These may include software products or an existing system. A brief narrative description of each subsystem should also, when appropriate, be included.

The objective of this section is to place the subsystem within the context of the overall system and provide the user with enough detail to understand.

3.1 Product Usage Specification: The Functionality Usage Definition from PSD may be used to identify the product functionality that will support each of the processes within the configuration scope. For each process to be provided by an application product, specify the way in which the product facilities are used, including:

- brief description of how the product facilities are used, including any options used or specifically suppressed
- any modifications to be made to the standard product (Enhancement Specifications may be included)

- ways of working to show how the product will be used to support the process
- where configuration of a product has taken place, the Configuration Script used may be included.

3.2 Process Specifications: For each automated and manual process contained within a subsystem, identify and define the process specification (from Activity SDS 3.4 and Activity SDS 3.5).

The amount of documentation will depend on system complexity, novelty, criticality, the number of subsystems and external data views. These details could be cross-referenced as appendices to the body of the report.

It is appropriate to include in the process specification a definition of all applicable:

- screen and report formats (developed during Task SDS 2)
- design standards with respect to error handling, menu structures, recovery, and overall design philosophy.

The process specification, once agreed, serves as the final, user approved definition of the system. Errors in the process specification will have a significant impact on the scope of effort and overall level of user satisfaction.

4. Data Model Packet: In this section, the data models are identified, defined, and described. The relationship between the custom data model and any product data model may be included in the form of the Data Cross-Reference defined in Activity PSD 2.4. The data models are the foundation upon which the data designers will develop detailed physical design specifications for the system's data base. The specification includes:

- 4.1 Logical Data Model: Is a high level representation of the data entities and their inter-relationships. It is independent of any particular Data Base Management System. It is intended as a representation of how data is used within the scope of the system. It is common practice to graphically represent a Logical Data Model as either a hierarchy or network. There are, however, several other methods available and the choice of method is based upon the requirements of the project and user expectations.

Documentation for this section is produced in Activity SDS 4.1.

- 4.2 Preliminary Physical Data Model: It is sometimes appropriate to define the physical representation of the logical database using the structures of the target Database Management System (DBMS). The Preliminary Physical Data Model will include physical implementation characteristics.

Although producing a Physical Data Model is technically a design activity, it may be necessary to take into account some physical considerations in order to perform a detailed technical evaluation and to produce the most reliable cost projections for the Design phase.

Documentation for this section is produced in Activity SDS 4.2.

5. External Interface Specifications: This section identifies and defines any interfaces between the system as specified and other systems. The interfaces should be specified in terms of the data to be transferred, the way in which the interface will operate (although technical details may not yet be decided) and the triggering events that cause the interface to operate. See Activity SDS 3.3

6. **Security and Control Requirements:** The subsystems as defined in Section 3 above could include security and control requirements that are unique to the subsystem. Any additional system, environmental, or regulatory security related requirements should be included in this section, including the relevant Configuration Scripts used where products have been configured.

Documentation for this section is produced in Activity SDS 5.2 and Activity PSD 2.6.

7. **Back-up and Contingency Requirements:** For the system as a whole and for critical subsystems, the following back-up requirements should be defined:
- data requiring back-up
 - back-up requirements such as frequency and timeliness
 - off-site storage requirements
 - contingency scenarios.

Documentation for this section is produced in Activity SDS 5.3.

8. **Technology Infrastructure:** This section sets out the technical environment in which the system is to be implemented. This includes:
- the hardware and operating system on which the system will run
 - the data base management and transaction processing environment in which the system will operate
 - the tools to be used to build the system (where appropriate).

Where these details are not yet known, this should be clearly stated. Any requirement for the evaluation and selection of technology products should be clearly identified and the requirements stated (to the extent to which they are known).

9. **Data Conversion System Specification:** This section outlines the requirements and process specifications that will be used during the Design and Build phase to construct the conversion system. Where data is being converted to or from a product, this section may include:
- product(s) data structures
 - specification of any existing data conversion routines
 - relevant implementation solutions.
 - This section should parallel the production subsystem specifications.

Documentation for this section is produced in Task SDS 6.

10. **Acceptance Test Specifications:** The Acceptance Test Specification identifies and defines the approach to be used in evaluating the quality of the completed system. It could include:
- acceptance test plan
 - acceptance test design specifications
 - acceptance test case specifications
 - acceptance threshold defining the maximum number and type of errors for system acceptance.

The contents of the Acceptance Test Specification should be revised as system and databases are revised.

For more details refer to the Activity SDS 7.3 which provides a more detailed definition of the content and structure of the Acceptance Test deliverables.

11. **Plans and Estimates:** In this section the plan and resources required for completing the remainder of the project will be identified and defined. Where a package is being implemented as part or all of the solution, the package configuration plan will be available. The level of detail and manner of presenting the plan will vary from project to project depending on the type of project (i.e.,

the particular route map used), the size of the project, and user expectations. The delivery test and transition plans should be emphasized since user commitment to their internal resources will often be needed.

At a minimum, the plan should contain enough information to provide the user with a baseline from which to evaluate the cost, time, risks, and resources required to complete the project and to allow the user to give an informed confirmation and approval for the project to continue.

The presentation approach and level of detail required is dependent upon the type of system under development, the structure of the project, the tools being used to support the planning activities, and the user's expectations.

Appendix D: Inspection Plan

INSPECTION SCHEDULE

Grp	Module	BRD w/t	BRD Pkg to Project Office	Doc. Pkg. to Inspect Team	Inspection Reports to Proj Office	Inspect. Meeting	Final Version
1	Infrastructure: Reporting / Schedule	06-Aug	13-Aug	14-Aug	18-Aug	20-Aug	21-Aug
	Infrastructure: e-mail, audit, security	05-Aug	06-Aug	13-Aug	14-Aug	18-Aug	20-Aug
2	Batch/Load Process: BI	05-Aug	07-Aug	12-Aug	13-Aug	14-Aug	18-Aug
	Batch/Load Process: TS/RN	05-Aug	07-Aug	12-Aug	13-Aug	14-Aug	18-Aug
3	Conversion - Processes	05-Aug	06-Aug	11-Aug	12-Aug	12-Aug	14-Aug
	Conversion - Reconciliation	07-Aug	10-Aug	12-Aug	12-Aug	12-Aug	14-Aug
4	On-line Processes: DO	28-Jul	06-Aug	07-Aug	10-Aug	12-Aug	19-Aug
	On-line Processes: LL	10-Aug	10-Aug	11-Aug	13-Aug	17-Aug	19-Aug
5	Reporting	06-Aug	07-Aug	10-Aug	13-Aug	17-Aug	19-Aug

Appendix E: Sample Inspection Reports

The following inspection reports are representative samples. They were completed on paper copies of the inspection report form, and have been keyed verbatim except that individuals' names have been removed. The first inspection report was completed by an analyst and the second by a user.

SDS Inspection Report

SDS Group Online Processes

Date Aug. 11/98 Inspector (deleted)

Inspection Element	Inspection Criteria	Defects	Questions
Description	concisely and accurately describes the business transaction addressed in this section	1	1
Sub Process	identifies all of the functional components (subsystems) that will be created to satisfy the business transaction, with a definition of the degree and manner of automation		
BRD XRef	correctly identifies the processes defined in the Business Requirements Definition that are wholly or partially addressed by these subsystems	2	
Process Dependency Diagram	correctly shows all interactions of these subsystems to external processes, whether automated or manual, and the linkages that will be provided between these subsystems		
Data Architecture	correctly shows all data entities and attributes used or manipulated by these subsystems, the relationships among the entities, and the domain, presentation format and default value (if any) for the attributes		
Process/Data Cross Reference (CRUD)	correctly shows the usage type of each data element by each subsystem		
Inputs and Outputs	correctly identifies all input sources and values (parameters and external files) and outputs (messages, reports and external files)		
Business Rules	correctly identifies the rules defined by the business that affect this process		
Special Edits	correctly identifies the rules governing the presentation, input and validation of data and the automated initiation of other processes; where a rule is defined, there is a set of cases that covers all possible data combinations		
User Interface	shows the format and functionality of the subsystem as it will be seen by the user, including derived or summary data, screen or page layout, and data grouping or sequencing		

Performance and Security	correctly defines performance metrics or security restrictions that vary from the system standard		
Special Processing	correctly defines any other events not included in process, data, user interface or performance and security elements		
Outstanding Issues	identifies any business or technical issues that have not been resolved and that may affect these subsystems; the assumptions upon which this design is based are explicitly defined		
Document Format	document conforms to project standard for layout, identification, and version control		
Other Comments		2 - 5	2 - 12

Defect #	Defect Description
1	BRD 1.1.3 last line of description should include "contact MHR"
2	BRD 1.4.1 and 1.4.2 are the same
3	EM001 - SDS writeup is missing
4	BRD - no page numbers & no file name in footer
5	Screen mockup, FB SIN History, Business Rules cannot be required if it is display only

Question #	Question
1	SN036 Description - it reads like we are maintaining screen reports. Is this intended?
2	SN036 Special Processing - what is meant by "scripts"
3	Can we combine Reports RP208a & b
4	SN028b can a Pulled Cheque be unapproved if sent to MHR
5	RP208b do we want a column for "Sent to MHR"
6	BRD 2.4.2 The drawings don't seem to be standard. Did we segment too much?
7	SN000 - Edits: Could we state the Client Summary changing to reflect the selected Payment in a more general way?
8	FB Client Summary has Eligible Dependents. Isn't this time/period dependent? What period
9	Where do we see the history of Entitlement changes (versions side by side)
10	Is sorting payments by cheque types within a month necessary? This is not a regular sort
11	Where do we go to see all the clients that have the same SIN?
12	NCBS Name History (SN121) why is there no To, From like record received from Revenue Canada?

SDS Inspection Report

SDS Group Online Processes

Date Aug. 14/98 Inspector (deleted)

Inspection Element	Inspection Criteria	Defects	Questions
Description	concisely and accurately describes the business transaction addressed in this section		
Sub Process	identifies all of the functional components (subsystems) that will be created to satisfy the business transaction, with a definition of the degree and manner of automation		
BRD XRef	correctly identifies the processes defined in the Business Requirements Definition that are wholly or partially addressed by these subsystems		
Process Dependency Diagram	correctly shows all interactions of these subsystems to external processes, whether automated or manual, and the linkages that will be provided between these subsystems		
Data Architecture	correctly shows all data entities and attributes used or manipulated by these subsystems, the relationships among the entities, and the domain, presentation format and default value (if any) for the attributes		
Process/Data Cross Reference (CRUD)	correctly shows the usage type of each data element by each subsystem		
Inputs and Outputs	correctly identifies all input sources and values (parameters and external files) and outputs (messages, reports and external files)		
Business Rules	correctly identifies the rules defined by the business that affect this process		
Special Edits	correctly identifies the rules governing the presentation, input and validation of data and the automated initiation of other processes; where a rule is defined, there is a set of cases that covers all possible data combinations		
User Interface	shows the format and functionality of the subsystem as it will be seen by the user, including derived or summary data, screen or page layout, and data grouping or sequencing		

Performance and Security	correctly defines performance metrics or security restrictions that vary from the system standard		
Special Processing	correctly defines any other events not included in process, data, user interface or performance and security elements		
Outstanding Issues	identifies any business or technical issues that have not been resolved and that may affect these subsystems; the assumptions upon which this design is based are explicitly defined		
Document Format	document conforms to project standard for layout, identification, and version control		
Other Comments			

Note written at bottom of page: "see over", referring to list of defects/questions

1	BRD 2.4.2 - SN043 - Column totals should be deducting negative amounts (e.g. FB Basic amount total has not accounted for negative \$100 pulled chq for Oct.)
2	BRS 2.4.2 - SN043 & SN008 - Special Processing: Revisit requirement to <u>highlight</u> the most current payment displayed by default when these screens are first displayed for a given client. Suggestion: do not highlight a payment until user selects a payment.
3	BRD 2.4.2 - SN027 - Suggest we do not need this screen. NCBS problems are resolved by RevCan. Comment can be entered on screen SN107 under the FB tab.
4	BRD 3.3.4 - BRD Document set out a requirement to log all changes made on this screen. No audit trail provision appears to be included in current write-up.

Appendix F: Sample Inspection Meeting Report

The following inspection report is a representative sample of the notes taken by the Moderator / Scribe at an inspection meeting, showing additional defects or clarifications made during the meeting. It was completed electronically and is included in its entirety, but names have been replaced with generic identifiers for individuals.

INSPECTION REPORT

Group: On-line processes, Reports, Report Inventory

Sub-group: All of 3.6, and 3.8.4

Date: August 19, 9:00 a.m.

Inspectors: User 1
User 2
Data Architect
Analyst 2 (author)
User 3
Analyst 2
Project Manager (moderator)
User 4

BRD	Module	Defect/Question
3.6.11	RP037 to RP044	Add "Annualized Estimate" to end of each report. Calculate by dividing year-to-date \$ total (for each of B, F, and Total cost) by #of months so far, and multiply by 12 to get projections for year for each of B, F, and Total Cost.
3.6.11	RP044	Revise Report heading to indicate "Bonus Year"
3.6.12	RP218	This report is actually part of the BRD 2.0 database load (payments) process ; Detail is not required; require only one lump sum/one line and can be included with other control statistics in appropriate BRD 2.0 report.
3.8.4	All	Report Inventory display will be changed so that entire list of Groups is displayed, then a group selected, the possible report types (activities) displayed, and a report type selected to display actual instances in the report inventory.
Data Model		No account code required in data model. (Data Architect)

Forward Plan: Analyst to correct defects listed in table above (first four entries).

Review only changes to 3.8.4 with User 2 and Data Architect (personally or via e-mail)

Deliver revised package (electronically) to Project Office by end-of-day, August 20.

Appendix G: Sample Inspection Evaluations

The following inspection evaluations are representative samples. They were completed on paper copies of the inspection evaluation form, and have been keyed verbatim except that individuals' names have been removed. The first inspection evaluation was completed by an analyst and the second by a user.

SDS Inspection Evaluation

SDS Section Online Processes

You have been involved in the inspection of a section of the System Delivery Specification (SDS). To ensure that the inspection process is effective and efficient, we require your evaluation. Please complete this form and return it to the Lead Analyst.

What is your role on the project?	<input checked="" type="radio"/> Analyst	User Representative	System Designer	Other
Were you involved in creating the SDS section that you inspected?	<input checked="" type="radio"/> Yes	No		
Were you adequately prepared to perform the inspection?	Yes	<input checked="" type="radio"/> No (please identify improvements)		
		<i>More design time with all stakeholders would have made a more complete SDS.</i>		
Were the SDS section and attachments complete?	<input checked="" type="radio"/> Yes	No (please identify missing items)		
Were you able to adequately complete the Inspection Report?	<input checked="" type="radio"/> Yes	No (please identify issues and suggestions)		
Did you find the inspection meeting useful?	<input checked="" type="radio"/> Yes	No (please identify issues and suggestions)		
		<i>People involved in the design should be in the inspection and vice versa.</i>		
Were you able to contribute to the inspection meeting?	<input checked="" type="radio"/> Yes	No (please identify reasons why)		
Were all of your defects and questions satisfactorily addressed at the inspection meeting?	Yes	No (please identify why and any suggestions)		
Were the right people at the inspection meeting?	Yes	<input checked="" type="radio"/> No (please identify who should / should not be included)		
		<i><arrow drawn to preceding comment></i>		
Did the revised SDS section address all of the inspection defects and questions?	<input checked="" type="radio"/> Yes	No (please identify remaining issues)		
		<i>But, most of the minor changes or edits were not given formal discussion.</i>		
Was the timing of the process steps appropriate and did you have enough time to complete your work?	Yes	<input checked="" type="radio"/> No (please identify issues and suggestions)		
		<i>As the inspections matured, there was less time for the later inspection and less turnaround time. Also more was requested of later SDS's.</i>		

Inspection Results Observations

You have been involved in an application of inspection techniques the System Delivery Specification (SDS), and have already evaluated the process itself. Now, we need your observations on the value of the process to the project. Please complete this form and return it to John Roberts.

What was your role on the project?

Analyst

User

System

Other

Representative

Designer

Do you feel that the process will result in a SDS with fewer defects than in other projects?

Yes

No

Why do you believe this?

More people have reviewed it. But the cost may not justify this method.

Do you feel that users have a better understanding of the system and how to use it than in other projects?

Yes

No Not sure

Why do you believe this?

The users seemed to be going on faith in some instances. The hybrid of logical/physical made for inconsistencies.

Do you feel that users will be able to apply this understanding in producing better quality procedures, documentation and training than in other projects?

Yes

No

Why do you believe this?

They should have a common understanding of their business.

Do you feel that users will identify fewer failures during acceptance testing, resulting from misunderstanding of the system, than in other projects?

Yes

No

Why do you believe this?

Everything had to be explained to them.

Do you feel that designers have a better understanding of the system and how to build it than in other projects?

Yes

No Maybe

Why do you believe this?

Designers have a better understanding now than before but that doesn't mean it is better than on other projects.

Do you feel that designers will be able to apply this understanding in producing better quality programs than in other projects?

Yes No *Maybe*

Why do you believe this?

Same as previous

Do you feel that users will identify fewer failures during acceptance testing, resulting from system defects, than in other projects?

Yes No

Why do you believe this?

<underlined "other projects">

They may identify fewer failures.

Do you feel that overall perceptions of quality will be higher for this system than for other systems?

Yes No *Possibly*

Why do you believe this?

More effort gives the perception of more quality.

Other Observations

- *Allow for more time (Client & Analyst)*
- *Use fewer reviewers*
- *Spend more time working/reviewing the connections between SDS's*
- *Don't change standards/environment once it is started*
- *If the SDS are to involve physical design elements, then it should be done with physical design considerations in mind. This would 1) manage client expectations, 2) minimize subsequent consultation with clients on physical design changes, 3) reduce effort/rework in TSD*

SDS Inspection Evaluation

SDS Section Online Processes

You have been involved in the inspection of a section of the System Delivery Specification (SDS). To ensure that the inspection process is effective and efficient, we require your evaluation. Please complete this form and return it to the Lead Analyst.

What is your role on the project?	Analyst	User Representative	System Designer	Other
Were you involved in creating the SDS section that you inspected?	<input checked="" type="radio"/> Yes	<input type="radio"/> No		
Were you adequately prepared to perform the inspection?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify improvements) <i>But very short turnaround time</i>		
Were the SDS section and attachments complete?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify missing items)		
Were you able to adequately complete the Inspection Report?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify issues and suggestions)		
Did you find the inspection meeting useful?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify issues and suggestions)		
Were you able to contribute to the inspection meeting?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify reasons why)		
Were all of your defects and questions satisfactorily addressed at the inspection meeting?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify why and any suggestions)		
Were the right people at the inspection meeting?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify who should / should not be included)		
Did the revised SDS section address all of the inspection defects and questions?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify remaining issues)		
Was the timing of the process steps appropriate and did you have enough time to complete your work?	<input checked="" type="radio"/> Yes	<input type="radio"/> No (please identify issues and suggestions) <i>1/2 day.</i>		

Other Comments

Inspection Results Observations

You have been involved in an application of inspection techniques the System Delivery Specification (SDS), and have already evaluated the process itself. Now, we need your observations on the value of the process to the project. Please complete this form and return it to John Roberts.

What was your role on the project?

Analyst User Representative System Designer Other

Do you feel that the process will result in a SDS with fewer defects than in other projects?

Yes No

Why do you believe this?

All participants have had input. All modules, reports and screens have been thoroughly analyzed.

Do you feel that users have a better understanding of the system and how to use it than in other projects?

Yes No

Why do you believe this?

Better understanding from the walkthroughs. Understand other areas (RevISB, PwC) concerns and methodology.

Do you feel that users will be able to apply this understanding in producing better quality procedures, documentation and training than in other projects?

Yes No

Why do you believe this?

Develop better quality tests based on greater understanding of processes & procedures.

Do you feel that users will identify fewer failures during acceptance testing, resulting from misunderstanding of the system, than in other projects?

Yes No

Why do you believe this?

Hopefully most of the concerns have been addressed or documented with further clarification.

Do you feel that designers have a better understanding of the system and how to build it than in other projects?

Yes No

Do you feel that designers will be able to apply this understanding in producing better quality programs than in other projects?

Yes No

Why do you believe this?

They have a greater understanding of the users needs. The majority of the concerns were addressed and resolved.

Do you feel that users will identify fewer failures during acceptance testing, resulting from system defects, than in other projects?

Yes No

Why do you believe this?

Why do you believe this?

More so than if we did not have this process.

Do you feel that overall perceptions of quality will be higher for this system than for other systems?

Yes No

Why do you believe this?

Most issues have been addressed and identified. Good input from all parties to enhance the quality.

Other Observations

Appendix H – Project Profiles

The following projects were referenced in the body of this thesis. The first project (ISAP) supplied data about defect frequencies, and served as a useful subjective comparison to the second project (FBR). During the FBR project, the inspection process was applied and the results form the basis for the thesis conclusions.

- Project:** **Information Systems Alignment Project (ISAP)**
- Client:** British Columbia Buildings Corporation (BCBC), the public sector organization responsible for management of all government property and buildings in BC.
- Started:** September 1995 (Coopers & Lybrand's involvement)
- Ended:** July 1997
- Scope:** Replacement of all corporate systems used by BCBC, including General Ledger, Purchasing, Accounts Payable, Fixed Assets, Accounts Receivable, Project Accounting, Inventory, Space Management, Pricing, Client Budget, Client Agreements, Requests, Project Tracking, Lease Control, Revenue Forecasting, Tax Management. The first six applications were replaced by Oracle Financial Applications, while the others were custom developed.
- Effort:** ~ 12,000 person-days
- Team Size:** 15-25 full time; 150 overall
- Users:** ~ 600
- Technology:** Sun / Solaris servers, Ethernet TCP/IP network and remote private line to Pentium / NT workstations. Oracle RDBMS, Designer 2000 and Developer 2000.

Functional Specification

- Approach:** JAD sessions and paper documentation with user focus groups. Business scenario charting in ABC Flowcharter. Data design in Oracle Designer 2000. Informal walkthroughs and document reviews.
- Size:** ~ 275 pages of text and diagrams, plus ~60 pages of business scenario charts.
- Effort:** ~ 400 person-days of analyst group; ~ 1200 person-days of users

Project: Family Bonus Renewal (FBR)

- Client:** Ministry of Finance and Corporate Relations (MFCR) of British Columbia; Revenue Division; Family Bonus Program. This group administers the transfer of information between the Federal government (Revenue Canada) and the Ministry of Human Resources to operate the Family Bonus Program, an income assistance program for low-income families.
- Started:** September 1997
- Ended:** planned March 1999
- Scope:** Creation of a new integrated application to support the business functions of receiving and verifying Revenue Canada Data, loading the Family Bonus Database and applying edits, maintaining control and summary data, viewing and reporting data, extracting data for transfer to Ministry of Human Resources, and security and audit.
- Effort:** ~ 4,000 person-days
- Team Size:** 15 - 20 full time
- Users:** ~ 20
- Technology:** IBM RS/6000 / AIX servers, Ethernet TCP/IP network to Pentium / NT workstations. Oracle RDBMS, Designer 2000 and PL/SQL, Visual Basic, Crystal Reports.

Functional Specification

- Approach:** JAD sessions and paper documentation with user experts. Functional design in MS Word. Data design in Oracle Designer 2000. Informal walkthroughs and document reviews. Formal Functional Specification inspections.
- Size:** ~ 350 pages of text and diagrams (not including CASE reports). 157 automated modules identified.
- Effort:** ~ 300 person-days of analyst group; ~ 200 person-days of users

VITA

Surname: Roberts

Given Names: John Michael

Place of Birth: Vancouver, British Columbia, Canada

Educational Institutions Attended:

University of Victoria	1992-1998
University of Victoria	1978-1981
Selkirk College	1977-1978

Degrees Awarded:

B.Sc. University of Victoria	1982
------------------------------	------

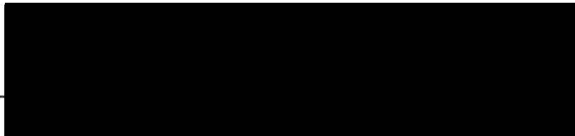
PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis:

Improving Software Quality By Inspecting Functional Specifications

Author :



John Michael Roberts

August 31, 1998